CAP 4630 - Artificial Intelligence

Instructor: Dr. Chen Chen

Homework #1 Linear regression

Be sure to show all the work involved in deriving your answers! If you just give a final answer without explanation, you may not receive credit for that question. For programming problem, you are recommended to use Python programming language.

[30 points] **Problem 1:**

We will use a dataset provided: "D3.csv". The first three columns are explanatory variables

x_1, x_2, x_3, and the fourth column is the dependent variable *y*.

Run linear regression simultaneously using all three explanatory variables. (1) Report the linear model you found by running the gradient descent algorithm. (2) Predict the value of *y* for new (x_1, x_2, x_3) values (1, 1, 1), for (2, 0, 4), and for (3, 2, 1).

Hints

- Don't forget the bias (intercept) term. A common way is to add a column of ones to your feature matrix.
- Feature scaling is crucial for gradient descent. Standardize features (subtract the mean and divide by the standard deviation) to help convergence.
- Choose a reasonable learning rate (e.g., 0.01–0.05) and sufficient iterations (e.g., 10k–20k). If your cost increases, the learning rate is likely too high.
- Track the cost function J(θ) over iterations to verify convergence.
- After gradient descent finishes, convert the coefficients back to the original feature scale (if you standardized).
- For verification, you may also compute the closed-form solution (normal equation) and check that the results match.

To receive full credit (30 points), your work should clearly show the following:

1. **Derivation and explanation (10 points):**
    - Show the cost function J(θ).
    - Show the gradient update rule.
    - Explain any preprocessing (e.g., feature scaling).
2. **Code implementation (10 points):**
    - Provide well-documented Python code for gradient descent.
    - Show the loss curve to demonstrate convergence.
3. **Final answers (10 points):**

- Report the fitted linear function (coefficients for $\theta_0, \theta_1, \theta_2, \theta_3$).
- Report the predicted y values for the three requested inputs.

[25 points] **Problem 2:**

A website specializing in dongles (dongles-r-us.com) wants to predict the total dollar amount that visitors will spend on their site. It has installed some software that can track three variables:

• time (the amount of time on the page in seconds): x_1,

• jiggle (the amount of mouse movement in cm): x_2, and

• scroll (how far they scroll the page down in cm): x_3.

Also, for a set of past customers they have recorded the

• sales (how much they spend on dongles in cents): y. We see a portion of their data set here with n = 11 customers:

time: x_1	jiggle: x_2	scroll: x_3	sales: y
232	33	402	2201
10	22	160	0
6437	343	231	7650
512	101	17	5599
441	212	55	8900
453	53	99	1742
2	2	10	0
332	79	154	1215
182	20	89	699
123	223	12	2101
424	32	15	8789

Let the first three columns of the data set be separate explanatory variables x_1, x_2, x_3, and the fourth column be the dependent variable y. Compute the closed-form solution (or analytical solution) for the hypothesis function parameters: $\boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2, \theta_3]$. **Show each step**. Use the normal equation to obtain $\boldsymbol{\theta}$. You can use Python to compute the matrix inverse in deriving your solution.

Hint: Include an **intercept** by augmenting your design matrix with a column of ones: [**1**, X].

To receive full credit (25 points), include:

1. **Derivation and explanation (10 points)**
   - Write the model in matrix form y= X $\boldsymbol{\theta}$, state the MSE objective, and derive the normal equation.
2. **Code implementation (10 points)**
   - Provide Python that constructs X and y from the 11 rows and computes $\boldsymbol{\theta}$ using either $(X^TX)^{-1}X^Ty$, np.linalg.lstsq, or np.linalg.pinv.
   - Print the learned $[\theta_0, \theta_1, \theta_2, \theta_3]$.
3. **Final answers (5 points)**
   - Report your numerical $\boldsymbol{\theta}$ (four coefficients) clearly, with reasonable rounding (e.g., 4–6 decimals).