Statistical Machine Learning: Coursework 3

General Instructions

- Work in groups of at least one and at most two.
- Include your student number and your name on your assignment.
- In your solutions, you should present your R output or snippets of R code as you deem appropriate. Make sure to present your results clearly.
- You should submit your work as a single PDF file on Blackboard by Thursday the 27^{th} February at 12 noon.

Additional Information

- There are **40 marks** available in total. Your mark out of 40 will be scaled by 2.5 to give the mark out of 100 that you see on Blackboard.

Problem 1: Theory (20 marks)

In this half of the assignment, we shall investigate a regression method known as *K-Nearest-Neighbours (K-NN) Regression*. To do so, we shall assume that we have training data $(X_j, Y_j)_{j=1}^N$, which are i.i.d. instances of the following random variables:

$$X \sim \text{Unif}([0,1]^p), \quad Y = f(X) + \varepsilon,$$

where $f : [0,1]^p \to \mathbb{R}$ is an unknown, continuous function that we aim to estimate, and the noise term ε is mean zero, variance σ^2 and independent of X. For convenience, we denote the training dataset as $T := \{(X_j, Y_j)\}_{j=1}^N$ which can be viewed as a random variable taking values in $\mathbb{R}^{N\times(p+1)}$.

For $K \in \{1, ..., N\}$, the K-NN regression estimator for f is given by:

$$\hat{f}_N(X) = \frac{1}{K}\sum_{i=1}^{K} Y_{(i)},$$

where $X_{(i)} \in (X_j)_{j=1}^N$ is the i^{th} closest training point to X and $Y_{(i)}$ is the observation of Y corresponding to $X_{(i)}$. In other words, $(\cdot) : \{1, ..., N\} \to \{1, ..., N\}$ is a permutation satisfying;

$$||X - X_{(1)}|| \leq ||X - X_{(2)}|| \leq ... \leq ||X - X_{(N)}||,$$

and $Y_{(i)}$ is defined as $Y_{(i)} = f(X_{(i)}) + \varepsilon_{(i)}$. For ease, you may assume that the ordering $X_{(1)}, ..., X_{(N)}$ is uniquely defined. To simplify matters, we shall assume that $K = 1$, so that $\hat{f}_N(X) = Y_{(1)}$. Our goal is to understand the behavior of the test error of $1-$NN regression as the training data sample size tends to infinity.

(*i*) **(3 marks)** Show that for any $\delta > 0$:

$$\mathbb{E}_X\left[\mathbb{1}[\|X_{(1)} - X\| \leq \delta] \;\middle|\; T\right] =$$
$$\int_{x\in[0,1]^p} \mathbb{1}\Big[||x - X_1|| \leq \delta \;\vee\; \ldots \;\vee\; ||x - X_N|| \leq \delta\Big]dx$$

where $\mathbb{1}[\cdot]$ is an indicator function which evaluates to 1 if the statement inside the brackets occurs, and 0 otherwise, and $\vee$ is the "logical or" operator.

(*ii*) **(5 marks)** Show that for any $\delta > 0$:

$$\lim_{N\to\infty} \mathbb{E}_T\left[\int_{x\in[0,1]^p} \mathbb{1}\Big[||x - X_1|| \leq \delta \;\vee\; \ldots \;\vee\; ||x - X_N|| \leq \delta\Big]dx\right] = 1.$$

You may assume, without proof, that the expectation and integral above can be exchanged freely. Additionally, assume the following property holds: there exists a constant $C \in (0, 1]$ such that, for any $x \in [0, 1]^p$:

$$\mathrm{Vol}(B_\delta(x) \cap [0, 1]^p) \geq C,$$

where $B_\delta(x) = \{y \in \mathbb{R}^p : ||y - x|| \leq \delta\}$ is the p-dimensional ball of radius $\delta > 0$ centered $x \in \mathbb{R}^p$ and $\mathrm{Vol}(A) = \int_{\mathbb{R}^p} \mathbb{1}[x \in A]dx$ is the p-dimensional volume of a measurable set A.

(*iii*) **(5 marks)** Using the results of parts (*i*) and (*ii*), show that:

$$\lim_{N\to\infty} \mathbb{E}_T\left[\mathbb{E}_X\left[\big(f(X_{(1)}) - f(X)\big)^2 \;\middle|\; T\right]\right] = 0.$$

Hint: Consider the logical statement $A_N := \{||X_{(1)} - X|| \leq \delta\}$ *and try to bound:*

$$\mathbb{E}_X\left[\big(f(X_{(1)}) - f(X)\big)^2 \mathbb{1}\big[A_N\big] \;\middle|\; T\right] \quad \text{and}$$
$$\mathbb{E}_X\left[\big(f(X_{(1)}) - f(X)\big)^2 \mathbb{1}\big[\neg A_N\big] \;\middle|\; T\right],$$

where $\neg$ *is logical negation. In your answer, you may need to employ some concepts from analysis such as "uniform continuity" and "boundedness of a function".*

(*iv*) **(4 marks)** Define the test error for this problem as:

$$\mathrm{Err}_{\mathrm{Test}} := \mathbb{E}_{(X,Y)}\left[\big(Y - \hat{f}_N(X)\big)^2 \;\middle|\; T\right].$$

Using the limit given in part (*iii*) show that $\mathbb{E}_T[\text{Err}_{\text{Test}}] \to 2\sigma^2$ as $N \to \infty$. *Hint: Note that* $\mathbb{E}_T[\epsilon_{(1)}] = 0$ *and* $\text{Var}_T[\epsilon_{(1)}] = \sigma^2$. *That is, choosing the "nearest neighbour",* $X_{(1)}$*, does not affect the noise distribution.*

(*v*) **(3 marks)** Interpret your answer to part (*iv*) using the bias-variance decomposition for Err_{Test} discussed in Lecture 13. What can you say about the behaviour of the KNN estimator $\hat{f}_N$ as $N \to \infty$? *Your answer should be no more than 3 sentences.*

Problem 2: Practical (20 marks)

In this section, we shall focus on the Tree-structured Parzen Estimator (TPE) algorithm for hyperparameter tuning. The dataset we shall use for this task, `Salary_Data.csv`, can be found on Blackboard, alongside this document.

The data consists of 6,699 observations related to employment salaries in India, compiled from multiple sources, including surveys, job postings, and publicly available data. The goal is to model `salary` using the other variables in the dataset, in order to understand which factors significantly impact earnings. The dataset includes the following variables:

- `Salary`: Monthly salary of the employee,
- `Age`: Age of the employee,
- `Gender`: Gender of the employee,
- `Education Level`: Employee's highest level of education,
- `Job Title`: Employee's role or designation,
- `Years of Experience`: Total professional experience,
- `Commute Distance`: Distance traveled for work,
- `Gym Membership`: Binary variable representing whether the employee has a gym membership,
- `Coffee Consumption`: Average cups of coffee consumed per day,
- `Hobbies Score`: A numerical score reflecting personal hobbies,
- `Survey Response`: Numeric measure of responses to a workplace survey.

Some variables in the dataset have been *synthetically generated* and do not come from real-world sources. These variables are uncorrelated with salary and should ideally be excluded from the model. For this reason, we will use LASSO regression to drop less relevant variables from the model. Your task will be to use TPE to estimate the LASSO regularisation parameter λ.

Note: The dataset has already been preprocessed, including rescaling and one-hot encoding where necessary. We do not expect you to do any further preprocessing for this task.

(i) **(2 marks)** Write code to read in the dataset `Salary_Data.csv` and perform a train-validation-test split.

(ii) **(3 marks)** Write a function `my_loss(lambda, train_data, valid_data)` which does the following:

- Takes as input:
  - A regularization parameter λ (`lambda`),
  - Training data (`train_data`),
  - Validation data (`valid_data`).
- Fits a LASSO regression model using the training data.
- Computes and returns the Mean Squared Error (MSE) on the validation set, given by:

$$\mathcal{L}(\lambda) := \frac{1}{L}\sum_{i=1}^{L}\left(Y_{\text{Val},i} - X_{\text{Val},i}^{\top}\hat{\beta}_{\lambda}^{\text{LASSO}}\right)^2,$$

where $Y_{\text{Val},i} \in \mathbb{R}$ is the i^{th} response variable in the validation set, $X_{\text{Val},i} \in \mathbb{R}^p$ is the vector of covariates for the i^{th} validation data point, L is the number of validation data points, and p is the number of covariates.

You may use standard packages to perform LASSO regression.

(iii) **(2 marks)** Write a function `sample_lambdas(N)` that generates N values of λ, sampled uniformly from the interval $[0, 1]$.

You may use standard packages for random number generation.

(iv) **(3 marks)** Write a function `my_kde(z_values, x_values)` that performs Kernel Density Estimation (KDE) on a set of observations. The function should:

- Take as input:
  - An array of observed values (`z_values`) which will be used to construct the KDE.
  - An array of x-values (`x_values`) which are points at which the KDE should be evaluated.
- Return an array where each element corresponds to the estimated kernel density $K(x)$ at a given x in `x_values`.

You should use a bandwidth of 0.1 for computing the kernel density estimates.

For this question, you should not use built-in KDE functions such as `density`.

(v) **(2 marks)** Sample 100 values, $\{z_1, ..., z_{100}\} \sim \text{Uniform}(0, 1)$, and use your `my_kde` function to plot the Kernel Density Estimate for this sample.

(vi) **(6 marks)** Using your functions from *(ii)*, *(iii)*, and *(iv)*, implement the Tree-structured Parzen Estimator (TPE) algorithm given on Slide 22 of Lecture 15. Your function should:

- Start with an initial λ^* and compute its loss z^* on the validation set.
- Perform an iterative procedure that does the following in each iteration:
  - Sample 100 values of λ from the range $[0, 1]$.
  - Approximate the distributions $g(\lambda)$ and $b(\lambda)$ using KDE.
  - Choose a new λ^* such that

$$\lambda^* = \arg\min_{\lambda} \frac{b(\lambda)}{g(\lambda)}.$$

- Return the best λ^* found.

Any iterations in which fewer than two values are observed for $z > z^*$ or $z \leq z^*$ should be skipped over, and the iterative procedure should stop when the number of iterations exceeds 100.

You must not use pre-built TPE implementations from existing libraries.

(vii) **(2 marks)** Use your TPE implementation from (vi) to find the optimal λ^* for LASSO regression on the salary dataset. Evaluate the performance of the final model by computing the test MSE, given by;

$$\frac{1}{M} \sum_{i=1}^{M} \left(Y_{\text{Test},i} - X_{\text{Test},i}^{\top} \hat{\beta}_{\lambda^*}^{\text{LASSO}} \right)^2,$$

where $Y_{\text{Test},i} \in \mathbb{R}$ is the i^{th} response variable in the test set, $X_{\text{Test},i} \in \mathbb{R}^p$ is the vector of covariates for the i^{th} test data point, and M is the number of test data points.

Your answer must print the final value of λ^* and the test MSE.