Exam Problem: Finding the Minimum Cost in OLS Regression

Context:

You are given a simple linear regression model without an intercept, where the predictor variable x and the dependent variable y are related by the model y=a1x. The dataset provided consists of the following x values: [1,2,3,4] and y values: [2,3,6,8].

Task:

For the coefficient *a*1, you are to calculate the cost function value (sum of squared errors) for each of the following values: [0.5,1,2,3]. Use the cost function defined as:

$$\text{Cost} = \sum_{i=1}^{n}(y_i - a_1 x_i)^2$$

where nn is the number of data points, $yiyi$ is the observed value, and $a1xia1xi$ is the predicted value based on the model.

Questions:

1. **Calculation**:
    - Calculate the sum of squared errors (Cost) for each value of a1 (0.5, 1, 2, 3).
2. **Identification**:
    - Identify which *a*1 value results in the lowest cost. What does this tell you about the fit of the model to the data?
3. **Interpretation**:
    - Interpret the implications if the *a*1 with the lowest cost is significantly different from the *a*1 values calculated through other methods like gradient descent or analytical solution.
4. **Discussion**:
    - Discuss the potential reasons why some *a*1 values result in higher costs. What might this indicate about the relationship between *x* and *y*?

Additional Instructions for Students:

- Show all your work. Calculations for each *a*1 value must be clear and organized.
- Explain your reasoning for each answer, particularly for the interpretation and discussion questions.

Solution:

Given:

- x **values**: $[1, 2, 3, 4]$
- y **values**: $[2, 3, 6, 8]$
- a_1 **values to test**: $[0.5, 1, 2, 3]$

Task:

Calculate the sum of squared errors (Cost) for each a_1 using the formula:
$\text{Cost} = \sum_{i=1}^{n} (y_i - a_1 x_i)^2$

Calculations:

1. **For $a_1 = 0.5$:**
   - $\text{Predicted } y = [0.5 \times 1, 0.5 \times 2, 0.5 \times 3, 0.5 \times 4] = [0.5, 1, 1.5, 2]$
   - $\text{Cost} = (2 - 0.5)^2 + (3 - 1)^2 + (6 - 1.5)^2 + (8 - 2)^2$
   - $\text{Cost} = 2.25 + 4 + 20.25 + 36 = 62.5$
2. **For $a_1 = 1$:**
   - $\text{Predicted } y = [1 \times 1, 1 \times 2, 1 \times 3, 1 \times 4] = [1, 2, 3, 4]$
   - $\text{Cost} = (2 - 1)^2 + (3 - 2)^2 + (6 - 3)^2 + (8 - 4)^2$
   - $\text{Cost} = 1 + 1 + 9 + 16 = 27$
3. **For $a_1 = 2$:**
   - $\text{Predicted } y = [2 \times 1, 2 \times 2, 2 \times 3, 2 \times 4] = [2, 4, 6, 8]$
   - $\text{Cost} = (2 - 2)^2 + (3 - 4)^2 + (6 - 6)^2 + (8 - 8)^2$
   - $\text{Cost} = 0 + 1 + 0 + 0 = 1$
4. **For $a_1 = 3$:**
   - $\text{Predicted } y = [3 \times 1, 3 \times 2, 3 \times 3, 3 \times 4] = [3, 6, 9, 12]$
   - $\text{Cost} = (2 - 3)^2 + (3 - 6)^2 + (6 - 9)^2 + (8 - 12)^2$
   - $\text{Cost} = 1 + 9 + 9 + 16 = 35$

Identification and Interpretation:

- **Lowest Cost**: The lowest cost is achieved when $a1$=2, with a cost of 1. This suggests that the model with $a1$=2 fits the data best among the given choices.
- **Implications**: The fact that $a1$=2 provides the best fit implies that the model where each unit increase in *xx* corresponds to a doubling of *yy* most accurately represents the relationship in the observed data. If other methods, like gradient descent or an analytical solution, suggested a significantly different $a1$, it might indicate issues like outliers, improper data scaling, or the need for more complex modeling (e.g., non-linear models or interaction terms).

Discussion:

- Higher costs for other $a1$ values indicate poor fit. For instance, very high or low $a1$ values (such as 0.5 or 3) produce larger errors, showing that they do not capture the underlying pattern as effectively. This could suggest that the simple linear model may not fully capture the complexity of the data relationship, or that these particular values are far from the optimal coefficient for this dataset.

Exam Problem: Minimizing a Quadratic Function Using Gradient Descent

Context:

You are given a simple quadratic function, which is often used in optimization problems to test algorithms like gradient descent due to its predictable curvature and known global minimum. The function is defined as follows:

$$f(x) = x^2 + 4x + 4$$

Objective:

Your task is to apply the gradient descent algorithm to find the minimum of this function starting from an initial point x_0=0. Use the gradient descent update rule, where xx is updated iteratively using:

$$x_{\text{next}} = x - \alpha \cdot \nabla f(x)$$

where α is the learning rate.

Given Parameters:

- Initial point, x_0=0
- Learning rate, α = 0.1

Task:

1. **Gradient Calculation**:
    - Calculate the gradient of the function $f(x)$, $\nabla f(x)$.
2. **First Iteration**:
    - Perform the first iteration of gradient descent to update xx from $x0x0$. Show your calculations.
3. **Subsequent Iterations**:

- Perform two more iterations of gradient descent, updating xx each time. Calculate and show each step.

4. **Discussion**:
   - Discuss how the choice of $\alpha\alpha$ affects the convergence of gradient descent in this scenario. What might happen if $\alpha\alpha$ was significantly larger or smaller?

5. **Conclusion**:
   - Based on your calculations, estimate how many iterations are needed to get x within a range of [−0.1,0.1] of the function's minimum.

Solution:

Objective:

Minimize $f(x)$ using gradient descent.

Task 1: Gradient Calculation

First, calculate the gradient (derivative) of the function $f(x)$:

$$\nabla f(x) = \frac{d}{dx}(x^2 + 4x + 4) = 2x + 4$$

Task 2: First Iteration

Apply the gradient descent update rule:

$$x_{\text{next}} = x - \alpha \cdot \nabla f(x)$$

For $x_0 = 0$:

$$x_1 = 0 - 0.1 \cdot (2 \cdot 0 + 4) = 0 - 0.1 \cdot 4 = 0 - 0.4 = -0.4$$

Task 3: Subsequent Iterations

Second Iteration:

$$x_2 = x_1 - \alpha \cdot \nabla f(x_1)$$
$$x_2 = -0.4 - 0.1 \cdot (2 \cdot (-0.4) + 4) = -0.4 - 0.1 \cdot (3.2) = -0.4 - 0.32 = -0.72$$

Third Iteration:

$x_3 = x_2 - \alpha \cdot \nabla f(x_2)$
$x_3 = -0.72 - 0.1 \cdot (2 \cdot (-0.72) + 4) = -0.72 - 0.1 \cdot (2.56) = -0.72 - 0.256 = -0.976$

Task 4: Discussion

The learning rate $\alpha = 0.1$ significantly influences the convergence speed. In our case, α is well-chosen and allows for steady convergence towards the minimum. If α were much larger, it might cause overshooting, where x bounces around or diverges from the minimum. If α were too small, convergence would be slow, requiring more iterations to reach the minimum.

Task 5: Conclusion

Estimating the number of iterations to reach a point within $[-0.1, 0.1]$ of the minimum (which is at $x = -2$):

By the third iteration, $x_3 = -0.976$, approaching the minimum. Observing the trend and the decrease in the distance between subsequent x values, it's likely that within 2-3 more iterations, x would converge close to -2. Thus, a total of about 5-6 iterations is a reasonable estimate to get x within the specified range of the minimum.