BIOS 7720
Homework 1
Due 04/01/2021

For all problems which involve simulation or fitting models to data, code to reproduce results must be included in your solutions. In addition, for problems 1-5, you may use *R* packages to help set up basis functions (e.g. *mgcv*), but estimation and smoothing parameter selection should be implemented manually. For these problems you may use either ordinary cross validation or generalized cross validation. For problems 7-8, students may use the *mgcv* package for model fitting and smoothing parameter selection.

Students are allowed to work together in groups of up to 3 on this assignment, though all students must submit their own independent write-ups. For coding problems, students may work together on designing and strategizing methods for implementation, but coding up of solutions must be done independently. Please include the name of any other students you worked with on this assignments in your submission.

P1. (10 points) Suppose we observe data of the form $\{(x_i, y_i); i = 1, \ldots, N\}$ and that we are interested in fitting the following model:

$$y_i = f(x_i) + \epsilon_i$$
$$\epsilon_i \sim N(0, \sigma^2_\epsilon)$$

where we model $f(x)$ using some basis expansion $f(x) = \sum_{k=1}^{K} \xi_k \phi_k(x)$. For the following problems, consider the penalized least squares criteria given by

$$\text{PENSSE}_\lambda = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

a. (2 points) Show that under this framework, $\lambda \int f''(x)^2 dx = \lambda \boldsymbol{\xi}^t \boldsymbol{S} \boldsymbol{\xi}$ where $\boldsymbol{S}$ is a $K \times K$ matrix and $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_k]^t$. State any necessary assumptions.

b. (1 points) Find $\boldsymbol{S}$ using the basis expansion $f(x) = \sum_{k=0}^{3} \xi_k x^k$ where integration is over the domain $x \in (0, 10)$.

c. (2 points) Show that the solution to the penalized least squares problem is $\hat{\boldsymbol{\xi}} = (\boldsymbol{\Phi}^t \boldsymbol{\Phi} + \lambda \boldsymbol{S})^{-1} \boldsymbol{\Phi}^t \boldsymbol{y}$ where $\boldsymbol{\Phi}$ is the $N \times K$ spline basis matrix associated with the observed covariate data, and $\boldsymbol{y} = [y_1, \ldots, y_N]^t$.

d. (2 points) Given the least squares solution $\hat{\boldsymbol{\xi}} = (\boldsymbol{\Phi}^t \boldsymbol{\Phi} + \lambda \boldsymbol{S})^{-1} \boldsymbol{\Phi}^t \boldsymbol{y}$, find $\text{Var}(\hat{f}(x))$ for a fixed λ.

e. (2 points) In lecture we discussed looping over candidate smoothing parameters λ and choosing the optimal λ based on leave-one-out cross-validated

prediction error. In practice, the procedure shown in the lecture notes can be quite slow for large K (relative to N).

We can potentially speed this procedure up and increase numerical stability up by creating an augmented design matrix ($\boldsymbol{\Phi}^*$) and response vector ($\boldsymbol{y}^*$):

$$\boldsymbol{y}^* = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{a} \end{bmatrix}, \qquad \boldsymbol{\Phi}^* = \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{B} \end{bmatrix}$$

and using these augmented data to obtain $\hat{\boldsymbol{\xi}}$ via linear regression (using, e.g. .lm.fit). This allows us to take advantage of methods for estimating linear regression via orthogonal decomposition. Find $\boldsymbol{a}$ and $\boldsymbol{B}$. Hint: note that $\boldsymbol{S}$ can be written as $\boldsymbol{D}^t\boldsymbol{D}$.

f. (1 point) Why don't we optimize the least squares criteria jointly for $\boldsymbol{\xi}$ and λ?

P2. (10 points) Write a function which takes as inputs a response vector $\boldsymbol{y}$, a spline basis matrix $\boldsymbol{\Phi}$, a penalty matrix $\boldsymbol{S}$, and a candidate smoothing parameter λ (under the notation from problem 1 above) and obtains both point and variance estimates for $\boldsymbol{\xi}$ (conditional on the smoothing parameter). Using the data simulated by the code below, apply your function to find the optimal smoothing parameter using cross-validation as your smoothing parameter selection criteria.

```
set.seed(5520)
N <- 100
f <- function(x) cos(2*pi*x)
x <- runif(N, min=-1, max=1)
y <- f(x) + rnorm(N, mean=0, sd=0.5)
```

You may use the code from the lecture 2 notes (*mgcv::smoothCon()*) to set up your basis matrix $\boldsymbol{\Phi}$ and obtain $\boldsymbol{S}$ using either B-splines or cubic regression splines.

P3. (10 points) Thus far we have discussed estimating spline coefficients using penalized least squares. However, because of our distributional assumptions on the residual term ($\epsilon_i \sim N(0, \sigma^2_\epsilon)$), for a fixed smoothing parameter λ, we can obtain point and uncertainty estimates on spline coefficients using maximum likelihood. Using an optimization routine (e.g. the *optim()* function in *R*):

a. (5 points) Obtain the maximum (penalized) likelihood estimate for $\hat{\boldsymbol{\xi}}$ and var($\hat{\boldsymbol{\xi}}$) for the data simulated in problem 2 above by adding the penalty term $-0.5\lambda\boldsymbol{\xi}^t\boldsymbol{S}\boldsymbol{\xi}$ to the standard log likelihood for your data (presented below) at the optimal value of λ you found in problem 2 (i.e. no need to do a search for the optimal λ, just use the one you found from the previous problem).

b. (5 points) Use your results for (a) to compare the estimated $\hat{f}(x)$ and $\hat{f}(x) \pm 2\text{SE}(\hat{f}(x))$ for $x \in (-1, 1)$ obtained using maximum likelihood to those from problem 2 (penalized least squares) at the optimal value of λ you found in problem 2.

Note that the penalized log likelihood (pl) referred to in part (a) take the form

$$pl(\boldsymbol{\xi}, \sigma^2; \boldsymbol{y}, \boldsymbol{x}) = l(\boldsymbol{\xi}, \sigma^2; \boldsymbol{y}, \boldsymbol{x}) - \frac{1}{2}\lambda \boldsymbol{\xi}^t \boldsymbol{S} \boldsymbol{\xi}$$

where $l(\boldsymbol{\xi}, \sigma^2; \boldsymbol{y}, \boldsymbol{x})$ is the log likelihood, $\boldsymbol{y} = [y_1, \ldots, y_N]^t$ are the response data and $\boldsymbol{x} = [x_1, \ldots, x_N]^t$ are the covariate data.

Hint(s): 1) by default the *optim()* function performs minimization (as opposed to maximization); 2) optimize on the log-likelihood scale; 3) using the *optim()* function, set the argument hessian = TRUE to obtain the Hessian matrix.

P4. (10 points) This question considers the polynomial basis expansion $f(x) = \sum_{k=0}^{K} \xi_k \phi_k(x) = \sum_{k=0}^{K} \xi_k x^k$. Suppose we observe we observe data of the form $\{(x_{i1}, x_{i2}, y_{i1}, y_{i2}); i = 1, \ldots, N\}$ where the data are generated as follows:

$$y_{ip} = f_p(x_{ip}) + \epsilon_{ip}$$

for $p = 1, 2$ where $x_{i1} \sim \text{Uniform}(0, 1)$, $x_{i2} \sim \text{Exponential}(\lambda = 0.01)$, $\epsilon_{i1} \sim N(0, \sigma^2 = 0.5^2)$, $\epsilon_{i2} \sim N(0, \sigma^2 = 0.5^2)$, and $f_p(x) = a_p x^2$ where $a_1 = 5, a_2 = 1 \times 10^{-5}$. Simulate data for $N = 1000$ using the code below and use these simulated data to answer questions (a)-(d) below.

```
set.seed(19870)
N <- 100
f  <- function(x, a) a*x^2
x1 <- runif(N, 0, 1)
y1 <- f(x1, a=5) + rnorm(N, sd=0.5)
x2 <- rexp(N, rate=1/100)
y2 <- f(x2, a=1e-05) + rnorm(N, sd=0.5)
```

a. (2 points) Set up the polynomial basis for $K = 5$ for both covariate vectors $\boldsymbol{x}_1 = [x_{11}, \ldots, x_{N1}]^t$ and $\boldsymbol{x}_2 = [x_{12}, \ldots, x_{N2}]^t$ and provide the code to do so.

b. (2 points) Estimate the coefficient vectors $\boldsymbol{\xi}_p = [\xi_{1p}, \ldots, \xi_{Kp}]^t$ for $p = 1, 2$ using unpenalized least squares.

c. (2 points) Estimate the coefficient vectors $\boldsymbol{\xi}_p = [\xi_{1p}, \ldots, \xi_{Kp}]^t$ for $p = 1, 2$ using penalized least squares where the smoothing parameter is selected using cross validation.

d. (2 points) Plot both unpenalized and penalized estimates of $\hat{f}_p(x)$ along with $\hat{f}_p(x) + / - 2\text{SE}(\hat{f}_p(x))$ over the range of observed x values.

e. (2 points) Suppose we modified the polynomial basis to be of the form $f(x) = \sum_{k=1}^{K} \xi_k \phi_k(x) = \sum_{k=2}^{K} \xi_k x^k$. What would happen to the estimated function $\hat{f}(x)$ as λ approaches ∞?

P5. (10 points) Design and implement a simulation study which simulates data according to the model:

$$y_i = f(x_i) + \epsilon_i$$
$$\epsilon_i \sim N(0, \sigma^2_\epsilon)$$

Using either a cubic regression spline or B-spline basis for $f(x)$, assess the bias, mean squared error, and coverage probability of $\hat{f}(x)$ estimated using both penalized and unpenalized least squares for various choices of number of observations (N) and association structure (f) and number of basis functions (K). At least three combinations of N, f, and K must be included in the simulation study (at least $3 \times 3 \times 3 = 27$ total scenarios). Summarize your results and support your claims using either tables or figures.

P6. (10 points) This question relates to the identifiability constraints imposed by the *mgcv* package in *R*. Suppose we have a full rank spline basis matrix $\boldsymbol{\Phi}_{N \times K}$ which contains the constant vector ($\boldsymbol{c}_{N \times 1}$) in it's column space. Show that column mean centering $\boldsymbol{\Phi}$ results in a new basis matrix $\boldsymbol{\Phi}^*$ which does not contain the constant vector $\boldsymbol{c}$ in it's column space.

P7. (10 points) Thus far we have assumed independent observations. This problem investigates the behavior of automatic smoothing parameter methods when data are correlated. Here, we will take this idea to the extreme where the correlation between observations is 1. To do so, we will be investigating the association between age and average total daily volume of physical activity measured via wearable hip worn accelerometers in the NHANES 2003-2006 accelerometry cohort.

Specifically, we will calculate average daily total volume of physical activity within individuals, and then fit our model to two datasets. The first dataset will contain duplicated rows (duplicated (x_i, y_i) pairs). The second dataset will contain a "correct" dataset with one row per subject. The code to set up the two datasets is below.

```
library("readr"); library("dplyr"); library("mgcv")
df       <- read_rds("NHANES_AC_processed.rds")
min_cols <- paste0("MIN",1:1440)
df$TAC   <- rowSums(df[,min_cols], na.rm=TRUE)
df_fit   <-
    df %>%
    filter(good_day == 1, !is.na(Age)) %>%
    group_by(SEQN) %>%
    mutate(TAC_mn = mean(TAC))
df_full <- df_fit
df_ind  <- df_fit[!duplicated(df_fit$SEQN),]
```

a. (5 points) Fit the additive model $\bar{\text{TAC}}_i = f(\text{Age}_i) + \epsilon_i$ where $\bar{\text{TAC}}_i$ is the subject average daily total activity count (TAC_mn in the code above) using both the full dataset with repeated observations ("df_full") and the dataset with one row per subject ("df_ind"). Use *mgcv::gam()* for model estimation with smoothing parameter selection done by generalized cross validation. Plot the estimated average TAC as a function of age plus/minus two standard errors for each fit. Comment on any differences you see.

b. (5 points) Choose the optimal λ using leave-one-subject-out cross validation fit on the full dataset with repeated observations ("df_full") and plot the estimated average TAC as a function of age plus/minus two standard errors. Comment on any differences you see with the results from part (a) and hypothesize on the reasons for any observed differences.

P8. (10 points) This problem investigates REML versus generalized cross-validation as a method for smoothing parameter selection. Using the same NHANES dataset from the previous problem, calculate the total daily activity counts (TAC) for each subject-day, filter out "bad days" (days with fewer than 10 hours of estimated wear time or individuals with poor quality data) using the "good_day" column, remove individuals with missing age data, and then obtain subject-specific average TAC. The code to do so is presented below.

```
library("readr"); library("dplyr"); library("mgcv")
df       <- read_rds("NHANES_AC_processed.rds")
min_cols <- paste0("MIN",1:1440)
df$TAC   <- rowSums(df[,min_cols], na.rm=TRUE)
df_ind   <-
    df %>%
    dplyr::select(-one_of(min_cols)) %>%
    filter(good_day == 1, !is.na(Age)) %>%
    group_by(SEQN) %>%
    summarize(Age=Age[1], TAC_mn = mean(TAC))
```

a. (5 points) Fit the additive model $\text{TAC}_i = f(\text{Age}_i) + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2_\epsilon)$ to the full data using penalized cubic regression splines with $K = 50$ using both REML and GCV for smoothing parameter selection. The code for model fitting can be found below. Using the fitted models, plot the estimated average TAC plus/minus two standard errors over the range of observed ages.

```
fit_GCV  <- gam(TAC_mn ~ s(Age, k=50, bs="cr"), method="GCV.Cp", data=df_ind)
fit_REML <- gam(TAC_mn ~ s(Age, k=50, bs="cr"), method="REML", data=df_ind)
```

b. (5 points) Using these data, obtain 1000 randomly selected sub-samples for each of $N \in \{100, 500, 1000, 2000\}$. Fit the additive model from the previous sub-question for each of these re-sampled datasets using both REML and GCV as smoothing parameter selection criteria. Compare the shapes of the estimated functions $\hat{f}$. Does either method tend to yield smoother versus

more wiggly estimated functions? How does the comparison change with sample size? Support your claims using numeric summaries of the estimated functions.