School of Mathematics and Statistics
MAST90083: Computational Statistics and Data Science
Assignment 2

Due date: No later than 14:00 on Monday 7th October 2024

Weight: 20%

Question 1 Nonparametric Regression

This question deals with the nonparametric estimation of f using n observations $\{(x_i, y_i), i = 1, \cdots, n\}$ generated from

$$y_i = f(x_i) + \epsilon_i, x_i \in [0, 1], y_i \in \mathcal{R},$$

where the regression function f has a sharp change and our objective besides the estimation of f is to estimate the position of the sharp change in the function f. The simulated observations are stored in the `simudata.csv` file.

1. Provide the form of the local constant estimator (polynomials of order zero estimator also known as the Nadaraya-Watson estimator). Create this local constant estimator function in R and apply it using the Gaussian kernel (Lecture 4, slide 13) with $h = 0.05$ and $h = 0.1$ to the simulated data. Plot the fitted curve with the observations. What do you observe? Are you able to identify the sharp change? What's the characteristic of the local-constant estimator with Gaussian kernel? (Hint: first use `function` to create a local constant estimator R function, with `kernel` as one of the input arguments. This way, you can define different kernel functions separately, and simply change the `kernel` argument to address questions 1.1 to 1.3.)

2. Instead of using the Gaussian kernel, here we use the kernel function defined as follows

   $$K_1(x) = 1.5(1 - x^2)\mathbf{I}[x \in (0, 1)],$$

   Repeat 1, plot the fitted curve obtained from the local constant estimator with kernel $K_1(x)$ and the observations. What do you observe? What's the characteristic of the local-constant estimator with kernel $K_1(x)$?

3. Repeat 1 using the following kernel function

   $$K_2(x) = 1.5(1 - x^2)\mathbf{I}[x \in (-1, 0)],$$

   plot the fitted curve obtained from the local constant estimator with kernel $K_2(x)$ and the observations. What do you observe? What's the characteristic of the local-constant estimator with kernel $K_2(x)$?

4. Using $K_1(x), K_2(x)$ and the local constant estimator derive an estimator of the sharp change and its position. Illustrate the results of the estimation using the simulated observations $\{(x_i, y_i), i = 1, \cdots, n\}$.

Question 2 Spline Regression

In this question, we first explore the relationship between the truncated spline basis and the B-spline basis (Lecture 5, slides 45-46) in questions 2.1 to 2.3. Following this, we will apply penalized spline regression (Lecture 5, slides 30-32) in questions 2.4 and 2.5.

1. Let T_1, T_2, T_3 be three functions on $[0, 1]$ given by

   $$T_1(x) = 1, \quad T_2(x) = x, \quad T_3(x) = (x - \frac{1}{2})_+.$$

   Also, let B_1, B_2, B_3 be three functions on $[0, 1]$ given by

   $$B_1(x) = (1 - 2x)_+, \quad B_2(x) = 1 - |2x - 1|, \quad B_3(x) = (2x - 1)_+.$$

   Plot T_i and $B_i, i = 1, 2, 3$ in two plots, each plot will show three functions. Give the expressions for B_1, B_2, B_3 in terms of T_1, T_2, T_3. (Hint: you can first find what is $B_1 + B_2 + B_3$).

2. Find the 3×3 matrix $\mathbf{L}_{TB}$ (Lecture 5, slide 46) such that

   $$[B_1(x) \quad B_2(x) \quad B_3(x)] = [T_1(x) \quad T_2(x) \quad T_3(x)]\mathbf{L}_{TB}$$

   for any $x \in [0, 1]$. Compute the determinant of $\mathbf{L}_{TB}$ and establish that $\mathbf{L}_{TB}$ is invertible.

3. The above two questions imply that $\{B_1, B_2, B_3\}$ is an alternative basis for the vector space of functions spanned by $\{T_1, T_2, T_3\}$. It's known as the linear B-spline basis, and has better numerical properties that the truncated line basis $\{T_1, T_2, T_3\}$ (Lecture 5, slides 45-46). Let's generate a set of predictor values $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}, n = 100$ using `sort(runif(n))` function in R and the response $\mathbf{y}$ =`cos(2*pi*x)+0.2*rnorm(n)`. Let

   $$\mathbf{X}_T \equiv \begin{bmatrix} T_1(x_1) & T_2(x_1) & T_3(x_1) \\ \vdots & \vdots & \vdots \\ T_1(x_n) & T_2(x_n) & T_3(x_n) \end{bmatrix}_{n\times 3} \quad \text{and} \quad \mathbf{X}_B \equiv \begin{bmatrix} B_1(x_1) & B_2(x_1) & B_3(x_1) \\ \vdots & \vdots & \vdots \\ B_1(x_n) & B_2(x_n) & B_3(x_n) \end{bmatrix}_{n\times 3}$$

   be design matrix for the two bases. For each of these two design matrices, we perform the ordinary least square regression by doing the following two tasks.

   (a) Show that the fitted values $\hat{\mathbf{y}}_B$ with design matrix $\mathbf{X}_B$ are identical with the fitted values $\hat{\mathbf{y}}_T$ with design matrix $\mathbf{X}_T$.

   (b) Plot the fitted values $\hat{\mathbf{y}}_B, \hat{\mathbf{y}}_T$ together with the data in the same figure to verify your proof in (a).

4. We use the same $\mathbf{x}, \mathbf{y}$ generated in question 2.3. Now let's consider the following linear spline regression model

   $$y_i = \beta_0 + \beta_1 x_i + \sum_{K=1}^{20} u_K (x_i - \kappa_K)_+ + \varepsilon_i, i = 1, 2, \cdots, n \tag{1}$$

   where $\kappa_1, \cdots, \kappa_{20}$ are equally spaced knots over the range of the data, $\boldsymbol{\theta} = \{\beta_0, \beta_1, u_1, \cdots, u_{20}\}$ are the unknown coefficients.

(a) Write the matrix form of model (1) and define your matrices.

(b) Now use the least squares method to fit this linear spline regression model (1). Provide a plot that includes both the data and the fitted curve.

(c) How does the fitted results look? Is there evidence of overfitting?

5. To improve the performance of the linear spline regression model (1), we can add a penalty term. Here we consider using the penalized least squares, given by

$$\text{minimize}\left[\sum_{i=1}^{n}\left(y_i-\beta_0-\beta_1 x_i-\sum_{K=1}^{20}u_K(x_i-\kappa_K)_+\right)^2+\lambda\sum_{K=1}^{20}u_K^2\right]$$

(a) Provide the matrix form of the penalized spline fitting criterion, derive the expression of the penalized least square estimates of the unknown coefficients $\boldsymbol{\theta}$ and the associated expression of fitted values.

(b) Next, set $\lambda = 0.0001, 1, 100$ and generate a plot that includes the data along with the fitted curves for each of these three λ values. What do you observe from these three plots?

Question 3: Expectation-Maximization Algorithm

For this question, you are not allowed to use R built in function for the expectation-maximization (EM) algorithm. You have already learnt how to code EM algorithm for a Gaussian mixture consisting of two probability density functions (pdfs) in Tutorial 6, where both densities had the same standard deviation. Here, you will extend that algorithm to three cases, where all three have different standard deviations.

1. In order to load relevant libraries and the provided image that is needed for the subsequent questions, use the following code

```
library("plot.matrix")
library("png")
library("fields")
I <- readPNG("CM.png")
I=I[,,1]
I=t(apply(I, 2, rev))
par(mfrow=c(2,1))
image(I, col = gray((0:255)/255))
plot(density(I))
```

2. The above code will generate Figure 1 and as you can see from the density plot of the cameraman picture (Figure 1b) that it is a mixture of three Gaussian distributions with different standard deviations. Therefore, extend the EM algorithm so that it can estimate the parameters for these three pdfs. The following three instructions can help you complete this question.

   - You may consider reshaping the image data matrix $\mathbf{I}$ to one dimensional vector before applying your algorithm.

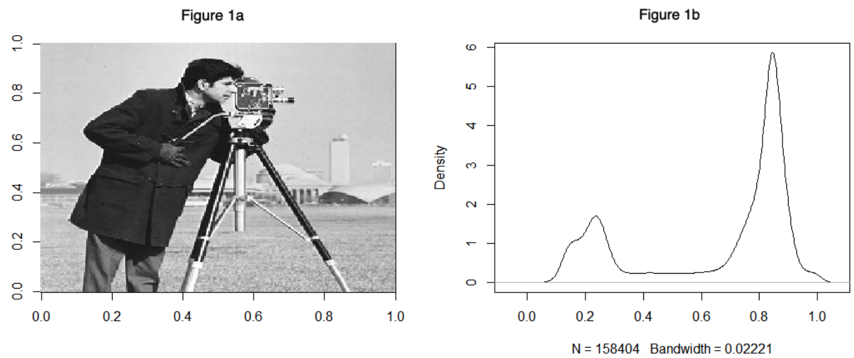

Figure 1: Cameraman picture and its density plot

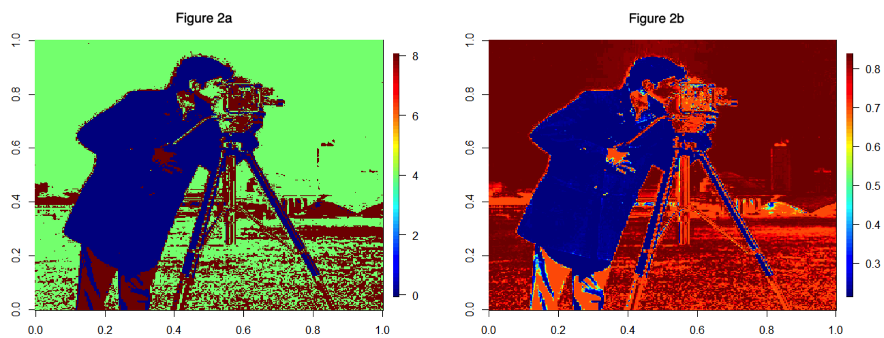

Figure 2: Cameraman picture labeling

- You must terminate the algorithm when the stopping criteria $|e_j - e_{j-1}|$ becomes less than 10^{-6}, where $e_j = \sum_{i=1}^{3}(m_i^{(j)} - m_i^{(j-1)}) + \sum_{i=1}^{3}(c_i^{(j)} - c_i^{(j-1)})$. Here, e_j and e_{j-1} correspond to the values from the current and previous iteration, respectively. $m_i^{(j)}$ and $c_i^{(j)}$ represent i-th mean and associated standard deviation at iteration j, whereas $m_i^{(j-1)}$ and $c_i^{(j-1)}$ represent i-th mean and associated standard deviation at iteration $j-1$.
- Use command `print(round(c(m, c, p),4))` to print all 9 values in each iteration, where p_i (the i-th entry of vector **p**) is a mixing probability for the i-th distribution and denoted as $\hat{\pi}$ in the Lecture 6, slide 24.

3. Using the results of EM algorithm in 2, label the image pixels by first using each pixel's pdf estimation (you must be able to read the assignment value from Figure 2a, you may want to use `dnorm` function for this part) and second using the posterior pdfs multiplied with their mean.

4. Plot both images obtained from question 3.3 and compare the results with Figure 2. You must obtain an exact match. You may want to use function `image.plot` for better visualization.