Advanced Statistical Modeling

Non-parametric models - Generalized nonparametric regression model

Haoran Mo, Alexandra Yamaui

December 9th, 2017

When we are working with nonparametric logistic model the regression function is approximated as $\theta(x) \approx \beta_0^t + \beta_1^t(x-t)$, x being close to t and where $\theta(x) = log(\frac{p(x)}{1-p(x)})$ and $p(x) = E(Y|X=x)$.

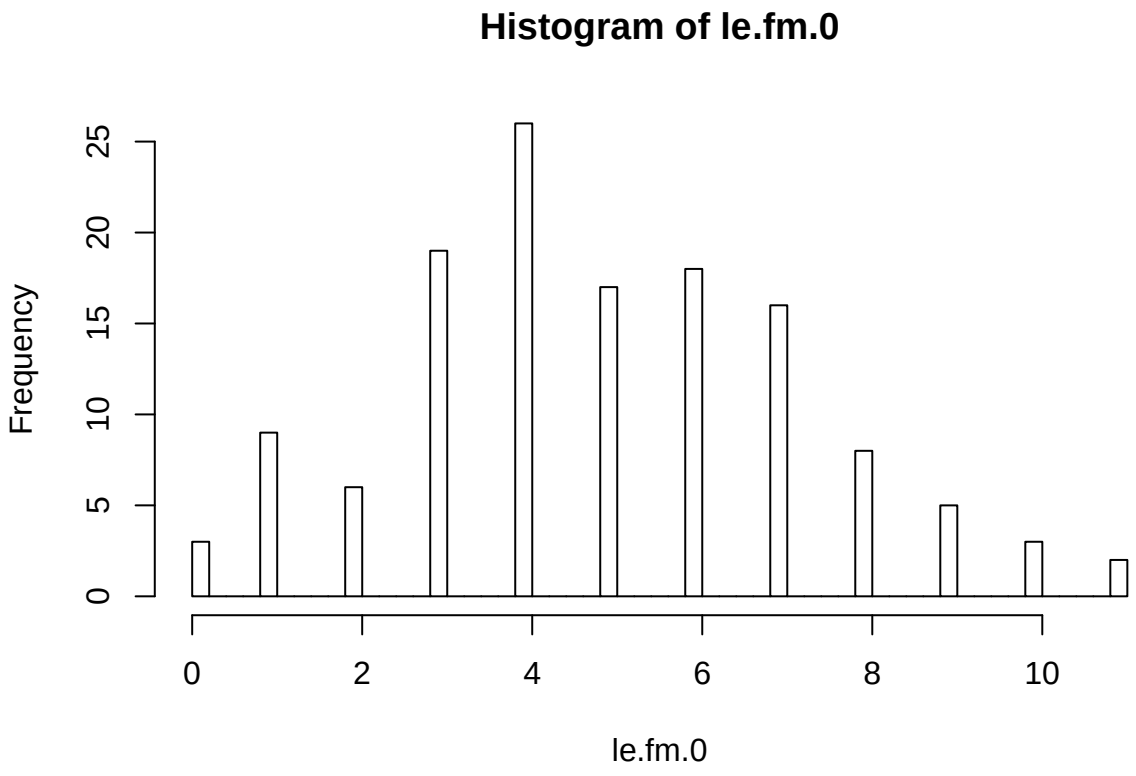
Fitting a local logistic regression involves choosing the right bandwidth, and in order to do that we are going to use the method of maximum log-likehood for new observations, which we will estimate using leave-one-out cross-validation. Below is the function of log-likelihood using h as bandwith

$$l_{CV}(h) = \sum_{i=1}^{n} log(\hat{Pr}_h^{(-1)}(Y = y_i|X = x_i)) = \sum_{i=1}^{n}(y_i log(\frac{p_i^{(-i)}}{1-p_i^{(-i)}}) + log(1-p_i^{(-i)}))$$

where $(y_i log(\frac{p_i^{(-i)}}{1-p_i^{(-i)}}) + log(1-p_i^{(-i)}))$ is the contribution to the log-likelihood function of each observation.

In this exercise we are going to choose the bandwidth for a logistic regression model that has as response variable the difference in life expectancy between men and women (le.fm) in 132 countries. The data set used contains development indicators of the different countries. The variable le.fm always takes non-negative values, except for one country, so we are going to take 0 as minimum value.

```
countries<-read.table(file="countries.txt",head=T,row.names=2,dec=",")
attach(countries)
le.fm.0 <- pmax(0,le.fm)
hist(le.fm.0,br=40)
```



We can see that the frequency of difference of life expectancy between men and women has its highest value at 4. Because this variable is discrete and bounded at 0 we will use local Poisson regression to estimate this variable.

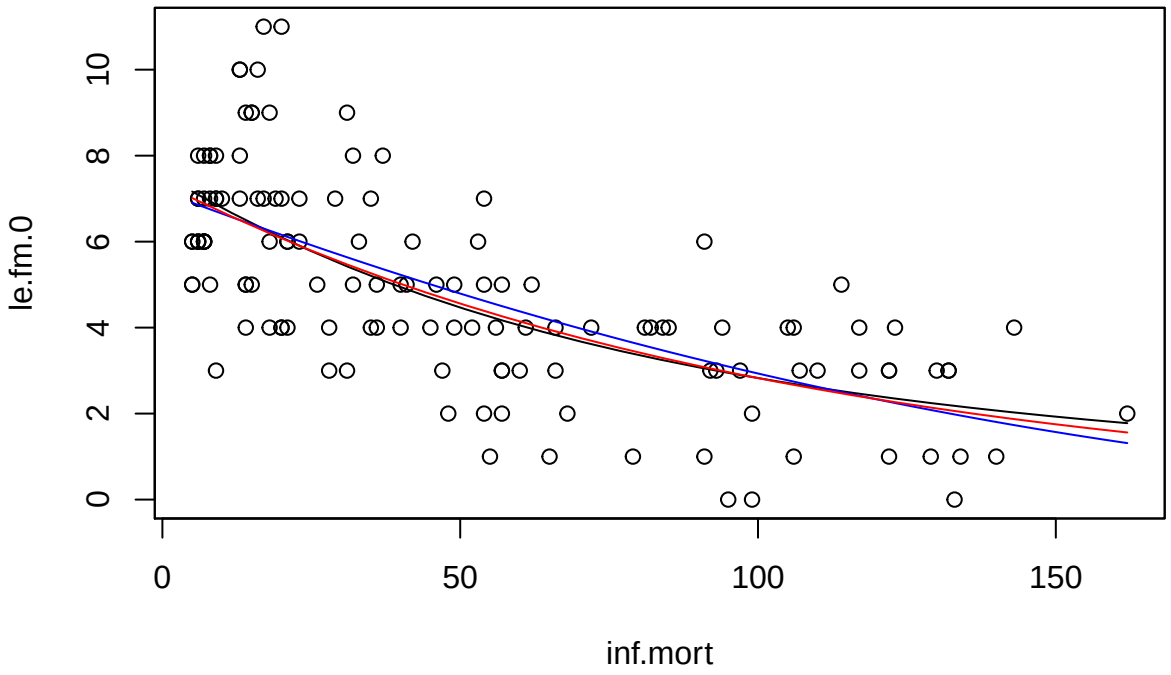
First we are going to implement the function to choose the right bandwith for local Poisson regression.

```
# Write your own CV bandwidth choice script for the local Poisson regression.
# h.cv.sm.possion.R
#
# method can be equal to 'loglik.CV' (default) or 'prob.missclas.CV'
h.cv.sm.poisson <- function(x, y, rg.h = NULL, l.h = 10, method = prob.missclas.CV){
  cv.h <- numeric(l.h)
  if (is.null(rg.h)){
    hh <- c(h.select(x,y,method="cv"), h.select(x,y,method="aicc"))#,hcv(x,y))
    rg.h <- range(hh)
  }
  i <- 0
  gr.h <- exp( seq(log(rg.h[1]/1.1), log(rg.h[2]*1.1), l=l.h))
  for (h in gr.h){
    i <- i+1
    cv.h[i] <- method(x,y,h)
  }
  return(list(h=gr.h,cv.h=cv.h, h.cv = gr.h[which.min(cv.h)]))
}
# method 1
prob.missclas.CV <- function(x,y,h){
  n <- length(x)
  pred <- sapply(1:n,
                 function(i,x,y,h){
                   sm.poisson(x=x[-i],y=y[-i],h=h,eval.points=x[i],display="none")$estimate
                 },   x,y,h)
  return(sum(abs(pred-y)>.5)/n)
}
# method 2
loglik.CV <- function(x,y,h){
  n <- length(x)
  pred <- sapply(1:n,
                 function(i,x,y,h){
                   sm.poisson(x=x[-i],y=y[-i],h=h,eval.points=x[i],display="none")$estimate
                 },   x,y,h)
  return(-sum( y*log(pred/(1-pred)) + log(1-pred) ))
}
```

Now, we are going to use the previous function to choose the bandwidth.

```
h.list <- h.cv.sm.poisson(inf.mort,le.fm.0)#rg.h=c(4,11)

aux <- sm.poisson(inf.mort,le.fm.0,h=h.list$h.cv, col=1) # h=10
sm.regression(inf.mort,le.fm.0,h=h.list$h.cv,col=4,add=T) # h=10
aux.glm <- glm(le.fm.0 ~ inf.mort,family=poisson)
pred <- predict(aux.glm,
                newdata=data.frame(inf.mort=aux$eval.points),
                type="response")
lines(aux$eval.points,pred,col=2)
```




```
h.list <- h.cv.sm.poisson(life.exp,le.fm.0)#rg.h=c(4,11)

aux <- sm.poisson(life.exp,le.fm.0,h=h.list$h.cv, col=1) # h=10
sm.regression(life.exp,le.fm.0,h=h.list$h.cv,col=4,add=T) # h=10
aux.glm <- glm(le.fm.0 ~ life.exp,family=poisson)
pred <- predict(aux.glm,
                newdata=data.frame(life.exp=aux$eval.points),
                type="response")
lines(aux$eval.points,pred,col=2)
```