Assignment2 : Cs215

Saikiran-200050023,kamal-200050142

Question 3

Instructions to run code are given at the end

1 Set-1

We are given with the data set in the file ”points_Set1.mat”, which is the data from the 300 independent draws from the random variables X and Y.the data plot is given in the figure 1. The pca line we want plot must pass through the (x_{mean}, y_{mean}) and should have the least sum of perpendicular distances from all points in the data set. We can use the linear regression for calculating the slope of the line and constant in the line Eq $y = mx + c$, we first calculate the cross deviation and then deviation about x,

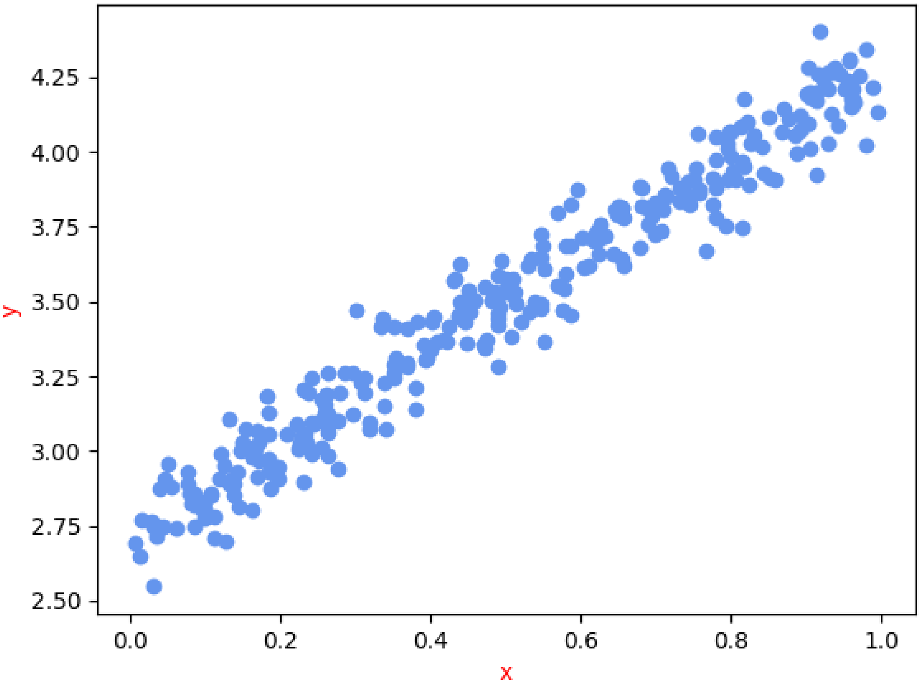

Figure 1: Data

$$cross\ deviation = np.sum(x * y) - n * x_{mean} * y_{mean}$$

$$cross\ deviation = (x_1 \cdot y_1 + x_2 \cdot y_2 + \ldots + x_n \cdot y_n) - n \cdot x_{mean} \cdot y_{mean}$$

$$deviation\ about\ x = np.sum(x * x) - n * x_{mean} * mean$$

$$deviation\ about\ x = n \cdot (x_1^2 + x_2^2 + \ldots + x_n^2) - n * x_{mean}{}^2$$

From the above two equations we can find the expected slope of the pca line as

$$slope = \frac{cross\ deviation}{deviation\ about\ x}$$

Finding the slope using the above eq is a better approximation than finding the mean of all $y - y_{mean}/x - x_{mean}$ because when the data is spread wide the spread of the $y - y_{mean}/x - x_{mean}$ will also be wide and thus not giving us the better approximation of pca line slope, as we already have the condition that the pca line must pass through the (x_{mean}, y_{mean}), so from this condition we can find the constant in the line Eq $y = mx + c$ as

$$c = y_{mean} - slope * x_{mean}$$

The slope and constant in line Eq $y = mx + c$ obtained from the equations are 1.572649 and 2.714159.

2 Set-1 plot

For Extracting the data from the ”points_Set1.mat” we have used the python package **h5py**(pip install h5py will download the h5py) ,The plot of the pca line for the data set ”points_Set1.mat” is given below in the figure 2, the line aprox Eq is $y = 1.57x + 2.71$, and the graph we can say that the plot is linear as the all the data points are in linear form

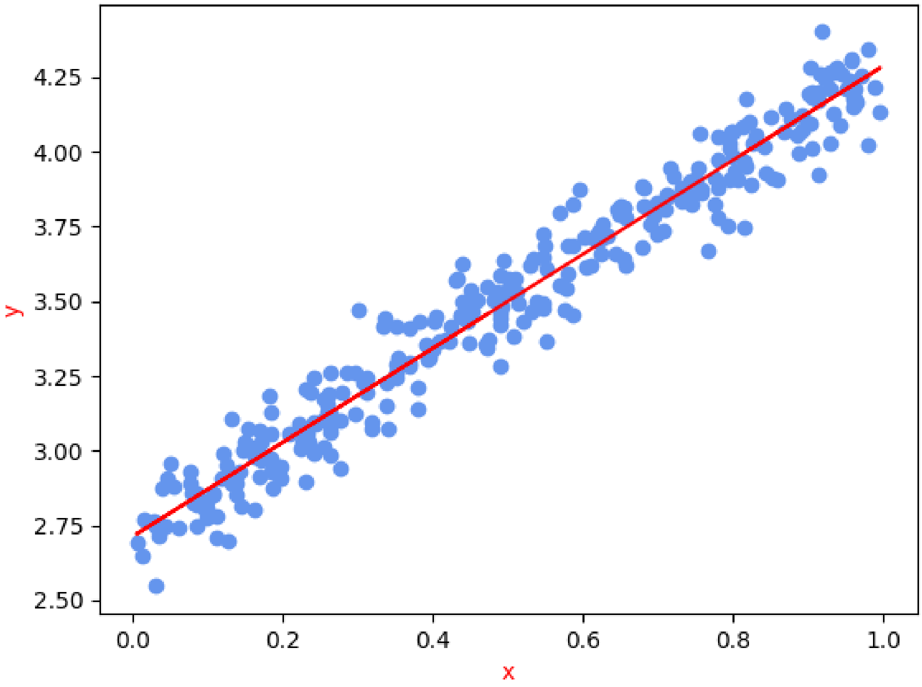

Figure 2: pca line plot

3 Set-2

All the arguments are same for Set-2 except that the plot is not linear this time and we have a total of 1000 data points, given is the plot of data extracted from the ”points_Set2.mat” using the same **h5py** python package as mentioned above and the 3 is the plot of the non linear data

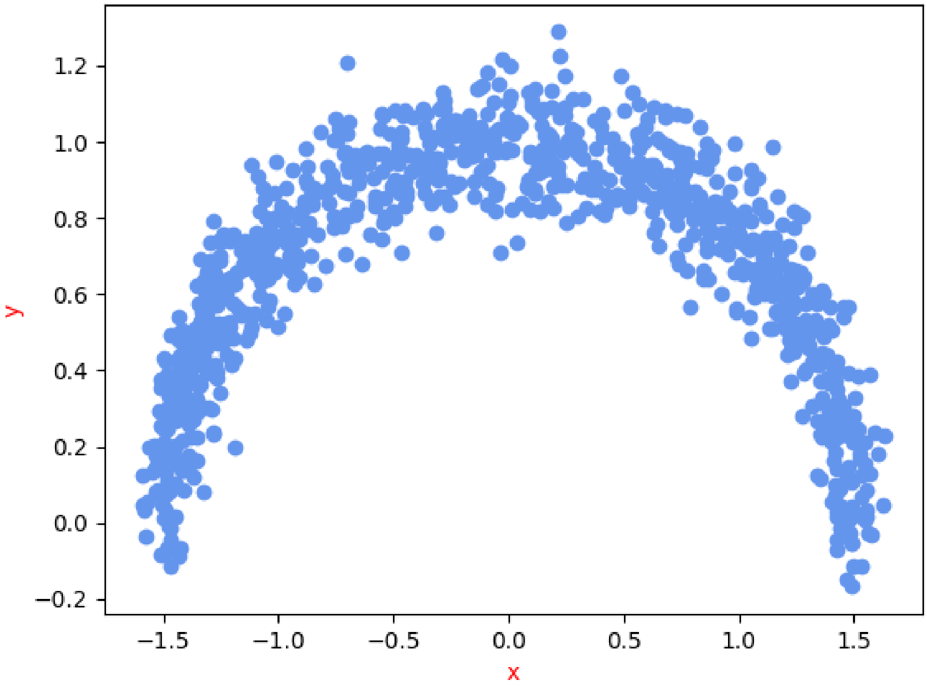

Figure 3: Non linear data

All the calculations we did for the Set-1 can be repeated here for finding the expected slope and constant in the line eq $y = mx + c$ and corresponding values of slope and constant are 0.014776 and 0.653231. The plot after inserting the line $y = 0.0147x + 0.6532$ is given below in the figure 4.

$$cross\ deviation = np.sum(x * y) - n * x_{mean} * y_{mean}$$

$$cross\ deviation = (x_1 \cdot y_1 + x_2 \cdot y_2 + \ldots + x_n \cdot y_n) - n \cdot x_{mean} \cdot y_{mean}$$

$$deviation\ about\ x = np.sum(x * x) - n * x_{mean} * mean$$

$$deviation\ about\ x = n \cdot (x_1^2 + x_2^2 + \ldots + x_n^2) - n * {x_{mean}}^2$$

From the above two equations we can find the expected slope of the pca line as

$$slope = \frac{cross\ deviation}{deviation\ about\ x}$$

as we already have the conditon that the pca line must pass through the (x_{mean}, y_{mean}), so from this condition we can find the constant in the line Eq $y = mx + c$ as

$$c = y_{mean} - slope * x_{mean}$$

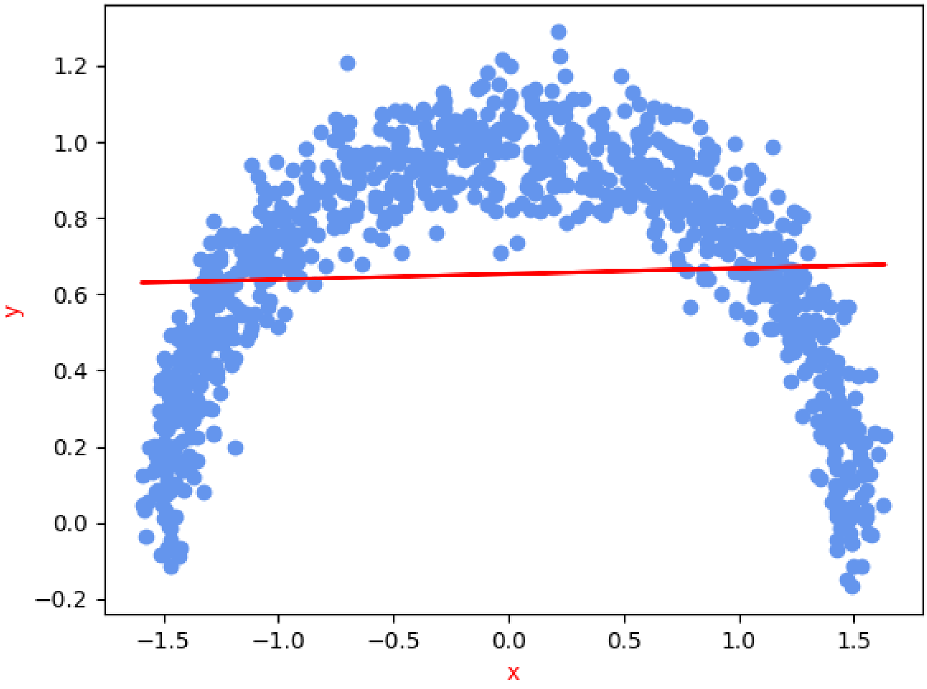

Figure 4: pca line plot

Instructions for running the code
Please move to the Q3 directory

- python3 ./code/q3_set1.py will plot the graph of the data points and pca line and save it to the results directory as q3_set1.png
- python3 ./code/q3_set2.py will plot the graph of the data points and pca line and save it to the results directory as q3_set2.png