Smooth Regression Assignment

Naveen Narayanan Meyyappan

05/12/2019

## Question

Construct code for bootstrapping the Press statistic, bootstrapping along the rows.

And add the press statistic into each the gaussian smooths

Construct confidence intervals for each press statistic for the gaussian smooths.

## Answer

Smooth regression involves drawing a smooth curve on a scatter diagram to summarize a relationship, in a fashion that makes few assumptions initially about the form or strength of the relationship. It is related to (and is a special case of) nonparametric regression, in which the objective is to represent the relationship between a response variable and one or more predictor variables, again in way that makes few assumptions about the form of the relationship. In other words, in contrast to “standard” linear regression analysis, no assumption is made that the relationship is represented by a straight line (although one could certainly think of a straight line as a special case of nonparametric regression).

Another way of looking at scatter diagram smoothing is as a way of depicting the “local” relationship between a response variable and a predictor variable over parts of their ranges, which may differ from a “global” relationship determined using the whole data set. (And again, the idea of “local” as opposed to “global” relationships has an obvious geographical analogy.)

Let us define the functions for each of the smooth curves.

NOAA1=read.csv("NOAA+GISS.csv")  
my.hat.w<-function(x,wt){  
 x1<-cbind(1,x)  
 x1%\*%solve(t(x1)%\*%diag(wt)%\*%x1)%\*%t(x1)%\*%(diag(wt))  
}  
bin.mean<-function(x,y,nbin,xcol=2)  
{  
 o1<-order(x)  
 x1<-x[o1]  
 y1<-y[o1]  
 r1<-range(x)  
 inc<-(r1[2]-r1[1])/nbin  
 yvec<-NULL  
 smat<-NULL  
 for(i in 1:nbin){  
 bin.low<-r1[1]+(i-1)\*inc  
 bin.high<-r1[1]+i\*inc  
   
 I1<-x1>=bin.low  
 if(i<nbin){  
 I2<-x1<bin.high  
 }else{  
 I2<-x1<=(bin.high+200)  
 }  
 I3<-as.logical(I1\*I2)  
 yval<-mean(y1[I3])  
 n1<-sum(I3)  
 matdum<-NULL  
 for(i in 1:n1){  
 matdum<-rbind(matdum,I3\*1/n1)  
 }  
 smat<-rbind(smat,matdum)  
 yvec<-c(yvec,rep(yval,n1))  
 }  
 n99<-length(x1)  
 dferror<-length(x1)-sum(diag(2\*smat-smat%\*%(t(smat))))  
 delta1<-sum(diag(t(diag(n99)-smat)%\*%(diag(n99)-smat)))  
 R<-t(diag(n99)-smat)%\*%(diag(n99)-smat)  
 delta2<-2\*sum(diag(R%\*%R))  
 lines(x1,yvec,col=xcol)  
 ypred<-y1  
 ypred<-smat%\*%y1  
 resid<-y-ypred  
 pressvec<-sum((resid/(1-diag(smat)))^2)   
 list(smat=smat,df=sum(diag(smat)),dferror=dferror,delta1=delta1,delta2=delta2,resid=resid,pred=ypred,press=pressvec)  
   
}  
gauss.mean<-function(x,y,lambda,xcol=3,do.plot=T)  
{  
 o1<-order(x)  
 x1<-x[o1]  
 y1<-y[o1]  
 r1<-range(x)  
 smat<-NULL  
 n1<-length(x1)  
 for(i in 1:n1){  
 v1<-dnorm(x1,x1[i],lambda)  
 v1<-v1/sum(v1)  
 smat<-rbind(smat,v1)  
 }  
 yhat<-smat%\*%y1  
 if(do.plot){  
 lines(x1,yhat,col=xcol)  
 }  
 n99<-length(x1)  
 dferror<-length(x1)-sum(diag(2\*smat-smat%\*%(t(smat))))  
 delta1<-sum(diag(t(diag(n99)-smat)%\*%(diag(n99)-smat)))  
 R<-t(diag(n99)-smat)%\*%(diag(n99)-smat)  
 delta2<-2\*sum(diag(R%\*%R))  
 resid<-y1-smat%\*%y1  
 ypred<-y1  
 ypred[o1]<-smat%\*%y1  
 pressvec<-sum((resid/(1-diag(smat)))^2)   
 list(smat=smat,df=sum(diag(smat)),dferror=dferror,delta1=delta1,delta2=delta2,resid=resid,pred=ypred,press=pressvec)  
}  
gauss.mean.trunc<-function(x,y,lambda,nnn,xcol=5,do.plot=T)  
{  
 o1<-order(x)  
 x1<-x[o1]  
 y1<-y[o1]  
 r1<-range(x)  
 smat<-NULL  
 n1<-length(x1)  
 trunc.val<-n1-nnn  
 for(i in 1:n1){  
 v1<-dnorm(x1,x1[i],lambda)  
 o2<-order(v1)  
 thresh<-v1[o2[trunc.val]]  
 v1<-v1\*(v1>thresh)  
 v1<-v1/sum(v1)  
 smat<-rbind(smat,v1)  
 }  
 yhat<-smat%\*%y1  
 if(do.plot){  
 lines(x1,yhat,col=xcol)  
 }  
 n99<-length(x1)  
 dferror<-length(x1)-sum(diag(2\*smat-smat%\*%(t(smat))))  
 delta1<-sum(diag(t(diag(n99)-smat)%\*%(diag(n99)-smat)))  
 R<-t(diag(n99)-smat)%\*%(diag(n99)-smat)  
 delta2<-2\*sum(diag(R%\*%R))  
 resid<-y1-smat%\*%y1  
 ypred<-y1  
 ypred[o1]<-smat%\*%y1  
 pressvec<-sum((resid/(1-diag(smat)))^2)   
 list(smat=smat,df=sum(diag(smat)),dferror=dferror,delta1=delta1,delta2=delta2,resid=resid,pred=ypred,press=pressvec)  
   
}  
gauss.reg<-function(x,y,lambda,xcol=4,do.plot=T)  
{  
 o1<-order(x)  
 x1<-x[o1]  
 y1<-y[o1]  
 r1<-range(x)  
 smat<-NULL  
 n1<-length(x1)  
 for(i in 1:n1){  
 v1<-dnorm(x1,x1[i],lambda)  
 v1<-v1/sum(v1)  
 H1<-my.hat.w(x1,v1)  
 smat<-rbind(smat,H1[i,])  
 }  
 yhat<-smat%\*%y1  
 if(do.plot){  
 lines(x1,yhat,col=xcol)  
 }  
 n99<-length(x1)  
 dferror<-length(x1)-sum(diag(2\*smat-smat%\*%(t(smat))))  
 delta1<-sum(diag(t(diag(n99)-smat)%\*%(diag(n99)-smat)))  
 R<-t(diag(n99)-smat)%\*%(diag(n99)-smat)  
 delta2<-2\*sum(diag(R%\*%R))  
 resid<-y1-smat%\*%y1  
 ypred<-y1  
 ypred[o1]<-smat%\*%y1  
 pressvec<-sum((resid/(1-diag(smat)))^2)   
 list(smat=smat,df=sum(diag(smat)),dferror=dferror,delta1=delta1,delta2=delta2,resid=resid,pred=ypred,press=pressvec)  
}  
gauss.reg.trunc<-function(x,y,lambda,nnn,xcol=6,do.plot=T)  
{  
 o1<-order(x)  
 x1<-x[o1]  
 y1<-y[o1]  
 r1<-range(x)  
 smat<-NULL  
 n1<-length(x1)  
 trunc.val<-n1-nnn  
 for(i in 1:n1){  
 v1<-dnorm(x1,x1[i],lambda)  
 o1<-order(v1)  
 thresh<-v1[o1[trunc.val]]  
 v1<-v1\*(v1>thresh)  
 v1<-v1/sum(v1)  
 H1<-my.hat.w(x1,v1)  
 smat<-rbind(smat,H1[i,])  
 }  
 yhat<-smat%\*%y1  
 if(do.plot){  
 lines(x1,yhat,col=xcol)  
 }  
 n99<-length(x1)  
 dferror<-length(x1)-sum(diag(2\*smat-smat%\*%(t(smat))))  
 delta1<-sum(diag(t(diag(n99)-smat)%\*%(diag(n99)-smat)))  
 R<-t(diag(n99)-smat)%\*%(diag(n99)-smat)  
 delta2<-2\*sum(diag(R%\*%R))  
 resid<-y1-smat%\*%y1  
 ypred<-y1  
 ypred[o1]<-smat%\*%y1  
 pressvec<-sum((resid/(1-diag(smat)))^2)   
 list(smat=smat,df=sum(diag(smat)),dferror=dferror,delta1=delta1,delta2=delta2,resid=resid,pred=ypred, press=pressvec)  
}

Let us plot the different smooth regression curves.

plot(NOAA1[,3],NOAA1[,2],xlab="temperature rise",ylab="rate of billion dollar weather disasters")  
dum<-bin.mean(NOAA1[,3],NOAA1[,2],6)  
dum<-gauss.mean(NOAA1[,3],NOAA1[,2],.063)$df  
gauss.reg(NOAA1[,3],NOAA1[,2],.078,do.plot=T)$df

## [1] 6.190322

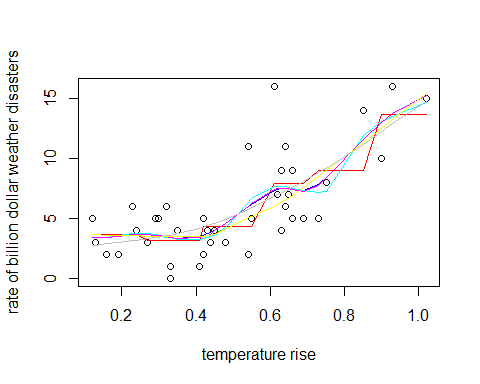
gauss.mean.trunc(NOAA1[,3],NOAA1[,2],.063,20,do.plot=T)$df

## [1] 6.18046

gauss.reg.trunc(NOAA1[,3],NOAA1[,2],.08,17,do.plot=T)$df

## [1] 6.249733

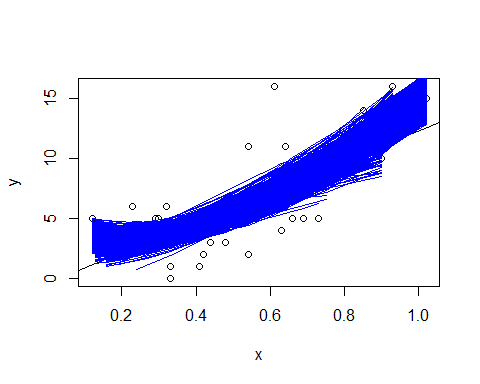
lines(lowess(NOAA1[,3],NOAA1[,2]),col=7)  
lines(smooth.spline(NOAA1[,3],NOAA1[,2]),col=8)



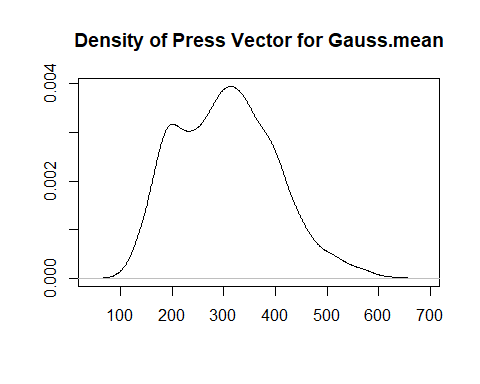
We have to bootstrap the press vector of different smooth regression curves and also plot the density of the press vector

1. Bootstrapping Gauss.mean function

bootstrap<-function(x=NOAA1[,3],y=NOAA1[,2],nboot=1000){  
 plot(x,y)  
 dum.reg<-gauss.reg(x,y,.2,do.plot=T)  
 lstr<-lsfit(x,y)  
 abline(lstr)  
 pressvec0<-dum.reg$press  
 pressvec<-NULL  
 for(i in 1:nboot){  
 v1<-sample(length(x),length(x),replace=T)  
 x1<-x[v1]  
 y1<-y[v1]  
 dum.reg1<-gauss.reg(x1,y1,.2,do.plot=T)  
 pressvec1<-dum.reg1$press  
 pressvec<-c(pressvec,pressvec1)  
 }  
 error <- qt(0.975,df=length(pressvec)-1)\*sd(pressvec)/sqrt(length(pressvec))  
 LC <- mean(pressvec)-error  
 UC <- mean(pressvec)+error  
 list(pressvec=pressvec,press=dum.reg$press, CI=list(LC,UC))  
}  
result=bootstrap()

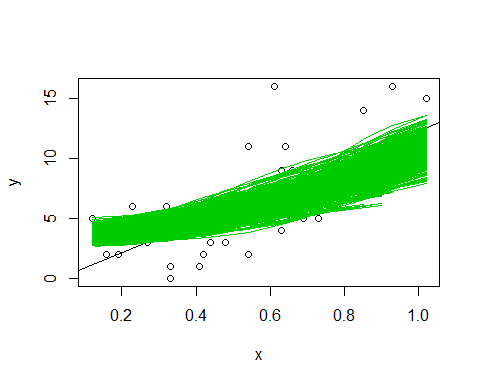


d<-density(result$pressvec)  
plot(d, main="Density of Press Vector for Gauss.mean",xlab="",ylab="")

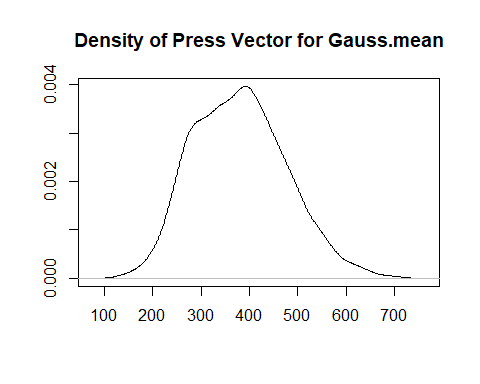


1. Bootstrapping Gauss.mean Function

bootstrap<-function(x=NOAA1[,3],y=NOAA1[,2],nboot=1000){  
 plot(x,y)  
 dum.reg<-gauss.mean(x,y,.2,do.plot=T)  
 lstr<-lsfit(x,y)  
 abline(lstr)  
 pressvec0<-dum.reg$press  
 pressvec<-NULL  
 for(i in 1:nboot){  
 v1<-sample(length(x),length(x),replace=T)  
 x1<-x[v1]  
 y1<-y[v1]  
 dum.reg1<-gauss.mean(x1,y1,.2,do.plot=T)  
 pressvec1<-dum.reg1$press  
 pressvec<-c(pressvec,pressvec1)  
 }  
 error <- qt(0.975,df=length(pressvec)-1)\*sd(pressvec)/sqrt(length(pressvec))  
 LC <- mean(pressvec)-error  
 UC <- mean(pressvec)+error  
 list(pressvec=pressvec,press=dum.reg$press, CI=list(LC,UC))  
}  
result=bootstrap()

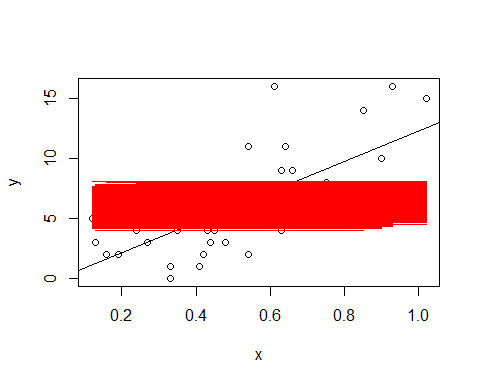


d<-density(result$pressvec)  
plot(d, main="Density of Press Vector for Gauss.mean",xlab="",ylab="")

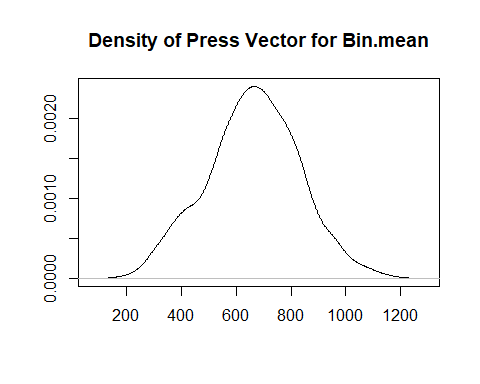


1. Bootstrapping Bin.mean Function

bootstrap<-function(x=NOAA1[,3],y=NOAA1[,2],nboot=1000){  
 plot(x,y)  
 dum.reg<-bin.mean(x,y,.2)  
 lstr<-lsfit(x,y)  
 abline(lstr)  
 pressvec0<-dum.reg$press  
 pressvec<-NULL  
 for(i in 1:nboot){  
 v1<-sample(length(x),length(x),replace=T)  
 x1<-x[v1]  
 y1<-y[v1]  
 dum.reg1<-bin.mean(x1,y1,.2)  
 pressvec1<-dum.reg1$press  
 pressvec<-c(pressvec,pressvec1)  
 }  
 error <- qt(0.975,df=length(pressvec)-1)\*sd(pressvec)/sqrt(length(pressvec))  
 LC <- mean(pressvec)-error  
 UC <- mean(pressvec)+error  
 list(pressvec=pressvec,press=dum.reg$press, CI=list(LC,UC))  
}  
result=bootstrap()



d<-density(result$pressvec)  
plot(d, main="Density of Press Vector for Bin.mean",xlab="",ylab="")



## References

1. <http://geog.uoregon.edu/bartlein/old_courses/geog414f03/lectures/lec05.htm>