# The Coders' Club

Machine Learning: G1

Week 2: Assignment

**Topics:**

- Linear Regression with Multiple Variables
- Polynomial Regression
- Octave/MATLAB
- MATLAB Onramp Course

**Instructions:**

- Upload your answers on the following Google Drive link strictly in PDF format.
  **https://drive.google.com/open?id=1kFVsZB4rmrwyoqf0Mta6KpBU1atGxCKo**
- File name should be your name
- Show necessary calculations and steps for numerical questions.
- Share the screenshot of your MATLAB Onramp certificate in your answer
- Deadline: 06/01/2020

# Linear Regression with Multiple Variables

Q.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 are 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)$^2$. 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 $x_2^{(2)}$ ?

Q.2. Which of the following are reasons for using feature scaling?

- It is necessary to prevent gradient descent from getting stuck in local optima
- It speeds up gradient descent by making it require fewer iterations to get to a good solution
- It prevents the matrix $X^T X$ (used in the normal equation) from being non-invertible (singular/degenerate)
- It speeds up solving for θ using normal equation

Q.3. Suppose you have a dataset with m=50 examples 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?

- Gradient Descent, since it will always converge to optimal θ
- The normal equation, since gradient descent might be unable to find the optimal θ
- Gradient descent, since $(X^T X)^{-1}$ will be very slow to compute in the normal equation
- The normal equation, since it provides an efficient way to directly find the solution

Q.4. You run gradient descent for 15 iterations with α=0.3 and compute J(θ) after each iteration. You find that the value of J(θ) decreases quickly then levels off. Based on this, which of the following conclusions seems most plausible?

- Rather than use current value of α, it would be more promising to use a smaller value of α (say α=0.1)
- α=0.3 is an effective choice of learning rate
- Rather than use current value of α, it would be more promising to use a large value of α (say α=1.0)

Q.5. Suppose you have m=28 training examples with n = 4 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?

# Polynomial Regression

Q.1. Which of the following are true about Polynomial Regression? Check all that apply.

- Polynomial regression is a form of linear regression.
- Hypothesis function need to be linear.
- Combination of two or more features creating a new feature.
- Polynomial provides the best approximation of the relationship between the dependent and independent variable.

Q.2. Give the general form of the Polynomial Regression

Q.3. Suppose you want to predict a house's price as a function of its size. Your model is $h_\theta(x) = \theta_0 + \theta_1(\text{size}) + \theta_2\sqrt{((\text{size}))}$.

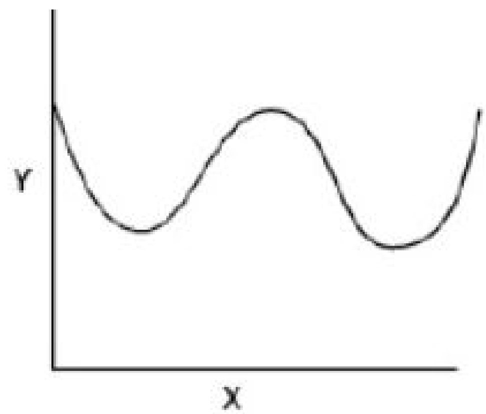
Suppose size ranges from 1 to 1000 ($\text{feet}^2$). You will implement this by fitting a model $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ .

Finally suppose you want to use feature scaling (without mean normalization). Which of the following choices for $x_1$ and $x_2$ should you use?

(Note: $\sqrt{1000} \approx 32$)

- $x_1 = \text{size}$, $x_2 = 32\sqrt{((\text{size}))}$
- $x_1 = 32(\text{size})$, $x_2 = \sqrt{((\text{size}))}$
- $x_1 = \text{size}/1000$, $x_2 = \sqrt{((\text{size}))} / 32$
- $x_1 = \text{size}/32$, $x_2 = \sqrt{((\text{size}))}$

Q.4. What will be the highest order of the equation in the given graph?



# MATLAB/Octave

Q.1 Suppose you have three vector valued variables u, v, w.

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}.$$

Your code implements the following:

```
for j=1:3
u(j) = 2 * v(j) + 5 * w(j)
end
```

How would you vectorize this code?

- u = 2 * v’ * v  * w + 5 * w’ * w * v
- u = 2 * v + 5 * w
- 5 * v + 2 * w
- u = 2 + v + 5 + w

Q.2. Suppose the following codes are executed in Octave/MATLAB.
A = [ 1 2; 3 4; 5 6];
B = [1 2 3; 4 5 6];
Which of the following are then valid commands? Check all that apply.

- C = A’ + B
- C = B * A
- C = A + B
- C = B’ * A

Q.3.

Let $A = \begin{bmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{bmatrix}$.

Which of the following indexing expressions gives

$B = \begin{bmatrix} 16 & 2 \\ 5 & 11 \\ 9 & 7 \\ 4 & 14 \end{bmatrix}$?

Check all that apply.

- B = A(: , 1:2)
- B = A(1:4, 1:2)
- B = A(0:2, 0:4)
- B = A(1:2, 1:4)

Q.4. Let A be a 10x10 matrix and x be a 10-element vector. Your friend wants to compute the product Ax and writes the following code:

```
v = zeros(10, 1);
for i = 1:10
  for j = 1:10
    v(i) = v(i) + A(i, j) * x(j);
  end
end
```

How would you vectorize this code to run without any for loops? Check all that apply.

- v = A * x
- v = Ax
- v = x’ * A
- v = sum(A * x)

Q.5. Say you have two column vectors v and w, each with 7 elements (i.e., they have dimensions 7x1). Consider the following code:

```
z = 0;
for i = 1:7
  z = z + v(i) * w(i)
end
```

- z = sum(v .* w)
- z = w’ * v
- z = v * w
- z = w * v

Q.6. In Octave/Matlab, many functions work on single numbers, vectors, and matrices. For example, the sin function when applied to a matrix will return a new matrix with the sin of each element. But you have to be careful, as certain functions have different behavior. Suppose you have a 7x7 matrix X. You want to compute the log of every element, the square of every element, add 1 to every element, and divide every element by 4. You will store the results in four matrices, A, B, C, D. One way to do so is the following code:

```
for i = 1:7
  for j = 1:7
    A(i, j) = log(X(i, j));
    B(i, j) = X(i, j) ^ 2;
    C(i, j) = X(i, j) + 1;
    D(i, j) = X(i, j) / 4;
  end
end
```

Which of the following correctly compute A, B, C or D? Check all that apply.

- C = X + 1
- D = X/4
- A = log(X)
- B = X^2

# MATLAB Onramp Course

- Create an account on www.mathworks.com

- Open the following link: https://matlabacademy.mathworks.com/R2019b/portal.html?course=gettingstarted&s_tid=course_mlor_start1

- Take MATLAB Onramp course

- Share the screenshot of your certificate in your submission PDF