MACHINE LEARNING

Answers:

1. The computational complexity of linear regression is:

B) (n)

2. Which of the following can be used to fit non-linear data?

A) Lasso Regression

3. Which of the following can be used to optimize the cost function of Linear Regression?

B) Gradient Descent

4. Which of the following method does not have closed form solution for its coefficients?

C) Lasso

5. Which gradient descent algorithm always gives optimal solution?

D) All of the above

6. Generalization error measures how well a model performs on training data.

A) True

7. The cost function of linear regression can be given as $(w0,w1)= 12m\Sigma(w0+ w1x(i)- y(i))2mi=1$.

The half term at start is due to:

B) it does not matter whether half is there or not.

8. Which of the following will have symmetric relation between dependent variable and independent variable?

C) Correlation

9. Which of the following is true about Normal Equation used to compute the coefficient of the Linear Regression?

A) We don’t have to choose the learning rate.
B) It becomes slow when numbers of features are very large.
C) We need to iterate.

10. Which of the following statement/s are true if we generated data with the help of polynomial features with 5 degrees of freedom which perfectly fits the data?

A) Linear Regression will have high bias and low variance.
B) Polynomial with degree 5 will have low bias and high variance.

11. Which of the following sentence is false regarding regression?

C) It discovers causal relationship.

12. Which Linear Regression training algorithm can we use if we have a training set with millions of features?

We could use batch gradient descent, stochastic gradient descent, or mini-batch gradient descent. SGD and MBGD would work the best because neither of them need to load the entire dataset into memory in order to take 1 step of gradient descent. Batch would be ok with the caveat that we have enough memory to load all the data.

The normal equations method would not be a good choice because it is computationally inefficient. The main cause of the computational complexity comes from inverse operation on an (n x n) matrix.

O n2. 4 to O n3

13. Which algorithms will not suffer or might suffer, if the features in training set have very different scales?

The normal equations method does not require normalizing the features, so it remains unaffected by features in the training set having very different scales.

Feature scaling is required for the various gradient descent algorithms. Feature scaling will help gradient descent converge quicker.