**3. (15 points) PCA and Hyperplane Fitting.**

**3.a)(5 points) How can principal component analysis (PCA) be used to best approximate a linear relationship between random variables X and Y . Describe the method clearly, using appropriate mathematical descriptions for clarity. Your description should be clear enough to lead to a programmable implementation**

**Solution:**

⇒ Acc to given, we have 2 random variables X and Y and now we have sampling points (x,y)

⇒ Now that we have many points (x,y) we can find an approximate linear relation between X and Y and this is best accomplished by PCA.

⇒ In many Situations , we have large amounts of data and it is tough to handle such a huge amount of data , So we can do reduction of data such that the loss of information is minimized.

⇒ This can be done by following the below steps

⇒ step 1: First find the mean of the distribution (x0,y0) by just taking avg Separately on X and Y

⇒ Now just shift the points such that the mean is the origin.

⇒ Now that we have all points whose mean is (0,0). find a line that passes through the origin and exists in such a way that the sum of distances of projection of points from the origin is maximised. Or the distance of points from the line is minimised.

⇒ This is PCA1.

⇒ and line perpendicular to the PCA1 forms PCA2

⇒ Now the slope of PCA defines the linear relation between Y and X.

⇒ Now How can we easily execute the above process, PCA works here...

⇒ First standardise of the date (missing out will result in biased outcome)

⇒ Computing the covariance matrix. It is essential to identify heavily dependent variables because they contain biased and redundant info which reduces overall performance.

**Covariance\_Variance and mean is computed in below script…**

**def find\_ML\_estimates(final\_X):**

**data\_mean = np.matrix([[0.0], [0.0]])**

**data\_cov = np.matrix([[0.0, 0.0], [0.0, 0.0]])**

**n = len(final\_X)**

**for vec in final\_X:**

**data\_mean += vec**

**data\_cov += vec\*vec.transpose()**

**data\_mean /= n**

**data\_cov /= n**

**data\_cov -= data\_mean \* data\_mean.transpose()**

**return (data\_mean, data\_cov)**

⇒ Principal Components are basically a new set of variables that are obtained from the initial set . They compress and possess most of the useful information that was scattered among the initial values.

⇒ Now calculate eigenvalues and eigenvectors of the covariance matrix. These play a key role in determining PCA Components.

**Computing eigenvalues and eigenvectors**

**[e\_values, e\_vectors] = np.linalg.eig(covar)**

**idx = np.argsort(e\_values)[::-1]**

**e\_values = e\_values[idx]**

**e\_vectors = e\_vectors[:,idx]**

**Computing PCA1 and PCA2**

**PCA1 = e\_values[0]\*e\_vectors[:,0]**

**PCA2 = e\_values[1]\*e\_vectors[:,1]**

⇒ For a note PCA1 is the most significant and stores the maximum possible info.similarly PCA2 is second most and so on..

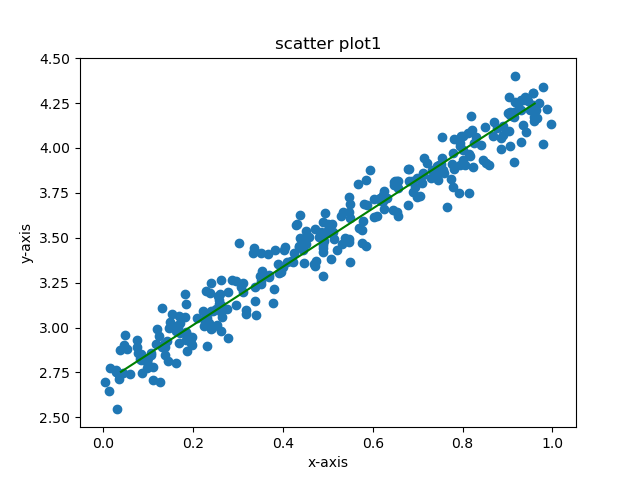
⇒ Now order the eigenvectors in descending order of eigenvalues such that the first one is PCA1 and so on

⇒ **So, linear relation between Y and X is the slope of PCA1**

**3b)(5 points) Show a scatter plot of the points. Overlay on the scatter plot, the graph of a line showing the linear relationship between Y and X.**

**Plot:**

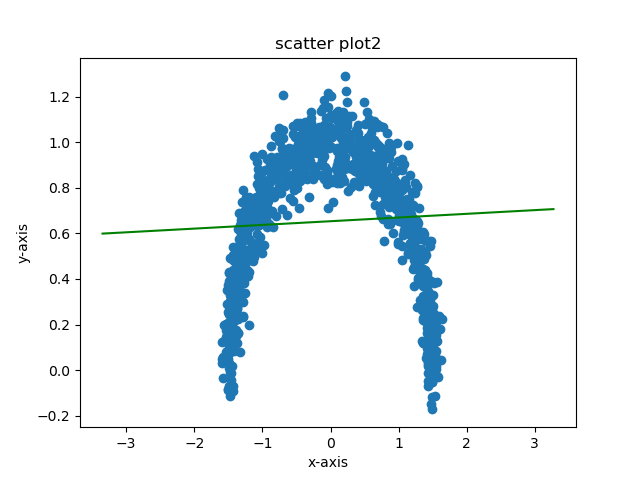
The green line is PCA1 and this is the scatter plot for **points2D\_Set1.mat**

****

**3c)(5 points) Repeat the same analysis for the set of points in "points2D\_Set2.mat". Show a scatter plot of the points. Overlay on the scatter plot, the graph of a line showing the linear relationship between Y and X. Compared to the result on the other set of points, justify the quality of the approximation resulting in this question using logical arguments.**

**Solution:**

The green line is PCA1 and this is the scatter plot for **points2D\_Set2.mat**

****

**⇒ By Observing the above 2 plots we can say that in both cases Y is positively correlated with X Because the PCA1 has a positive slope.**

**⇒ But we can say that the quality of approximation is more for the first plot rather than the second plot.**

**⇒ This is because The line covers a lot of points in case 1 and also the graph is more linear and we can say that the distance of points from the line are minimized very well.**

**⇒ But in the second case we have the scatter plot in the shape of an arc, clearly the linear relation between X and Y is not very helpful to analyse the data.**