LAHORE UNIVERSITY OF MANAGEMENT SCIENCES
Department of Electrical Engineering

EE 514 (CS 535) Machine Learning
Quiz 5 Solutions

Name: _____

Campus ID: _____

Total Marks: 10
Time Duration: 15 minutes

We have gone through the details of Ridge Regression and have studied gradient descent algorithm. In this quiz we will reinforce the same understanding. To keep things really simple – Since you must have completed your first homework, let us redo the Problem 2 (exactly the same ... or is it ?) from your homework.

Question 1 (7 marks)

You need to compute the gradient $\nabla J(\theta)$ and derive the gradient descent update rule $\theta^{(t+1)}$ for the regularized objective:

$$J(\theta) = \frac{1}{2n}\|y - A\theta\|^2 + \frac{\lambda}{2}\|\theta\|^2$$

.

Solution: We aim to minimize the regularized cost function:

$$J(\theta) = \frac{1}{2n}\|y - A\theta\|^2 + \frac{\lambda}{2}\|\theta\|^2$$

where: - $y \in \mathbb{R}^n$ is the vector of output values, - $A \in \mathbb{R}^{n\times(M+1)}$ is the design matrix for polynomial features, - $\theta \in \mathbb{R}^{M+1}$ is the parameter vector, - λ is the regularization parameter.

We will compute the gradient $\nabla_\theta J(\theta)$ and derive the gradient descent update rule.

The cost function is given as:

$$J(\theta) = \frac{1}{2n}(y - A\theta)^T(y - A\theta) + \frac{\lambda}{2}\theta^T\theta$$

The gradient of $J(\theta)$ with respect to θ is:

$$\nabla_\theta J(\theta) = -\frac{1}{n}A^T(y - A\theta) + \lambda\theta$$

Expanding $y - A\theta$:

$$\nabla_\theta J(\theta) = -\frac{1}{n}A^T y + \frac{1}{n}A^T A\theta + \lambda\theta$$

The gradient descent update rule is:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha\nabla_\theta J(\theta)$$

Substitute $\nabla_\theta J(\theta)$ into the update rule:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha\left(-\frac{1}{n}A^T y + \frac{1}{n}A^T A\theta^{(t)} + \lambda\theta^{(t)}\right)$$

Simplify the expression:

$$\theta^{(t+1)} = \theta^{(t)} + \alpha\frac{1}{n}A^T y - \alpha\left(\frac{1}{n}A^T A\theta^{(t)} + \lambda\theta^{(t)}\right)$$

The final update rule is:

$$\theta^{(t+1)} = \theta^{(t)} + \frac{\alpha}{n}A^T y - \alpha\left(\frac{1}{n}A^T A\theta^{(t)} + \lambda\theta^{(t)}\right)$$

This rule is used to iteratively update θ until convergence.

Question 2 (1 mark)

Explain the impact of λ (very high / very low values) in terms of model under-fitting and over-fitting.

Solution:

- When λ is very high: The model is too simple because the regularization forces the coefficients to be very small. This can cause the model to miss important patterns in the data, leading to underfitting (poor performance on both training and test data).
- When λ is very low: The model becomes too flexible, fitting the training data very well but also capturing noise. This can lead to overfitting, where the model performs well on training data but poorly on new, unseen data

Question 3 (2 marks)

In gradient descent, the cost per update for SGD is $O(1)$ which is much lower then $O(m)$ of Mini-Batch GD. In general SGD takes large number of epoch to converge as compared Mini-Batch Gradient descent. (e.g 200 Vs 100). Based on this information, justify which one is more computationally **expensive** ?

Solution: SGD $O(200n)$ more expensive then mini-batch $O(100n/m)$