HW04 Functional Data Analysis

Doug Nychka

2022-10-07

- You are encouraged to use web resources and class materials
- You can work in a group but hand in your assignment as an individual.
- Be spare in what you include and you will lose credit if you include too much extraneous output or information. All questions count for an equal number of points.
- Any subproblems marked GRAD are required for 500 level students but will serve as an extra credit question for the 400 level students.
- Please send me email if you have questions or any concerns. `nychka@mines.edu`
- Hand in your work in pdf format in Gradescope. You can keep the questions as part of what you hand in but you should begin your *answer* on a separate page. You can use `\newpage` to create a page break in your work.
- To comment out the answers without just deleting them use the html commenting format

```
<!--
This text is now commented out and will not be part of the rendered output.
-->
```

and check this Rmarkdown document for more details.

Points

All subsections of the problems count equally for 10 points:

- 400 level 1(30), 2(40)
- 500 level 1(40), 2(40)

Some setup

```
setwd("~/Dropbox/Home/Teaching/FDA/theCourse/Homework/HW04Smoothing")
suppressMessages(library( fields))
```

Problem 1

This problem will give you practice setting up and interpreting a Monte Carlo study of a statistical method. This one looks at how well the GCV criterion works in choosing a good data based smoothing parameter, lambda.

Refer back to the R script **4.3cubicSmoothingSpline.R** as an example. Even though the smoothing is controlled in a spline by the lambda parameter, note that throughout we looking at effective degrees of freedom as a more useful measure of the amount of smoothing rather than the lambda value itself. One can go back and forth between these two version in the same way one can look at a logged or unlogged value.

Given below is a breakdown of the steps (A-E) to do this. Some of you may already be familiar with this setup but it is detailed here for completeness.

Recall that a test function and data are created using the code

```
set.seed(123)
N <- 150
s <- runif(N)
s <- sort(s)
# asymetric bump function is the true curve
true <-  9 * s * (1 - s) ^ 3
errors<- .1 * rnorm(N)
y <-  true + errors
```

(A) *Before* the **for** loop create two arrays to hold some information and set the random seed (so the results are repeatable.)

```
EffDF<- rep( NA, 200)

MSE1<- rep( NA, 200)
MSE2<- rep( NA, 200)
```

(B) *Before* the **for** loop generate all of the synthetic data. This is useful if you want to go back and look at a particular sample. Here **Y** is a matrix where the columns index the 200 samples.

```
set.seed(498)
errors<-  matrix( .1 * rnorm(N*200), N, 200)
Y <-  true + errors
```

(C) Code the loop to have 200 repetitions – that is the for loop has the indexing/structure `for( k in 1:200) { #  analyze  each sample, Y[,k] }`

Within the for loop, for each of the generated datasets estimate the curve using a cubic smoothing spline with the function **Tps**. Do this in two ways, first finding the spline with fixed effective degrees of freedom at 5.0 and the other when this is estimated using GCV
E.g.

```
fitObjFixed<- Tps(s,Y[,k],df=5)
fitObjGCV<- Tps(s,Y[,k])
```

In the second case lambda (and the corresponding effective degrees of freedom) is being found by the GCV criterion.

(D) *Within* the for loop, save the estimated effective degrees of freedom and MSE for the curve estimate. Note that we can compute this because being a simulation we know what the true function is! For a real data set we do not know **true**.

```
EffDF[k]<- fitObjGCV$eff.df
fHatFixed<- predict( fitObj5)
fHatGCV<- predict( fitObjGCV)
MSE1[k]<- mean(  ( true - fHatFixed)^2 )
MSE2[k]<- mean(  ( true - fHatGCV)^2 )
```

(E) *After* the loop analyze the saved information (see questions below)

1(a)

Across the 200 samples, is there a difference between the the EffDF found by GCV and the fixed value at 5? Use a histogram and a vertical line at 5 to explain your results.

1(b)

Based on the mean squared error which is a more accurate estimate of the true curve, effDF=5 or GCV?

1(c)

Is there any dependence between the effective DF from GCV and the corresponding MSE. When does the estimated curve do poorly?

1(d) GRAD

Find the worst MSE out of the 200 samples and go back and look at scatterplot of the generated data and add the two estimated curves using df=5 and GCV. Are there any features that you see in the data to tip you off that GCV not working well?

Problem 2

Revisit the hourly Golden Ozone data for 2021 that you worked on for the take home.

2(a)

Create a matrix where each column indexes a day and the rows are the hourly values. Note that this is similar to the **Y** from Problem 1. Also to make this simpler omit all days that do not have a complete set of observations. The code below does this wrangling for you.

```
load("GOzone2021.rda")
library( lubridate)
O3<- matrix( NA, nrow=365, ncol=24)
O3[ cbind(GOzone2021$day, GOzone2021$hour+1) ]<- GOzone2021$O3
dim( O3)
```

```
## [1] 365  24
```

```
ind<-  colSums( is.na(O3)) ==0
O3<- O3[ ind,]
tday<- 1:365
tday<- tday[ind]
O3<- t( O3)
dim( O3)
```

```
## [1]  24 213
```

```
sHour<- 0:23
```

Plot these data as individual boxplots over the days of the year (**tday**). Note that you might have to tranpose the data back to rows with hours and columns of days to use the **boxplot** function.

2(b)

For each day smooth the 24 hourly measurements using a cubic smoothing spline. Find the GCV estimate of the effective degrees of freedom and summarize the results.

Tn searching for the GCV minimum if it is at either end of the DF range, 2 or n, the Tps function it will give a warning message. Don't worry about getting these except to note some of the searches are finding eff.df that are close to 24. (I see 7 cases with 22.8 being reported as the endpoint.)

2(c)

From your results in 2(b) identify a day where the effective degrees of freedom is less than 10 and plot the hourly measurements and add the GCV spline. Do the same for a case where the effective degrees of freedom is above 20. In either case, do the smooth curves seem reasonable?

2(d)

Find the average of the measurements for each hour across the different day resulting in a data vector of length 24. (for example `meanO3<- rowMeans(O3)`) Smooth these data with a cubic smoothing spline using GCV, create a scatter plot of the data and add the fitted curve. Compare the effective degrees of freedom found in this case to the ones above.

2(e) EXTRA CREDIT

When I work 2(d) I find eff.df is about 14.1. Return to 2(b) and use this value for all the days in the Tps function. For either case, GCV or df=14.1 save the value of the GCV criteria found for the curve. Compute the GCV criterion for either case "by hand" For example

```
look<- Tps(sHour, O3[,k])
n<- length( sHour)
GCV[k]<- mean( look$residuals^2)/ (1- look$eff.df/n)^2
```

Compare the GCV values for the fixed degrees of freedom at 14.1 to the GCV when it is minimized across all the days in the data set. How different are they? Does it suggest 14.1 might be a good fixed choice?