ROBOT LEARNING
EXCERCISE- MACHINE LEARNING

SUMMER SEMESTER 2021

Problem 1

In this excercise you will implement the linear regression algorithm in a computer language of your choice. A simple dataset is provided with two variables, X denoting the Population of a city and Y denoting the Profit of a Coffee Shop in that city. Dataset can be downloaded here:
https://github.com/sa32953/lero
This exercise is developed on the work in Stanford Machine Learning Course CS229 of Andrew NG.

A. Segregate the data into Input and Output Variable and plot the dataset to visualize.

B. Implement the Gradient Descent algorithm with Square Loss function and report the results.

(Consider error allowance in weight vector as 10^{-4} for convergency; Learning Rate = 0.01).

C. We introduce some outlier points: (17.5, 0), (18.5, 0), (19.5, 0). Implement Gradient Descent with Absolution Loss function characterizing Learning Rate = 0.02. Compare the result with that of Sqaure Loss (with given Outliers). Hint: Use subgradient for Absolute Loss func with parameters in part B.

D. Vary the learning rate in the interval [0.005, 0.02] and observe the number of iterations algorithm takes to converge. Plot the graph of Rate vs Iterations in Sqaured Loss func.

E. Comment if any random Learning Rate would allow Gradient Descent algorithm to reach a minimum ?

(Refer: https://medium.com/@ramsane/learning-rate-in-gradient-descent-what-could-possibly-go-wrong-6e58614941a0)

Problem 2

A. Prove that with Squared Loss function, analytical solution can be denoted as :

$$w^\star = \left(\sum_{i=1}^{m} x_i x_i^T\right)^{-1} \left(\sum_{i=1}^{m} x_i y_i\right)$$

B. Prove that the same can be represented compactly in ‚Normal Equations' format:

$$w^\star = (X^T X)^{-1} X^T y$$

C. Is is true for all types of loss functions ? Why not ?

Problem 3

Statement: Sqaured Loss can lead to a improper result in case of Outliers ? Is there a better Loss function in that case? If yes, compare the above two Loss functions.

Problem 4

A. Suppose that each output *y* is equal to hypotheses function $h_w(x)$ plus some Gaussian Noise *e.*

$$y = h_w(x) + e$$

With the probability density function $p(e) = \frac{1}{2\pi\sigma^2} exp(\frac{-e^2}{2\sigma^2})$. Prove that this approach is similar to that of Least-Sqaured Regression (minimizing Squared Loss func).

B. In Problem 3 we saw that results depend on type of loss function we choose. Is this problem really solved with using Probabilistic Representation?

Problem 5

In this problem you will implement the non-inear regression algorithm with a dataset with non-linear property. The given dataset is used from Kaggle, where the Input varibale X is the age (in years) of a human being and Output Variable Y is the height (in cms) of the given input human being. Dataset can be downloaded here: https://github.com/sa32953/lero/tree/master/ml/ex5. User can deploy the similar code to plot the data-points and realize that a straight line would not generalize the set anymore.

A. Write a code to find a regression polynomial of degree 2 that best represents the dataset. (Consider error allowance in weight vector as 10^{-5} for convergency; Learning Rate = 10^{-1}).

B. The degree of the polynomial governs the Losses generated by hypothesis function for given training data points. Find the regression polynomial for higher degree polynomials (degree = 3, 4, 6, 7, 8, ...). Plot all results on the same graph.

C. As degree increases further, regression tends to overfit training data and produces high losses in unknown data. Compute the prediction Error produced in ‚*Unknown data*' for a range of degrees [2,30]. (Hint: For ‚*Unknown data*' user can divide the given dataset into Training and Test groups using python library *sklearn*). (prediction Error = f(True Output - Estimated Output).

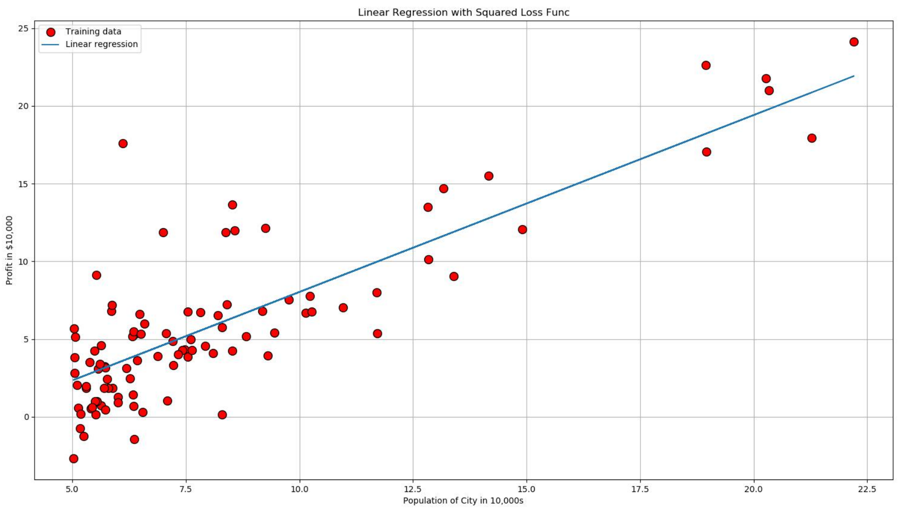
D. From the set of degrees in Part C, find an optimal degree that best predicts the „*Unknown data*".

SOLUTIONS

1b. The optimized weight vector is [1.13774908 -3.34547133].

Solving criteria with Sq Loss Func: Convergency = 0.0001 and Learining Rate = 0.01

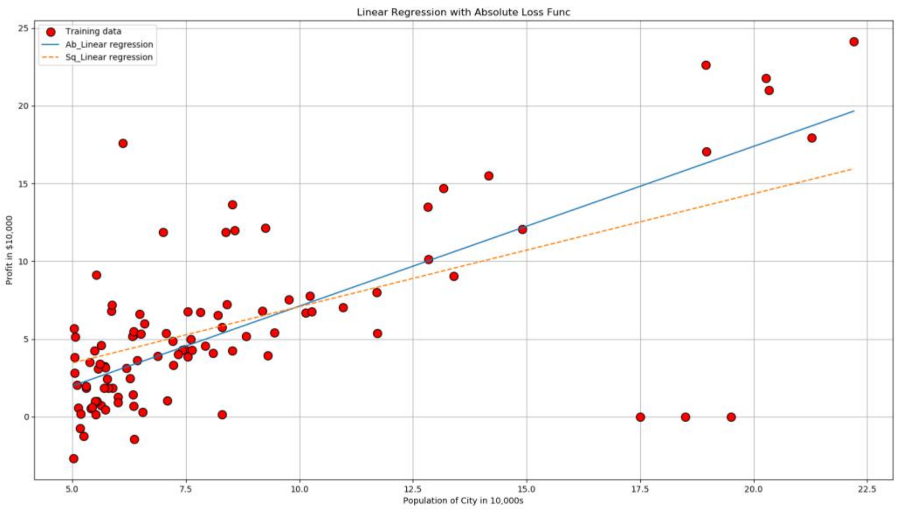
Total iterations done = 1096



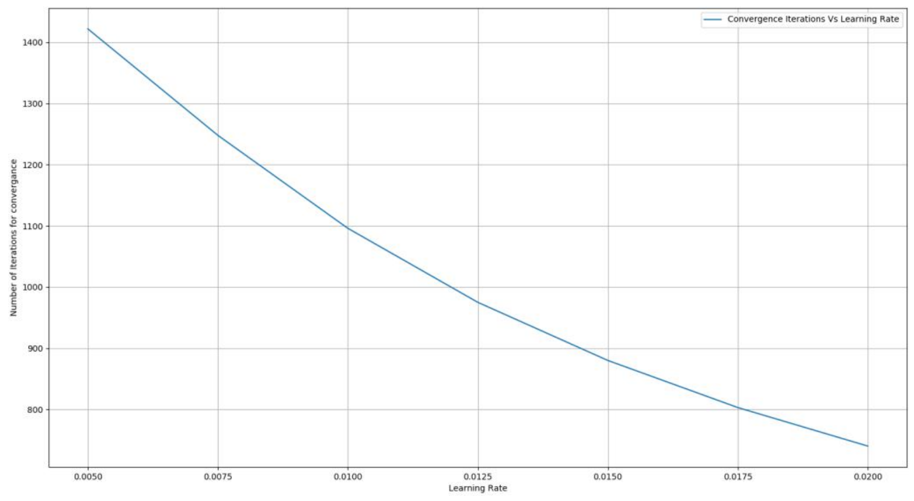
1c. The optimized weight vector is [1.02712024 -3.1534].

Solving criteria with Abs Loss Func: Convergency = 0.0001 and Learining Rate = 0.02

Total iterations done = 1862



1d.

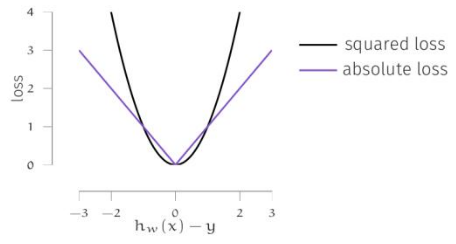


2c. The closed form solution cannot be written for every type of Loss function. Eg: For Absolute Loss function, it is not possible to write gradient and then equate it to zero to get optimal weight vector. In that case gradient descent is a good option.

3a. Absolute Loss function

squared loss: $L(h_w(x), y) = (h_w(x) - y)^2$

absolute loss: $L(h_w(x), y) = |h_w(x) - y|$

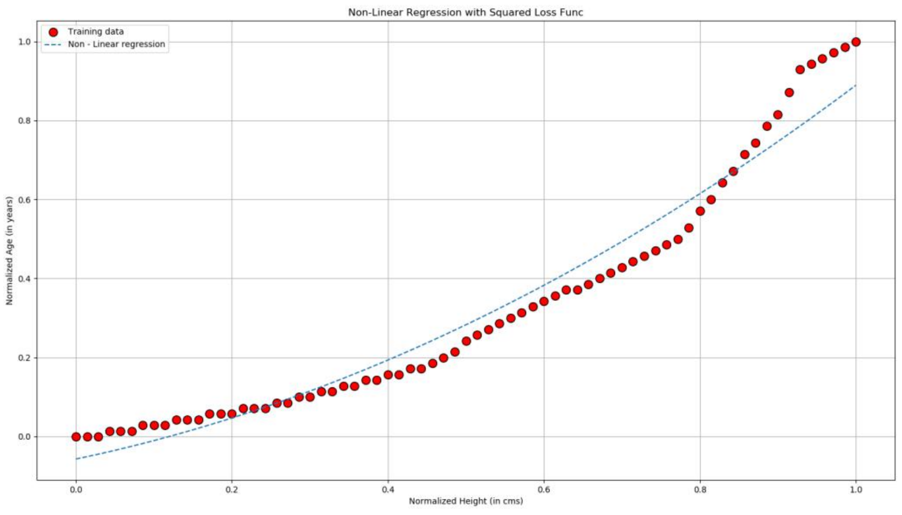


4b. Analogous to multiple types of los functions, we have multiple types of Probability Distributions which can be employed to maximize the likelihood estimation. So, we have just pushed the ques from ‚which type of loss' to ‚which type of distribution'.

5a. The optimized weight vector is [-0.05699084   0.41293153   0.53306576].

Solving criteria with Sq Loss Func: Convergency = 1e-05 and Learining Rate = 0.1

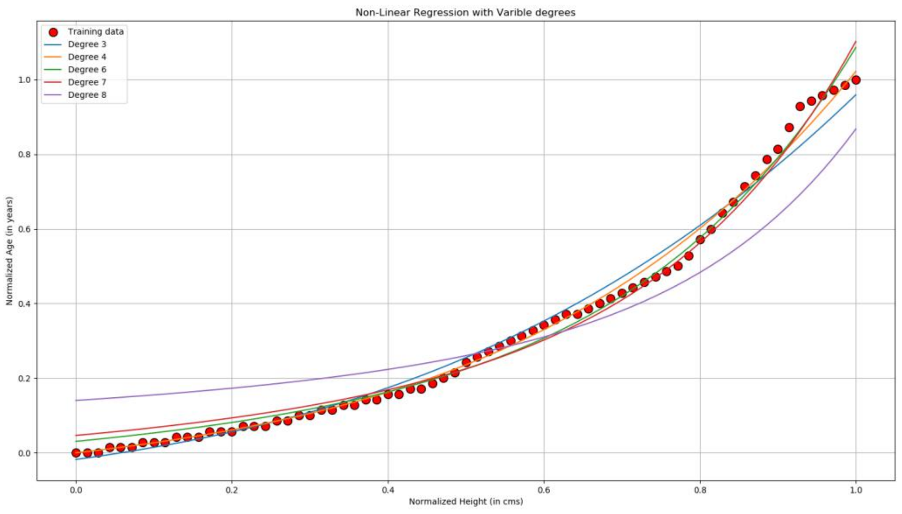
Total iterations done = 243



5b. The weight vectors are optimized for each degree specification.

Solving criteria with Sq Loss Func: Convergency = 1e-05 and Learining Rate = 0.1

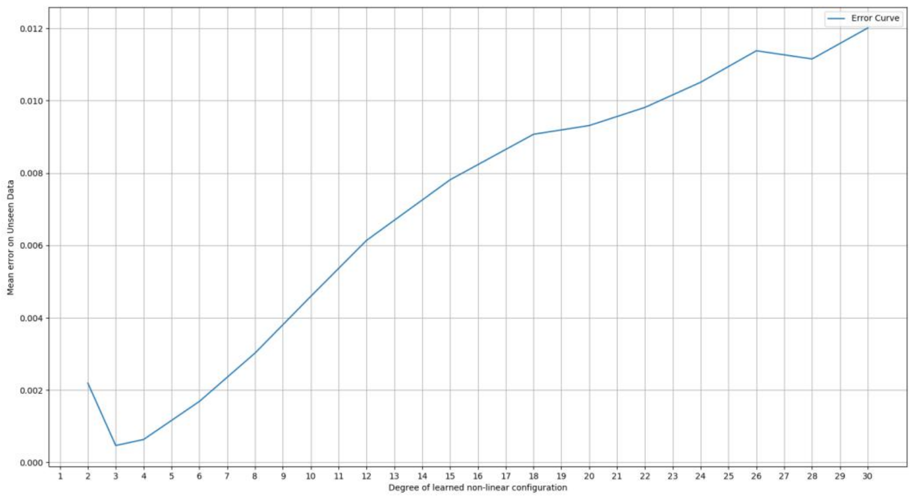
Total iterations done = [172, 171, 113, 85, 13]. (In same order as degree specification)



5c. The weight vectors are optimized for each degree specification.

Solving criteria with Sq Loss Func: Convergency = 1e-05 and Learining Rate = 0.1

Total iterations done = [239, 164, 167, 104, 60, 38, 26, 18, 16, 12, 11, 11, 12, 9, 10]. (In same order as degree specification)



5d. From the results of part c we see that at degree 4, the losses generated in Test data are minimum. We can safely say that for the given training data, we can deploy a non-linear regression of degree 4 and it would give us best possible results.