Instruction of Programming Homework 1

Introduction of Machine Learning

Lecturer: Prof. Peggy Lu, peggylu@cs.ccu.edu.tw
Due date: 2025/4/10 23:59

Submission

Each student must work independently. Please upload a file called `HW1_StudentID_Name.zip` to ecourse2:

This file should contain:

- `HW1_StudentID_Name.pdf`: A PDF file containing the solution to all the exercises. This file must clearly indicate:
  - The problem number and question to which the solution refers.
  - Answers to all questions of the 2 problems assigned.
- Subfolders `Hw1-1` and `Hw1-2`: These should contain the corresponding Python code and figures for each problem.

  Running one of the **main** files should automatically generate all necessary figures and `.txt` files.

Note

- You should write your full name in your .pdf presentation.
- Do NOT write code in the .pdf file. Provide it in other place.

Problem 1

TA: David Lee, david20010603@alum.ccu.edu.tw

Tasks

Given a number of input-output data pairs, we seek to estimate the mapping function between the input and output data. Consider specifically the datasets `x.txt` and `y1.txt`, which you could download from the course's site under *moodle*. Apply polynomial curve fitting to estimate the mapping between the input set x and the output set $y1$ in the following cases:

(a) The order of the polynomial is $m = 2$ for $\{x, y1\}$. Please write down the values of the parameters of the polynomial w_0, w_1, w_2.

(b) Choose an appropriate order of the polynomial m so that fitting is successful.

(c) Consider two different polynomial orders $m = 3$ and $m = 8$. Estimate the average sum of squared errors for the estimation in each case.

Note

For questions 1 and 2, students must **also provide two figures** (one for each question) showing:

- The dataset points.
- The obtained curve with different colors.
- Implement polynomial curve fitting(without using built-in Python libraries).

Problem 2

TA: Oscar Wu, oscarwu217@alum.ccu.edu.tw

Tasks

Considering the dataset called *Dataset_2.txt*, which consists of a matrix of $N \times 3$ points, shown in Figure 1.

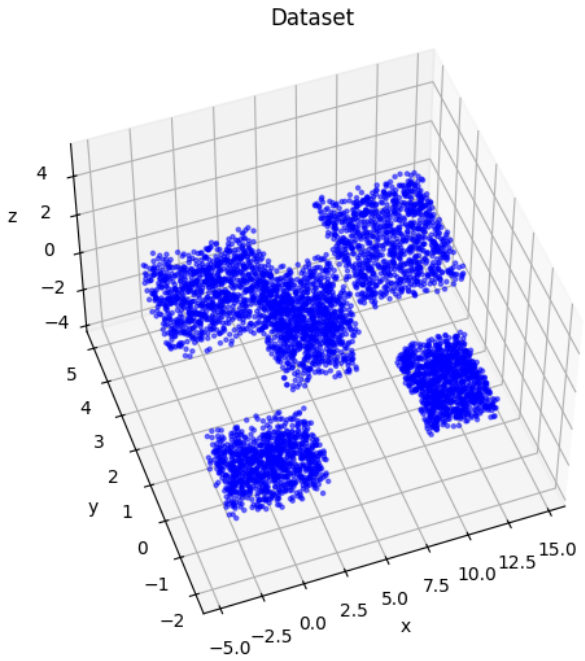

Figure 1: Dataset used in Problem 2.

To implement K-means (without using built-in Python libraries), and the Non-Uniform Binary Split algorithm, address the following questions:

(a) Classify the dataset points using the K-means ($K = 5$) algorithm. Use the Euclidean distance as the distortion function and initialize the centroids as follows:

$$y_1 = m_1 = \begin{bmatrix} 4.0 \\ -0.5 \\ 2.0 \end{bmatrix}, \quad y_2 = m_2 = \begin{bmatrix} 2.0 \\ 2.5 \\ 1.0 \end{bmatrix}, \quad y_3 = m_3 = \begin{bmatrix} 10.0 \\ 2.0 \\ -1.0 \end{bmatrix}$$

$$y_4 = m_4 = \begin{bmatrix} 7.0 \\ 0.5 \\ -0.5 \end{bmatrix}, \quad y_5 = m_5 = \begin{bmatrix} 12.0 \\ 1.0 \\ -1.0 \end{bmatrix}$$

After each iteration, (1) display the clustering result and (2) plot the distortion function values after each E-step and M-step.

(b) Analyze the behavior of the K-means ($K = 5$) algorithm when using the following initial centroids:

$$y_1 = m_1 = \begin{bmatrix} 0.0 \\ -0.3 \\ -2.0 \end{bmatrix}, \quad y_2 = m_2 = \begin{bmatrix} -1.3 \\ 1.5 \\ 4.0 \end{bmatrix}, \quad y_3 = m_3 = \begin{bmatrix} 11.3 \\ 3.0 \\ 0.2 \end{bmatrix}$$

$$y_4 = m_4 = \begin{bmatrix} 5.7 \\ 3.0 \\ -2.0 \end{bmatrix}, \quad y_5 = m_5 = \begin{bmatrix} 10.0 \\ -1.0 \\ 1.2 \end{bmatrix}$$

After each iteration, (1) display the clustering result and (2) plot the distortion function values after each E-step and M-step.

(c) Classify the dataset points using the Non-Uniform Binary Split algorithm. Compare the results with those obtained from the K-means approach. After each iteration, display the clustering result.

Notes

Cluster	1	2	3	4	5
Color	blue	black	red	green	magenta

Table 1: Mapping between clusters and colours

- You **cannot** use built-in Python libraries such as `sklearn.cluster.KMeans` or `scipy.cluster.vq.kmeans`.
- Submit Python code along with three figures, one corresponding to each sub-question ((a), (b), and (c)). Each figure should visualize the dataset, with points belonging to different clusters represented using distinct colors. Ensure that the colors match the cluster assignments as specified in Table 1.