Lab 04 - Nonparametric regression

[Sicily Xie #54385315]

2022-10-19

Brief description

In the lectures last week, you learned about the elastic net as a "hybrid" shrinkage method between the ridge regression and LASSO, as well as the concept of regression splines.

This week, we will perform spline regression using B-splines. The training and test data sets contain the head acceleration (`accel`, in g) as a function of time (`times`, in ms) after impact in a simulated motorcycle accident. The original data set is in the `MASS` package and a description can be found by typing `?MASS::mcycle`.

Questions

In spline regression, the number of knots K controls how adaptive the regression line is. We will compare three fits with $K = 5$, 10 and 20. First, read the training data using the following statement:

```
# ?MASS::mcycle
dat.tr <- readRDS("mcycle_train.RDS")
```

Then, create a vector of 5 knots using the following statement:

```
knot_maker <- function(k) {
  as.numeric(quantile(dat.tr$times, (1:k) / (k + 1)))
}
knot_maker(5)
```

```
## [1] 14.60000 16.73333 23.70000 27.53333 39.50000
```

Use similar statements to create the vectors for $K = 10$ and 20.

```
# some code
knot_maker(10)
```

```
##  [1] 10.27273 14.60000 15.65455 17.60000 21.58182 24.81818 26.78182 31.78182
##  [9] 37.67273 43.90909
```

```
knot_maker(20)
```

```
##  [1]  7.895238 10.580952 13.742857 14.600000 15.400000 15.800000 16.733333
##  [8] 17.600000 19.514286 22.457143 24.200000 25.342857 26.219048 27.533333
## [15] 30.771429 33.723810 36.285714 40.742857 43.523810 51.666667
```

Q1 Explain how are the locations of the knots determined. Are they equally spaced along the x-coordinate?

The locations of the knots are determined by the k/(k+1) quantile of the time in each iteration. They are not equally spaced along the x-coordinate.

Fit the cubic splines using the B-spline basis (`bs` function in package `splines`) defined by the knots you created above. Refer to the lecture R code last Thursday (Oct 6) on how you can use the `bs` function inside `lm`.

Note 1: We use cubic splines this time, and thus you do not need to specify the `degree` argument.

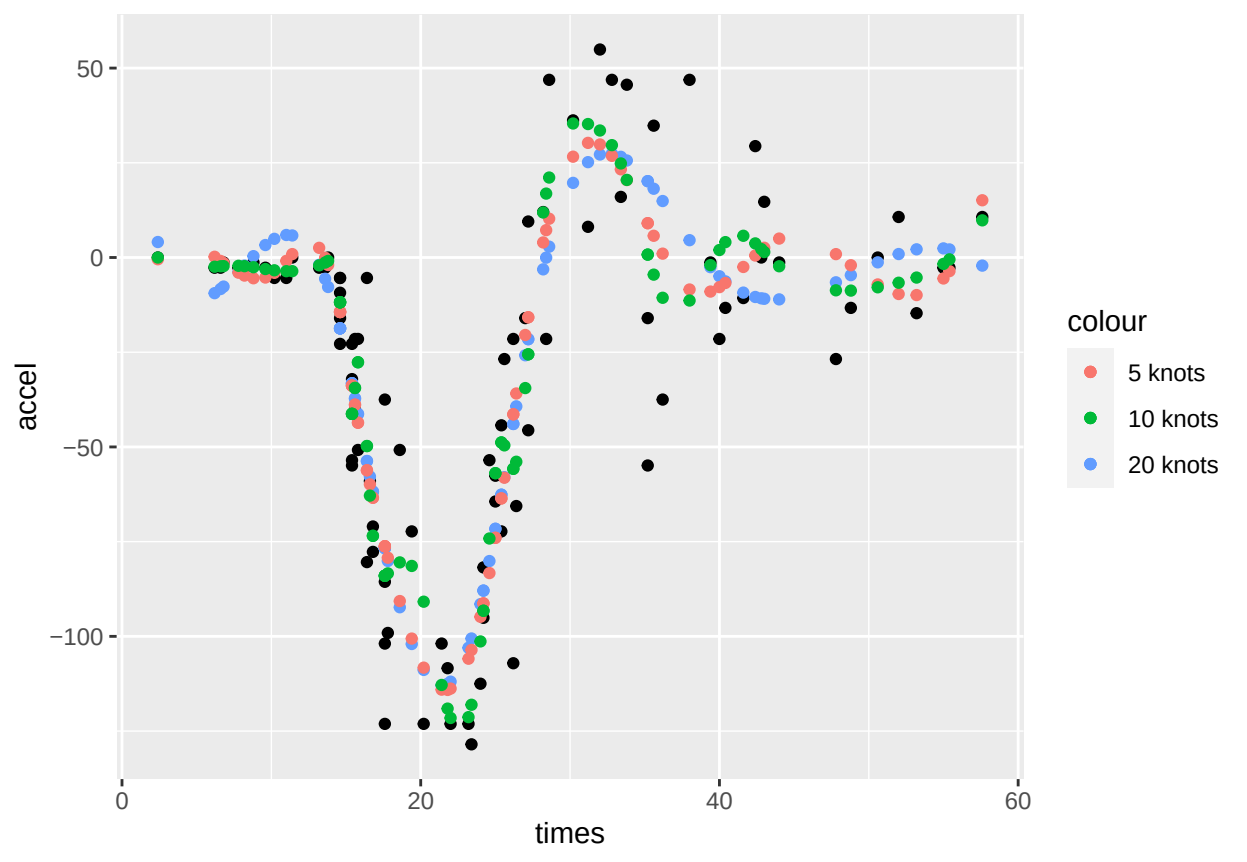
Note 2: You need to fit the three splines separately, each using one set of knots created above.

```
library(splines)
knot_5 <- knot_maker(5)
knot_10 <- knot_maker(10)
knot_20 <- knot_maker(20)
y <- dat.tr$accel
x <- dat.tr$times
knot_5_new <- lm(y ~ splines::bs(x,knots=knot_5))
knot_10_new <- lm(y ~ splines::bs(x,knots=knot_10))
knot_20_new <- lm(y ~ splines::bs(x,knots=knot_20))
```

Q2 Plot the resulting fits. Which one would you choose for prediction? Explain your choice. **Do not use the test set at this point**.

Hint: Use the `predict` function to get fitted values, either on the observed times or a sequence of user-specified times.

```r
# some code
library(ggplot2)
pred_5 <- predict(knot_5_new,newdata = dat.tr)
pred_10 <- predict(knot_10_new,newdata = dat.tr)
pred_20 <- predict(knot_20_new,newdata = dat.tr)
ggplot(data = dat.tr ) + geom_point(aes(x = times,y = accel))+
geom_point(aes(x = times,y = pred_5,color = "red")) +
geom_point(aes(x = times,y = pred_10,color = "blue")) +
geom_point(aes(x = times,y = pred_20,color = "green")) +
scale_color_hue(labels = c("5 knots", "10 knots", "20 knots"))
```



I would choose 20 knots for prediction since it is more close to the original data.

Q3 Now, use the test set to estimate the mean squared prediction errors of the three regression models. Which model appears to be the best?

```
test_data <- readRDS("mcycle_test.RDS")
library(splines)
temp1 <- lm(accel ~ splines::bs(times,knots = knot_5),
data = dat.tr)
temp2 <- lm(accel ~ splines::bs(times, knots = knot_10),
data = dat.tr)
temp3 <- lm(accel ~ splines::bs(times, knots = knot_20),
data = dat.tr)
mse <- function(x,y) mean((x-y)^2)
predict_5 <- predict(temp1,test_data)
predict_10 <- predict(temp2,test_data)
predict_20 <- predict(temp3,test_data)
mse_5 <- mse(test_data$accel, predict_5)
mse_10 <- mse(test_data$accel, predict_10)
mse_20 <- mse(test_data$accel, predict_20)
print(c(mse_5,mse_10,mse_20))
```

```
## [1] 511.6066 555.8892 699.8929
```

The model with knots = 5 is the best model since it has the smallest mse value of 511.6066.

Recall that a smoother is linear if the fitted values can be written as

$$\hat{\boldsymbol{y}} = \boldsymbol{S}\boldsymbol{y},$$

where $\boldsymbol{y}$ is the vector of response with fitted values $\hat{\boldsymbol{y}}$, and $\boldsymbol{S}$ is a matrix that does not depend on $\boldsymbol{y}$. For the linear smoother, the effective degrees of freedom is given by $\text{tr}(\boldsymbol{S})$.

Q4 Are the above smoothing splines linear smoothers?

Yes, it is because they are linear regression in a transformed space. They are linear in the transformed space of [x, x^2, x^3, . . .].

Q5 Is it possible to obtain the effective degrees of freedom for the three splines you have just fitted? If yes, write down the answer below. If not, explain why.

When knots = 5, the effective degree of freedom = 5 + 3 = 8. When knots = 10, the effective degree of freedom = 10 + 3 = 13. When knots = 20, the effective degree of freedom = 20 + 3 = 23.