Feedback — IV. Linear Regression with Multiple Variables

Help

You submitted this quiz on **Wed 26 Mar 2014 10:35 AM IST**. You got a score of **5.00** out of **5.00**.

Question 1

Suppose $m = 4$ students have taken some class, and the class had a midterm exam and a final exam. You have collected a dataset of their scores on the two exams, which is as follows:

midterm exam	(midterm exam)^2	final exam
89	7921	96
72	5184	74
94	8836	87
69	4761	78

You'd like to use polynomial regression to predict a student's final exam score from their midterm exam score. Concretely, suppose you want to fit a model of the form $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$, where x_1 is the midterm score and x_2 is (midterm score)^2. Further, you plan to use both feature scaling (dividing by the "max-min", or range, of a feature) and mean normalization.

What is the normalized feature $x_2^{(4)}$? (Hint: midterm = 89, final = 96 is training example 1.) Please enter your answer in the text box below. If applicable, please provide at least two digits after the decimal place.

You entered:

-0.469

Your Answer		Score	Explanation
-0.469	✔	1.00	
Total		1.00 / 1.00	

Question Explanation

The mean of x_2 is 6675.5 and the range is $8836 - 4761 = 4075$ So $x_1^{(1)}$ is

$\frac{4761-6675.5}{4075} = -0.47$.

Question 2

You run gradient descent for 15 iterations with $\alpha = 0.3$ and compute $J(\theta)$ after each iteration. You find that the value of $J(\theta)$ **decreases** quickly then levels off. Based on this, which of the following conclusions seems most plausible?

Your Answer		Score	Explanation
Rather than use the current value of α, it'd be more promising to try a larger value of α (say $\alpha = 1.0$).			
$\alpha = 0.3$ is an effective choice of learning rate.	✔	1.00	We want gradient descent to quickly converge to the minimum, so the current setting of α seems to be good.
Rather than use the current value of α, it'd be more promising to try a smaller value of α (say $\alpha = 0.1$).			
Total		1.00 / 1.00	

Question 3

Suppose you have $m = 23$ training examples with $n = 5$ features (excluding the additional all-ones feature for the intercept term, which you should add). The normal equation is $\theta = (X^T X)^{-1} X^T y$. For the given values of m and n, what are the dimensions of θ, X, and y in this equation?

Your Answer		Score	Explanation
X is 23×5, y is 23×1, θ is 5×1			
X is 23×6, y is 23×6, θ is 6×6			
X is 23×6, y is 23×1, θ is 6×1	✔	1.00	
X is 23×5, y is 23×1, θ is 5×5			

Total	1.00 / 1.00

Question Explanation

X has m rows and $n+1$ columns (+1 because of the $x_0 = 1$ term). y is an m-vector. θ is an $(n+1)$-vector.

Question 4

Suppose you have a dataset with $m = 1000000$ examples and $n = 200000$ features for each example. You want to use multivariate linear regression to fit the parameters θ to our data. Should you prefer gradient descent or the normal equation?

Your Answer		Score	Explanation
Gradient descent, since $(X^T X)^{-1}$ will be very slow to compute in the normal equation.	✔	1.00	With $n = 200000$ features, you will have to invert a 200001×200001 matrix to compute the normal equation. Inverting such a large matrix is computationally expensive, so gradient descent is a good choice.
The normal equation, since gradient descent might be unable to find the optimal θ.			
The normal equation, since it provides an efficient way to directly find the solution.			
Gradient descent, since it will always converge to the optimal θ.			
Total		1.00 / 1.00	

Question 5

Which of the following are reasons for using feature scaling?

Your Answer		Score	Explanation
☐ It speeds up gradient descent by making each iteration of gradient descent less expensive to compute.	✔	0.25	The magnitude of the feature values are insignificant in terms of computational cost.
☐ It is necessary to prevent gradient descent from getting stuck in local optima.	✔	0.25	The cost function $J(\theta)$ for linear regression has no local optima.
☐ It speeds up solving for θ using the normal equation.	✔	0.25	The magnitude of the feature values are insignificant in terms of computational cost.
☑ It speeds up gradient descent by making it require fewer iterations to get to a good solution.	✔	0.25	Feature scaling speeds up gradient descent by avoiding many extra iterations that are required when one or more features take on much larger values than the rest.
Total		1.00 / 1.00	