Nonparametric Statistics practice

Daniel Alonso

March 19, 2021

Exercises

Category A: Problem 6

- **Exercise 5.11**. The *challenger.txt* dataset contains dataset contains information regarding the state of the solidrocket boosters after launch for 23 shuttle flights prior the Challenger launch. Each row has, among others, the variables *fail.field* (indicator of whether there was an incident with the O-rings), *nfail.field* (number of incidents with the O-rings), and *temp* (temperature in the day of launch,measured in degrees Celsius).

a) Fit a local logistic regression (first degree) for *fails.field* ~ *temp*, for three choices of bandwidths: one that oversmooths, another that is somehow adequate, and another that undersmooths. Do the effects of *temp* on *fails.field* seem to be significant?

b) Obtain $\hat{h}_{LCV}$ and plot the LCV function with a reasonable accuracy.

c) Using $\hat{h}_{LCV}$, predict the probability of an incident at temperatures -0.6 (launch temperature of the Challenger) and 11.67 (specific recommendation by the vice president of engineers).

d) What are the local odds at -0.6 and 11.67? Show the local logistic models about these points, in spirit of Figure 5.1, and interpret the results.

Category B: Problem 4

- **Exercise 4.9**. Perform the following tasks:

a) Code your own implementation of the local cubic estimator. The function must take as input the vector of evaluation points x, the sample *data*, and the bandwidth h. Use the normal kernel. The result must be a vector of the same length as x containing the estimator evaluated at x.

We have implemented the local polynomial estimator for any $0 \leq p < 8$ (however, p=3 is the default parameter)

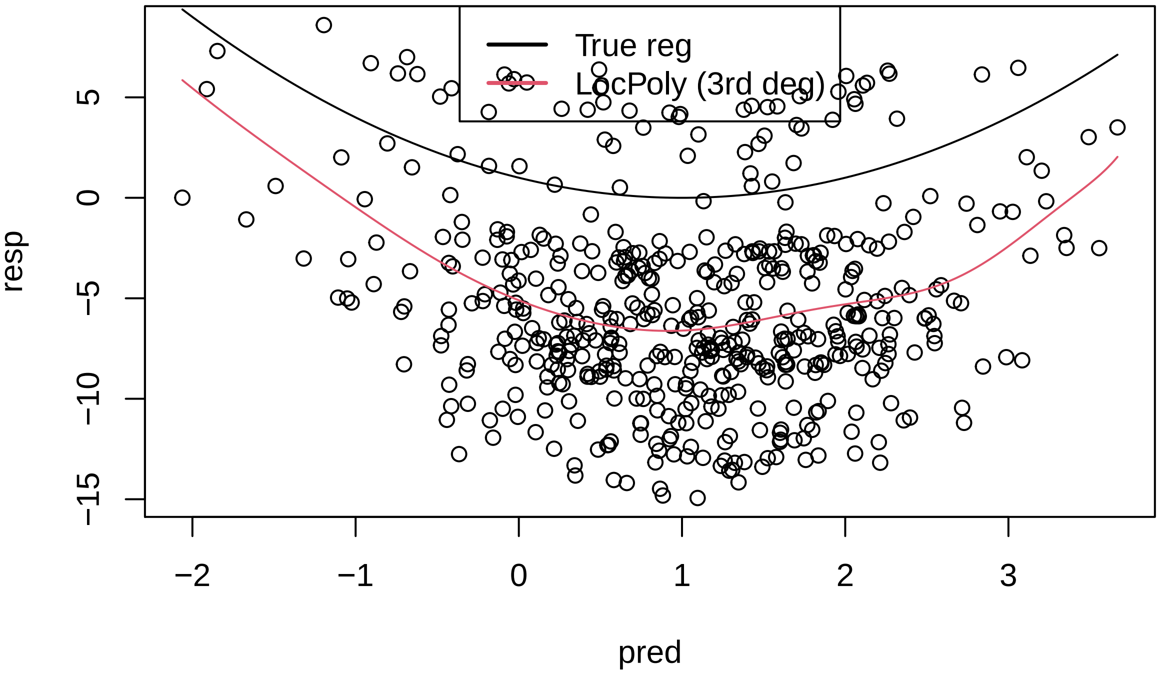
```
lce <- function(x, data, h, p=3) {
    # Resulting vector initilization
    result <- c()

    # Predictors
    predictors <- data[,1]

    # Response
    Y <- data[,2]

    # e_1
    e_1 <-  matrix(c(c(1), rep(0,p)),nrow=p+1,ncol=1)
```

```r
    # X matrix
    X <- list()
    for (par in 1:length(x)){
        # X matrix
        X[[par]] <- matrix(,nrow=length(predictors),ncol=p+1)

        # Filling up the X matrix
        for (i in 1:dim(X[[par]])[1]) {
            for (j in 1:dim(X[[par]])[2]) {
                X[[par]][i,j] <- (predictors[i] - x[par])^(j-1)
            }
        }

        # Weights
        weights <- c()
        for (i in 1:length(predictors)) {
            weights[i] <- pnorm((predictors[i] - x[par])/h)/h
        }
        weights <- diag(weights)

        # W_i^P (x[i])
        result[par] <- t(e_1) %*% solve(t(X[[par]]) %*% weights %*% X[[par]]) %*% t(X[[par]]) %*% weight
    }

    return(result)
}
```

b) Test the implementation by estimating the regression function in the location model $Y = m(X) + \epsilon$, where $m(x) = (x-1)^2$, $X \sim N(1,1)$, and $\epsilon \sim N(0, 0.5)$. Do it for a sample of size $n = 500$.

```r
# function to estimate
m = function(x) ((x-1)^2)

# bandwidth
h = 0.5

# data simulation
pred <- rnorm(500, mean=1, sd=1)
resp <- c()
for (i in 1:length(pred)) {
    resp <- c(resp, m(pred[i])) + rnorm(1,mean=0, sd=0.5)
}

# appending to list object
simulated_dataset <- matrix(,nrow=length(pred),ncol=2)
simulated_dataset[,1] <- pred
simulated_dataset[,2] <- resp

# Running the custom implementation
imp <- lce(x=rnorm(500, mean=1.1, sd=1), data=simulated_dataset, h=h)
```

```r
# grid to plot and test the accuracy of the implementation
x_grid <- seq(min(pred),max(pred), l=500)
plot(pred,resp)
lines(x_grid, m(x_grid), col=1)
lines(x_grid, lce(x=x_grid, data=simulated_dataset, h=h), col=2)
legend("top", legend = c("True reg","LocPoly (3rd deg)"), lwd=2, col=1:2)
```



Category C: Problem 4

- **Exercise 3.30**. Load the *ovals.RData* file.

a) Split the dataset into the training sample, comprised of the first 2,000 observations, and the test sample (rest of the sample). Plot the dataset with colors for its classes. What can you say about the classification problem?

b) Using the training sample, compute the plug-in bandwidth matrices for all the classes.

c) Use these plug-in bandwidths to perform kernel discriminant analysis.

d) Plot the contours of the kernel density estimator of each class and the classes partitions. Use coherent colors between contours and points.

e) Predict the class for the test sample and compare with the true classes. Then report the successful classification rate.

f) Compare the successful classification rate with the one given by LDA. Is it better than kernel discriminant analysis?

g) Repeat f with QDA.