CSE 544, Spring 2020, Probability and Statistics for Data Science

Assignment 3: Non-Parametric Inference Due: 3/4, in class

(7 questions, 75 points total)

I/We understand and agree to the following:

(a) Academic dishonesty will result in an 'F' grade and referral to the Academic Judiciary.

(b) Late submission, beyond the 'due' date/time, will result in a score of 0 on this assignment.

(write down the name of all collaborating students on the line below)

---

1. MSE in terms of bias **(Total 5 points)**

For some estimator $\hat{\theta}$, show that MSE = bias$^2(\hat{\theta})$ + Var$(\hat{\theta})$. Show your steps clearly.

2. Programming fun with $\hat{F}$ (Total 17 points)

For this question, we require some programming; you should only use Python. You may use the scripts provided on the class website as templates. Do not use any libraries or functions to bypass the programming effort. Please submit your code in the google form (will be announced) with sufficient documentation so the code can be evaluated. Attach each plot as a separate sheet to your submission. All plots must be neat, legible (large fonts), with appropriate legends, axis labels, titles, etc.

(a) Write a program to plot $\hat{F}$ (empirical CDF or eCDF) given a list of samples as input. Your plot must have y-limits from 0 to 1, and x-limits from 0 to the largest sample. Show the input points as crosses on the x-axis. (3 points)

(b) Use an integer random number generator with range [1, 99] to draw n=10, 100, and 1000 samples. Feed these as input to (a) to generate three plots. What do you observe? (2 points)

(c) Modify (a) above so that it takes as input a collection of list of samples; that is, a 2-D array of sorts where each row is a list of samples (as in (a)). The program should now compute the average $\hat{F}$ across the rows and plot it. That is, first compute the $\hat{F}$ for each row (student), then average them all out across rows, and plot the average $\widehat{F}$. Show all input points as crosses on the x-axis. (3 points)

(d) Use the same integer random number generator from (b) to draw n=10 samples for m=10, 100, 1000 rows. Feed these as input to (d) to generate three plots. What do you observe? (2 points)

(e) Modify the program from (a) to now also add 95% Normal-based CI lines for $\hat{F}$, given a list of samples as input. Draw a plot showing $\hat{F}$ and the CI lines for the q2.dat data file (799 samples) on the class website. Use x-limits of 0 to 2, and y-limits of 0 to 1. (3 points)

(f) Modify the program from (e) to also add 95% DKW-based CI lines for $\hat{F}$. Draw a single plot showing $\hat{F}$ and both sets of CI lines (Normal and DKW) for the q2.dat data. Which CI is tighter? (4 points)

3. Plug-in estimates **(Total 10 points)**

(a) Show that the plug-in estimator of the variance of X is $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2$, where $\bar{X}_n$ is the sample mean, $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. (3 points)

(b) Show that the bias of $\hat{\sigma}^2$ is $-\sigma^2/n$, where σ^2 is the true variance. (4 points)

(c) The kurtosis for a RV X with mean μ and variance σ^2 is defined as $Kurt[X] = E[(X-\mu)^4] \,/\, \sigma^4$. Derive the plug-in estimate of the kurtosis in terms of the sample data. (3 points)

4. Consistency of eCDF **(Total 10 points)**

Let D={X_1, X_2, …, X_n} be a set of i.i.d. samples with true CDF *F*. Let $\hat{F}$ be the eCDF for D, as defined in class.

(a) Derive E($\hat{F}$) in terms of *F*. Start by writing the expression for $\hat{F}$ at some α. (3 points)

(b) Show that bias($\hat{F}$) = 0. (2 points)

(c) Derive se($\hat{F}$) in terms of *F* and n. (3 points)

(d) Show that $\hat{F}$ is a consistent estimator. (2 points)

5. Histogram estimator **(Total 13 points)**

Histogram is a representation of sample data grouped by bins. Consider a true distribution X with range [0, 1). Let m ∈ Z^+ and b < 1 be such that m · b = 1, where m is the number of bins and b is the bin size. Bin i, denoted as B_i, where 1 ≤ i ≤ m, contains all data samples that lie in the range $[\frac{(i-1)}{m}, \frac{i}{m})$.

(a) Let p_i denote the probability that the true distribution lies in B_i. As in class, derive $\widehat{p_i}$ in terms of indicator RVs of i.i.d. data samples (the X_i) drawn from the true distribution, X. (3 points)

(b) The histogram estimator for some $x \in [0, 1)$ is defined as $\widehat{h}(x) = {\widehat{p_J}}/{b}$, where $x \in B_j$. Show that $E[\hat{h}(x)] = f(x)$ when b→0, where f(x) is the true pdf of X. (4 points)

(c) Use all of the weather.dat data on the class website and plot its histogram estimate (that is, plot the $\widehat{h}(x) = \frac{\widehat{p_j}}{b} = \hat{p}_j \; \forall \; x \in B_j$) using python with a bin size of 1. Do not use any in-built libraries to bypass the programming effort. Use the same instructions as in Q2 for legibility and format of plot submissions. Submit your code via the google form, labeled as q5.py. (3 points)

(d) Now use the histogram estimator ($\widehat{h}(x) = \frac{\widehat{p_j}}{b} \; \forall \; x \in B_j; \; b = 1$) as an estimate of pdf based on the weather.dat dataset. Based on these pdf estimates, plot the CDF of the dataset using python. Attach the plot to your hardcopy submission. (3 points)

6. Properties of estimators **(Total 5 points)**

Find the bias, se, and MSE in terms of θ for $\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} X_i$, where X_i are i.i.d. $\sim$ Poisson(θ). Show your work. Hint: Follow the same steps as in class, assuming the true distribution is unknown. Only at the very end use the fact that the unknown distribution is Poisson(θ) to get the final answers in terms of θ.

7. Kernel density estimation (Total 15 points)

As usual, submit all code for this Q on the google form. The histogram density estimation has several drawbacks such as discontinuities in the estimate and density estimate depends on the starting of the bin. To alleviate these short coming to a certain extent, we will use another type of non-parametric density estimation technique called Kernel density estimation (KDE). The formal definition of KDE is :

For a data sample $D = \{X_1, X_2, \dots, X_n\}$. The KDE of any point x is given by:

$$\hat{p}_{KDE}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)$$

where $K(.)$ is called the kernel function which should be a smooth, symmetric and a valid density function. Parameter $h > 0$ is called the smoothing bandwidth that controls the amount of smoothing.

(a) *Density Estimation*: Generate a sample of 800 data points $D = \{X_1, X_2, \dots, X_{800}\}$ which are i.i.d. and sampled from a Mixture of Normal distributions such that with prob. 0.25, it is Nor(0,1), with prob. 0.25, it is Nor(3,1), with prob. 0.25 it is Nor(6,1), and with remaining prob. 0.25, it is Nor(9,1). Note that this is the **true distribution**. A simple way to sample data from this distribution is to sample a RV $X \sim U[0,1]$, if $X \leq 0.25$ sample from the 1st Normal (that is, Nor(0,1)), if $X \in (0.25, 0.5]$ then sample from 2nd Normal (that is, Nor(3,1)), and so on. Now obtain the KDE estimate of the PDF $\hat{p}_{KDE}(\alpha)$ for $\alpha \in \{-5, -4.9, -4.8, \dots, 10\}$ (use **`np.arange`**(-5, 10, 0.1)) using Parzen window kernel, where the density estimate is defined by

$\hat{p}_{KDE}(\alpha) = \frac{1}{nh}\sum_{i=1}^{500} I\{|\alpha - x_i| \leq \frac{h}{2}\}$, where I() is the indicator RV.

Write a python function which takes as input (a) Data (D) and (b) Smoothing bandwidth (h), and returns a list of KDE estimates for all the points in the list **`np.arange`**(-5, 10, 0.1)). Using this function, generate plots (in the same figure) of the KDE estimate of the PDF for all the values of $\alpha \in \{-5, -4.9, -4.8, \dots, 10\}$ for the values of $h \in \{.1, 1, 7\}$ along with the true PDF of α; note that the true distribution is the mixture of Normals stated above. To numerically get the pdf of a given Normal in python, try scipy.stats.norm.pdf. The master plot should have on x-axis the alpha values, ranging from -5 to 10. You should have 4 lines: one for each h value and one for the true distribution. Make sure to have a useful legend. What are your observations regarding the effect of h on KDE estimate $\hat{p}_{KDE}(\alpha)$? (8 points)

(b) *Bias and Variance:* Now we will study the effect of parameter h on the bias and variance of the KDE estimates. Repeat the trial of generating 800 data points 150 times in the same way as above. Let each row represent a trial (so you should have a matrix with 150 rows, and each row (trial) should have 800 columns). Let $\hat{p}^i{}_{KDE}(\alpha)$ = KDE estimate at α for the i^{th} trial and let $p(\alpha)$ be the true pdf at α. Then the expectation, bias, Var and MSE are given by:

$E[\hat{p}_{KDE}(\alpha)] = \frac{1}{150}\sum_{j=1}^{150} \hat{p}^i{}_{KDE}(\alpha)$,

$Var\left(\hat{p}_{KDE}(\alpha)\right) = \frac{1}{150}\sum_{i=1}^{150}\left[\hat{p}^i{}_{KDE}(\alpha) - E[\hat{p}_{KDE}(\alpha)]\right]^2$,

$Bias\left(\hat{p}_{KDE}(\alpha)\right) = \left(\frac{1}{150}\sum_{j=1}^{150} \hat{p}^i{}_{KDE}(\alpha)\right) - p(\alpha)$,

$MSE\left(\hat{p}_{KDE}(\alpha)\right) = Var\left(\hat{p}_{KDE}(\alpha)\right) + Bias^2\left(\hat{p}_{KDE}(\alpha)\right)$.

To observe the effect of h on the bias and variance, first calculate the total bias and variance (across all points) as $Bias_{tot}^2(h) = \frac{\sum_{\alpha \in S} Bias^2(\hat{p}_{KDE}(\alpha))}{|S|}$ and $Var_{tot}(h) = \frac{\sum_{\alpha \in S} Var(\hat{p}_{KDE}(\alpha_i))}{|S|}$,

where $S = \{-5, -4.9, -4.8, \dots, 10\}$ is the set of points for which you are estimating the density. Write python code to solve the following questions:

(i) For each value of $h \in \{0.01, .1, .3, .6, 1, 3, 7\}$ calculate the bias and variance as defined above and generate two plots, one for $Bias_{tot}^2(h)$ vs h and another for $Var_{tot}(h)$ vs h. What do you observe from these plots? (5 points)

(ii) If we use MSE as a measure to select the optimal h, i.e., $h^* = argmax_h(Var_{tot}(h) + Bias_{tot}^2(h))$, what is the optimal value of h you should use? (2 points)