Assignment 3: Maximum Likelihood Estimation

Instructions: This assignment engages you in the practical application of maximum likelihood estimation for linear regression modeling. Follow the steps carefully, and submit your work as a single PDF file that includes all required code and graphs. **Attach** supplementary code files (.py and .ipynb) as used in this assignment.

Objective: Learn how to fit a linear regression model using maximum likelihood estimation and understand the effects of model adjustments, such as adding a bias term.

Step 1: Graphing the Relationship

```python
import numpy as np
import scipy.linalg
import matplotlib.pyplot as plt

X = np.array([-3, -1, 0.0, 1, 3]).reshape(-1,1) # 5x1 vector, N=5, D=1
y = np.array([2.2, 3.7, 3.14, 3.67, 4.67]).reshape(-1,1) # 5x1 vector
```

- Using the provided code, plot the relationship between variables x and y.
- Attach the Python code used to generate the graph in your PDF.
- Display the graph you obtain.

Step 2: Applying the Derived Equation

$$\theta_{ML} = (X^\top X)^{-1} X^\top y$$

- Implement the final equation we derived in class to compute the parameter Theta (θ).
  - Refer to the Python file attached with this assignment for the implementation guide.
- Display the resulting Theta value and attach the code used to obtain it.
- Attach the Python code used in your PDF.

Step 3: Plotting the Initial Model

- Apply the Theta obtained from Step 2 to a linear equation.
- Plot this equation on the same graph as in Step 1.
- Ensure the graph displays the original 5 data points and the linear model representing the obtained Theta.
- Attach the Python code used to generate the graph in your PDF.

Step 4: Improving the Model

- You will notice that the linear model does not fit well with the training points provided. This is because we did not include the bias of the linear equation when calculating Theta initially.
- Enhance your model by including a bias term. Stack a column of ones vertically with your **x** data (as shown in the provided code snippet).

```python
# ===============================================================================
print("="*10)
N, D = X.shape
X_aug = np.hstack([np.ones((N,1)), X]) # augmented training inputs of size N x (D+1)
print("X_aug = ")
print(X_aug)
print("="*10)
# ===============================================================================
```

-

```
==========
X_aug =
[[ 1. -3.]
 [ 1. -1.]
 [ 1.  0.]
 [ 1.  1.]
 [ 1.  3.]]
==========
```

- Recalculate Theta using this new matrix configuration.
- The resulting Theta should now have two values (shape (2,1)).
- Attach your code and display both values of Theta.
- Explain how these values differ from those obtained in Step 2.

Step 5: Plotting the Improved Model

- Use the new Theta values from Step 4 in your linear equation.
- Plot this updated model on the same graph as in Step 1.
- The graph should still display the original 5 data points, along with the new linear model.
- Attach the Python code used to generate the graph in your PDF.

Step 6: Analytical Explanation

- Explain why the addition of a column of ones in Step 4 (bias term) and subsequent recalculation improves the fit of the model to the data.

Submission Requirements:

- Submit your completed assignment as a PDF containing all necessary code snippets and graphs.
- Attach corresponding .py and .ipynb files containing executable code.
- Ensure all files are clearly labeled and organized.
- Late submissions will be subject to deductions according to the policy.

Evaluation Criteria: Your assignment will be graded based on the accuracy of your implementation, the clarity of your presentation, and the completeness of your submission. Specific attention will be given to how well you follow these guidelines:

- **Step-by-Step Explanation:** This assignment requires clear, step-by-step explanations. You must explicitly label and explain each step (e.g., Step 1, Step 2, Step 3, Step 4). Failure to provide detailed explanations for each required step will result in a proportional deduction in your score. For example, if the assignment requires four steps and you complete only three, this will result in a score of 75 out of 100.