Ell 409

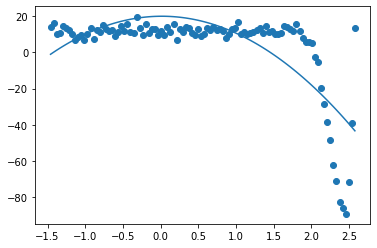
Assignment 1

Part 1 : PSEUDOINVERSE (PINV):

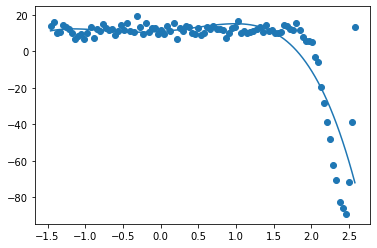
For pseudo inverse in part 1 the formulae for calculating PINV is Y = XB + E Where Y is matrix of predicted value by the model and the B is the matrix that contains the weight of our polynomial and X is the matrix of order N\*M where N is the number of training values and M is the order of polynomial that implies that B simply contains the polynomials for our polynomial and E is the Error matrix that contains the error between our actual and the predicted values of model. Now we have used the following formulas for polynomial fitting in our case:

Predicted\_Y = XB , B = (XtX)-1(Xt)Y where Xt is the transpose matrix of X, Error = Predicted\_Y -Y.

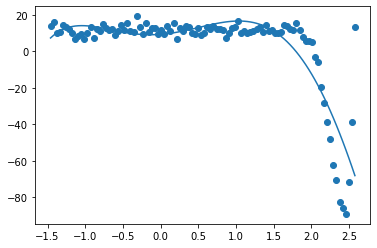
Now we will try to fit polynomials of various degrees in our given set of data containing gaussian error and will see if it really is the best possible polynomial or is it one among the overfitting and underfitting case.

So, we start with 2 degree polynomial and see that it is very much deviated from the testing data and is simply the case of underfitting.

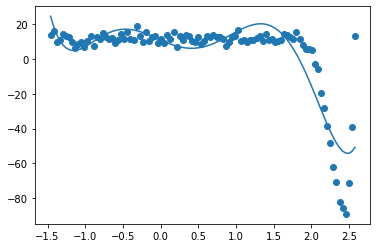
Hence we will increase the degree of our polynomial to see a better fitting of our curve and for that we will increase the degree to 4 and is shown below



We can clearly see that it is better than 3 but will again increase the degree to 5 to see if this is really the case of underfitting or overfitting and then will try to deduce something. Hence, I increase my degree by 1 more and is 5 now.The plot of 5 Is shown below



Again we can see that 5 is good and is as good as the polynomial with degree 4 so, lets increase by 1 more and make the total degree 6. The plot of data with degree 6 and the scattered given data is:



In this case as we can see that this is the case of overfitting because at last the plot is curling backwards so as to just go to the vicinity of a point which is probably the noise. Hence, we can deduce that the data given is basically a 4th or 5th order polynomial.

The programs used for plotting the data and polynomial regression is as below:

import numpy as np

import argparse

data = np.genfromtxt('gaussian.csv', delimiter = ',', skip\_header =0)

x = data[:,0]

x = x.astype(np.float32)

y = data[:,1]

y = y.astype(np.float32)

m = 5

A = np.zeros((len(x), m+1))

#b = np.zeros((m+1, 1))

for i in range(len(x)):

for j in range(m+1):

A[i ,j] = (x[i]\*\*(j))

b = np.dot(np.dot(np.linalg.inv(np.dot((A.T),A)),(A.T)),y)

print(b)

plt.scatter(x,y)

dict\_storage = {}

y\_predicted= np.dot(A,b)

for i in range(len(x)):

dict\_storage[x[i]] = y\_predicted[i]

x.sort()

y\_model =[]

for i in range(len(x)):

y\_model.append(dict\_storage[x[i]])

plt.plot(x, y\_model)