Car-Accident-Kernel-Regression.R

Jim

2019-11-30

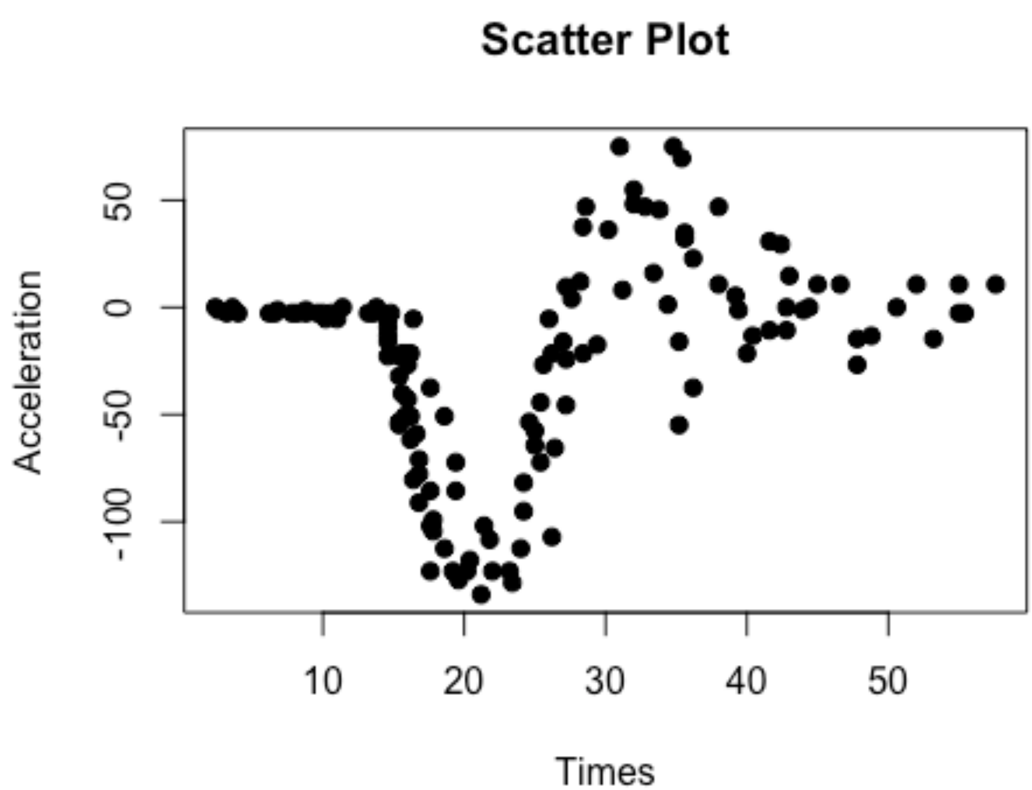
```
# Non-parametric Regression - Kernel Regression
# A data frame giving a series of measurements of head acceleration
# in a simulated motorcycle accident, used to test crash helmets.

# First, we should plot the data to see whether we should use Non-parametric
regression
# Clearly, we could see there is no linear relationship in this plot.
# So we use Kernel Regression for estimation.
library(MASS)
mcycle

##     times  accel
## 1     2.4    0.0
## 2     2.6   -1.3
## 3     3.2   -2.7
## 4     3.6    0.0
## 5     4.0   -2.7
## 6     6.2   -2.7
## 7     6.6   -2.7
## 8     6.8   -1.3
## 9     7.8   -2.7
## 10    8.2   -2.7
## 11    8.8   -1.3
## 12    8.8   -2.7
## 13    9.6   -2.7
## 14   10.0   -2.7
## 15   10.2   -5.4
## 16   10.6   -2.7
## 17   11.0   -5.4
## 18   11.4    0.0
## 19   13.2   -2.7
## 20   13.6   -2.7
## 21   13.8    0.0
## 22   14.6  -13.3
## 23   14.6   -5.4
## 24   14.6   -5.4
## 25   14.6   -9.3
## 26   14.6  -16.0
## 27   14.6  -22.8
## 28   14.8   -2.7
## 29   15.4  -22.8
## 30   15.4  -32.1
```

```
## 31   15.4  -53.5
## 32   15.4  -54.9
## 33   15.6  -40.2
## 34   15.6  -21.5
## 35   15.8  -21.5
## 36   15.8  -50.8
## 37   16.0  -42.9
## 38   16.0  -26.8
## 39   16.2  -21.5
## 40   16.2  -50.8
## 41   16.2  -61.7
## 42   16.4   -5.4
## 43   16.4  -80.4
## 44   16.6  -59.0
## 45   16.8  -71.0
## 46   16.8  -91.1
## 47   16.8  -77.7
## 48   17.6  -37.5
## 49   17.6  -85.6
## 50   17.6 -123.1
## 51   17.6 -101.9
## 52   17.8  -99.1
## 53   17.8 -104.4
## 54   18.6 -112.5
## 55   18.6  -50.8
## 56   19.2 -123.1
## 57   19.4  -85.6
## 58   19.4  -72.3
## 59   19.6 -127.2
## 60   20.2 -123.1
## 61   20.4 -117.9
## 62   21.2 -134.0
## 63   21.4 -101.9
## 64   21.8 -108.4
## 65   22.0 -123.1
## 66   23.2 -123.1
## 67   23.4 -128.5
## 68   24.0 -112.5
## 69   24.2  -95.1
## 70   24.2  -81.8
## 71   24.6  -53.5
## 72   25.0  -64.4
## 73   25.0  -57.6
## 74   25.4  -72.3
## 75   25.4  -44.3
## 76   25.6  -26.8
## 77   26.0   -5.4
## 78   26.2 -107.1
## 79   26.2  -21.5
## 80   26.4  -65.6
```

```
## 81   27.0  -16.0
## 82   27.2  -45.6
## 83   27.2  -24.2
## 84   27.2    9.5
## 85   27.6    4.0
## 86   28.2   12.0
## 87   28.4  -21.5
## 88   28.4   37.5
## 89   28.6   46.9
## 90   29.4  -17.4
## 91   30.2   36.2
## 92   31.0   75.0
## 93   31.2    8.1
## 94   32.0   54.9
## 95   32.0   48.2
## 96   32.8   46.9
## 97   33.4   16.0
## 98   33.8   45.6
## 99   34.4    1.3
## 100  34.8   75.0
## 101  35.2  -16.0
## 102  35.2  -54.9
## 103  35.4   69.6
## 104  35.6   34.8
## 105  35.6   32.1
## 106  36.2  -37.5
## 107  36.2   22.8
## 108  38.0   46.9
## 109  38.0   10.7
## 110  39.2    5.4
## 111  39.4   -1.3
## 112  40.0  -21.5
## 113  40.4  -13.3
## 114  41.6   30.8
## 115  41.6  -10.7
## 116  42.4   29.4
## 117  42.8    0.0
## 118  42.8  -10.7
## 119  43.0   14.7
## 120  44.0   -1.3
## 121  44.4    0.0
## 122  45.0   10.7
## 123  46.6   10.7
## 124  47.8  -26.8
## 125  47.8  -14.7
## 126  48.8  -13.3
## 127  50.6    0.0
## 128  52.0   10.7
## 129  53.2  -14.7
## 130  55.0   -2.7
```

```
## 131  55.0   10.7
## 132  55.4   -2.7
## 133  57.6   10.7
```

```
cycleDF <- as.data.frame(mcycle)
X <- cycleDF$times
Y <- cycleDF$accel
plot(X, Y, main = "Scatter Plot", xlab = "Times", ylab = "Acceleration", pch
= 19)
```

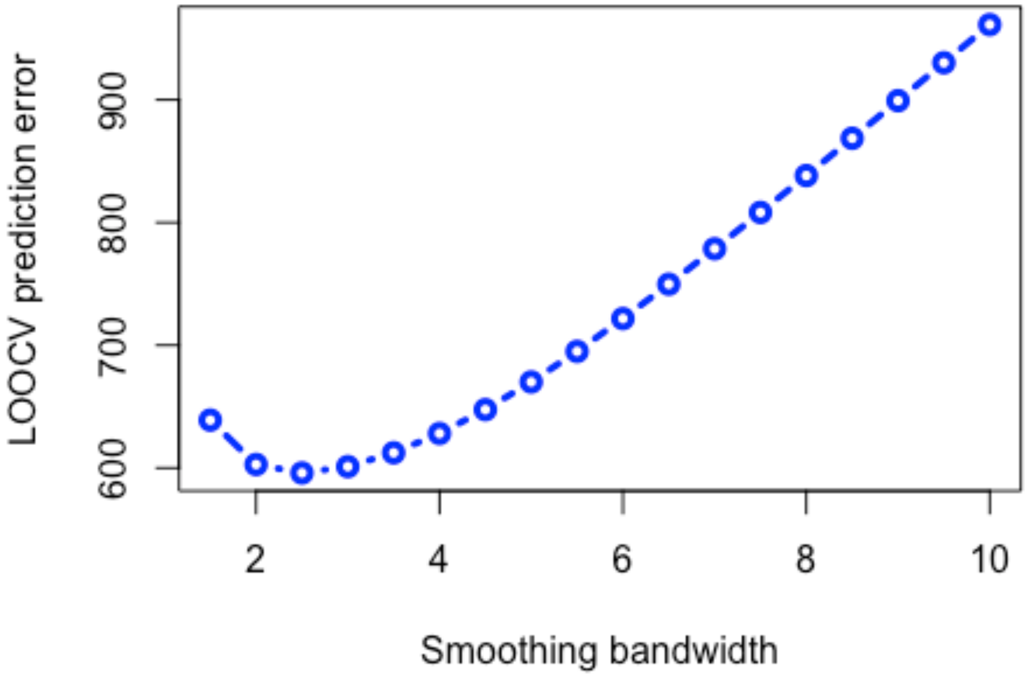



```
# Cross Validation
# Leave-one-out cross validation
n <- dim.data.frame(cycleDF)[1] # n: sample size

h_seq <- seq(from = 1.5,to = 10.0, by= 0.5)

# bandwidths we are using
CV_err_h <- rep(NA, length(h_seq))

for (j in 1:length(h_seq)){
  h_using <- h_seq[j]
  CV_err <- rep(NA, n)
  for (i in 1:n) {
```

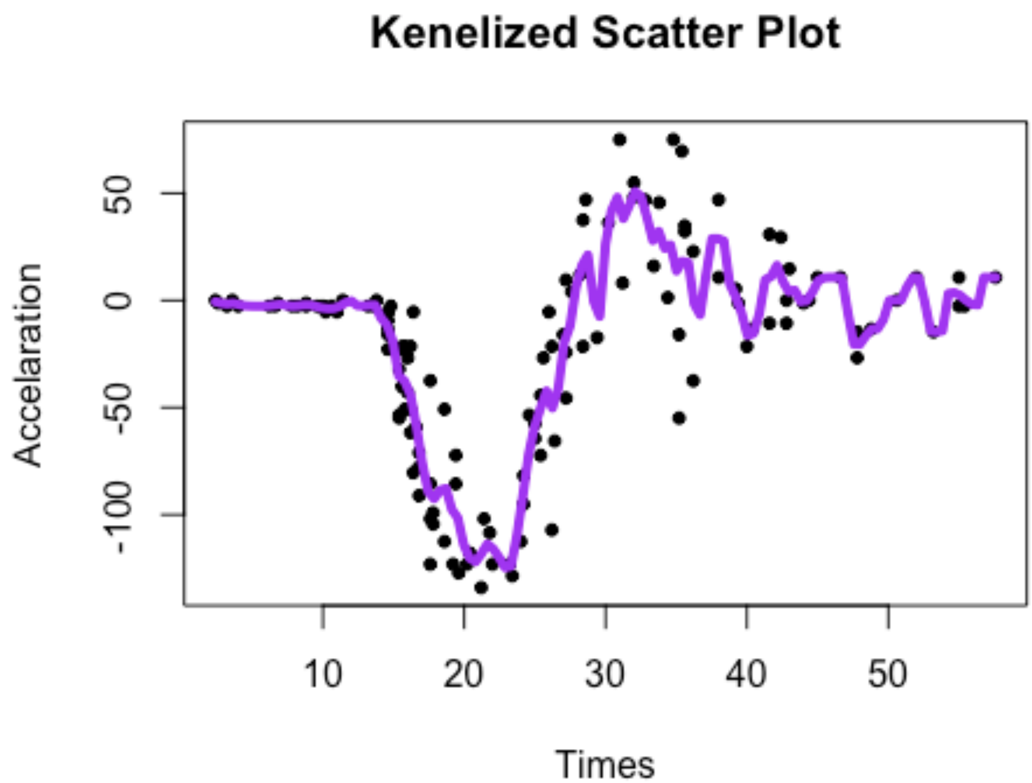
```r
    X_val <- X[i]
    Y_val <- Y[i]
    # Validation set
    X_tr <- X[-i]
    Y_tr <- Y[-i]
    # Training set
    Y_val_predict <- ksmooth(x = X_tr, y = Y_tr,
                            kernel = "normal", bandwidth = h_using, x.points
= X_val)
    CV_err[i] <- (Y_val - Y_val_predict$y)^2
  }
  CV_err_h[j] <- mean(CV_err)
}
CV_err_h
```

```
##  [1] 638.9920 602.8414 595.9586 601.1110 612.3813 628.1086 647.5628
670.0977
##  [9] 695.0158 721.7479 749.7548 778.6409 808.2178 838.2577 868.6444
899.2966
## [17] 930.1937 961.2848
```

```r
# Plot which bandwidth we will use
plot(x=h_seq, y=CV_err_h, type="b", lwd=3, col="blue", xlab="Smoothing
bandwidth", ylab="LOOCV prediction error")
```




```r
which(CV_err_h == min(CV_err_h))

## [1] 3

h_seq[which(CV_err_h == min(CV_err_h))]

## [1] 2.5

# We choose 2.5 as our bandwidth in this dataset
Kreg <- ksmooth(x=X,y=Y,kernel = "normal",bandwidth = 0.9)
plot(X,Y,pch=20, xlab = "Times", ylab = "Accelaration", main = "Kenelized
Scatter Plot")
lines(Kreg, lwd=4, col="purple")
```


Kenelized Scatter Plot
Accelaration
50
0
-50
-100
10
20
30
40
50
Times