Homework 2
STATS205 (Fall 2019–2020)

Please structure your writeups hierarchically: convey the overall plan before diving into details. You should justify with words why something's true (by algebra, convexity, etc.). There's no need to step through a long sequence of trivial algebraic operations. Be careful not to mix assumptions with things which are derived. **Up to two additional points will awarded for especially well-organized and elegant solutions.**

Due date: Wed, Oct 23th, 2019, 11pm.

1. Local linear regressions, splines, and cross-validation (*9 points*)

You will be asked to implement the local linear regression and cubic splines on various datasets. The goal is to help you understand the similarity and difference between these methods and the bias-variance tradeoff effect of changing the bandwidth parameter h or the smoothing parameter λ.

Download (A) the data on fragments of glass collected in forensic work from the book website for Exercise 5.3 and (B) the motorcycle data from the book website for Exercise 5.4. For dataset (A), the range of x that we consider is $[0.29, 3.50]$. For dataset (B), the range of x we consider is $[2.4, 57.6]$. When you are asked to plot a function, you can discretize the interval with $N = 200$ evenly-spaced x's for plotting.

a. (*3 points*) (local linear regression) Recall that the local linear regression is a linear smoother: the estimator $\hat{r}(x)$ can be written as

$$\hat{r}(x) = \sum_{j=1}^{n} \ell_j(x) Y_j \tag{1}$$

where

$$\ell_j(x) = \frac{b_j(x)}{\sum_{i=1}^{n} b_i(x)}, \tag{2}$$

$$b_j(x) = w_j \left(\left(\sum_{i=1}^{n} w_i u_i^2 \right) - u_j \sum_{i=1}^{n} w_i u_i \right) \tag{3}$$

where $u_j = x_j - x$ and $w_j = K\left(\frac{u_j}{h}\right)$.

Implement the the local linear regression estimator with the Gaussian kernel.

1. (1.5 point) Apply your local linear regression estimator to the dataset **(A)** with bandwidth $h \in \{0.02, 0.15, 1\}$. Plot the fitted $\hat{r}(x)$ as a function of x. In the same figure, also plot the data points. (In the figure, there should be three curves, a scatter graph of the data points, a proper legend and title.) Briefly **discuss** which one of these fitted curves intuitively is better and why.

2. (1.5 point) Download the dataset (C) from http://web.stanford.edu/class/stats205/a2data.csv. The dataset was generated synthetically by adding noises to the ground-truth $r(x) = x$. The covariate has column name 'x' and the response variable has column name 'y'. Apply your local linear regression estimator to it with bandwidth $h = 0.46$, and apply the kernel estimator with Gaussian kernel and bandwidth $h = 0.05$ to the dataset (C) as well. Plot the fitted $\hat{r}(x)$ as a function of x for $x \in [0, 1]$. In the same figure, also plot the data points and the ground-truth function $r(x) = x$ for references. (In the figure, there should be three curves, a scatter graph of the data points, and a proper legend and title.) Briefly **discuss** which one of these fitted curves is intuitively better and why. Note: the bandwidths we provided are the best based on the cross-validation.

b. (*2 points*) (Cross-validation) Recall that in the lecture, we have shown that we can use the following formula to compute the leave-one-out cross validation score for bandwidth h

$$\hat{R}_h = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{r}(x_i)}{1-\ell_i(x_i)}\right)^2 \tag{4}$$

where $\ell_i(x_i)$ is the i-th smoothing coefficient when evaluating at x_i.

1. (1 point) Compute the $\hat{R}_h$ for linear local regression with dataset (**B**) for $h \in \{k \cdot 10^{-2} : 150 \le k \le 200, k \text{ is integer}\}$. Plot the value of $\hat{R}_h$ as a function of h. You are supposed to observe a unimodal function. Briefly discuss why the theory in the class explains the shape.

2. (1 point) Compute and report the h that achieves the smallest cross-validation score (among the set of choices given above.). Plot the corresponding fitted curve $\hat{r}(x)$ for the optimal h, along with the data points.

c. (*4 points*) (Cubic splines) You will be asked to implement the smoothing cubic splines using an existing package. (Note that for other homework questions, unless explicitly mentioned otherwise, you are supposed to implement your algorithms from scratch.)

Note: this question was updated on Oct 13rd due to an error and unclarity in the previous version. If you have solved the problem with python, then likely you don't need to do anything. If you used R to solve the problem before, very likely you need to regenerate the plots with different parameters.

For python users, there seems to be no handy package in python that does exactly the penalized regression fit of cubic splines covered in the lecture and the textbook. We suggest you to use `scipy.interpolate.UnivariateSpline` as a surrogate. The argument **s** of the function controls the smoothing of the curve (the bigger **s** is, the more smoothing).

For R users, we suggest `smooth.spline` {`stats`} where the **spar** argument is related to the λ in the textbook or the lecture (the bigger **spar** is, the bigger λ is, and the more smoothing).

You only need to choose one language and one package to do this homework. Because the two packages implement different versions of cubic splines, the results also don't match each other.

Note: The spline smoothing algorithm requires that the inputs/covariants $x_1, \ldots, x_n$ to be strictly increasing, but there may be datapoints (x, y) with identical x. To use the packages, you need to sort by values of x and somehow remove the duplicates. For simplicity, you should merge the data points with the same x values into a single data point, by averaging the corresponding response variables. (E.g., if $x_i = x_j = z$, then you should merge them into a new data point $(x, y) = (z, \frac{x_i+x_j}{2})$.)

1. (1.5 point) Apply smoothing splines on the dataset (A) with different smoothing parameters. Users of `scipy` function in python should use $s = 10, 250, 500, 1000$. Users of `smooth.spline` in R should use $spar = 0.01, 0.2, 0.5, 1.1$. Please plot the corresponding fitted $\hat{r}(x)$. As usual, also plot the scatter graph of the data points in the same figure. (You are supposed to see a bias-variance trade-off as you vary the choice of the smoothing parameters **s** or **spar**.)

2. (2.5 points) Download the dataset D from http://web.stanford.edu/class/stats205/c2data.csv, where covariate has column name 'x' and response variable has column name 'y'. The dataset was generated by the course staff by using the ground-truth function $r(x) = \begin{cases} \cos(12\pi x) & 0 \le x < 1 \\ x & 1 \le x \le 2 \end{cases}$.

   - (0 point) Plot the true function $r(x)$ for reference.
   - (0.5 point) Apply smoothing cubic splines using $s = 30$ for python or $spar = 0.35$ for R. Plot the fitted $\hat{r}(x)$ in the same figure.

– (0.5 point) Apply the local linear regression with $h = 0.015$. In the same figure, plot the fitted $\hat{r}(x)$. Also plot the data points in the same figure.

– (1.5 points) Qualitatively compare the two methods. Which one of the two curves is better? Try yourself with a few other smoothing parameters (either **s** or **spar**) for cubic splines and h for the local linear regression, and compare the curves visually. (You are supposed to see the choice above by the course staff is reasonably fine.) Discuss why it is difficult to find a good h in the local linear regression to have a smooth fit. (Note: this question is somewhat open-ended, and may not have a unique answer. Hint: the data are heterogeneous.)