Notes and assignment on MIS-6.

Prepared by Prof. K.P. Soman on 15/8/19

Orthogonality, Projection matrices, approximate solution and Regression

On coming Saturday (17^{th} August), First one-hour, I will devote to the concept of Orthogonality and Projection matrices. Through this topic on 'regression', you are taking a first baby step into the beautiful world of 'AI and Data Science'.

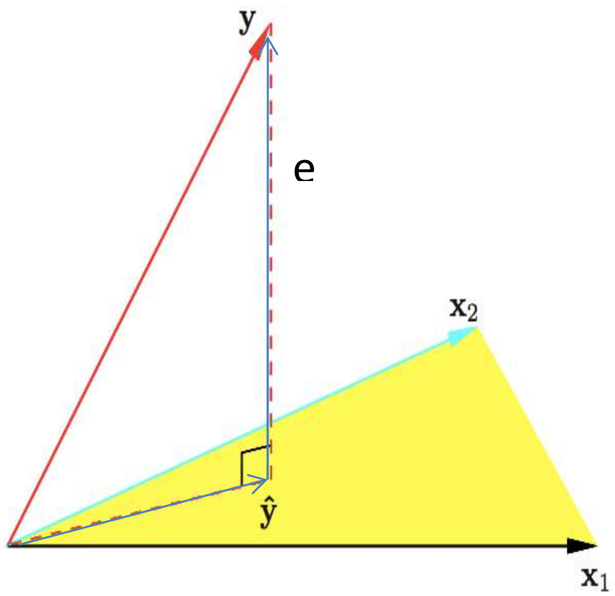
Mathematics Enthusiasts may learn the topic from this notes before the class and submit the assignment on 18^{th} August. Others need not be in panic. You may submit it on 23^{rd} August. With one more topic, Linear algebra for this semester is over. You will get enough time to slowly master the concepts covered. Soon we are moving to calculus.

Orthogonality and approximate solution to Ax=b

The concept of orthogonality can be utilized for finding approximate solution to Ax = **b** if solution does not exist. A solution does not exist for Ax=b if **b** is not in column s pace of A. (Above statement may be difficult to visualize initially)

We will start with a simple case with two columns in A. On the way to solve this problem, we will encounter a **special matrix** called projection matrix with several interesting properties.

Projection matrices



Consider a matrix A with x_1 and x_2 $\in \mathrm{R}^m$ as columns

x_1 and x_2 *span* column space of A

Let y $\in \mathrm{R}^m$ be a vector which is not in column space of $\mathrm{A}_{m\times 2}$

That is y is not in the plane spanned by x_1 and x_2

Let $\hat{y}$ be the projected vector of y in the column space of A

Now $\hat{y}$ is expressible as a linear combination of columns of A

$$\hat{y}=A\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix} = A\begin{matrix}|\\ \alpha\\ |\end{matrix} = \begin{bmatrix} | & | \\ x_1 & x_2 \\ | & | \end{bmatrix}\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix} = \alpha_1 \begin{matrix}|\\ x_1\\ |\end{matrix} + \alpha_2 \begin{matrix}|\\ x_2\\ |\end{matrix}$$

$$Now \;\; A\alpha + e = y$$

$$or \quad \begin{matrix}|\\ \hat{y}\\ |\end{matrix} + \begin{matrix}|\\ e\\ |\end{matrix} = \begin{matrix}|\\ y\\ |\end{matrix}$$

Note that e vector is orthogonal to column vectors of A

$$\therefore\ A^T e = \begin{bmatrix} x_1^T e \\ x_2^T e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \text{ vector}$$

$$Now\ \ A\alpha + e = y \Rightarrow A^T A\alpha + A^T e = A^T y \Rightarrow \alpha = \left(A^T A\right)^{-1} A^T y$$

$$\hat{y} = A\alpha = A\left(A^T A\right)^{-1} A^T y = Py \text{ where } P = A\left(A^T A\right)^{-1} A^T$$

$P = projection\ matrix$ for projecting a vector onto column space

This formula for P is applicable if matrix $A^T A$ is invertible.

$A^T A$ is invertible if all columns of A are independent. In the above example, the two column vectors are not collinear and hence independent.

In case, all columns are not independent, above formula do not work, since A^TA is not invertible. One way to solve this problem is to collect one set of **r** independent columns from A and form a matrix B, where **r** is the rank of the matrix A. Then the new projection matrix is $P=B(B^TA)^{-1} B^T$.

The most generic way of getting the projection matrix for projecting a vector into column space of A is given by the following formula.

$$P = A \times A^{\dagger} \quad \text{where } A^{\dagger} \text{ is pseudo inverse of } A$$

For any matrix (square and rectangular), there is a **pseudo inverse**.

You can appreciate this formula for projection matrix only after understanding the concept of **singular value decomposition of a matrix**.

So, the easiest and safest way of finding projected vector into column space is

$$\hat{y} = A \times A^{\dagger} \times y \quad \left(= AA^{\dagger} y\right)$$

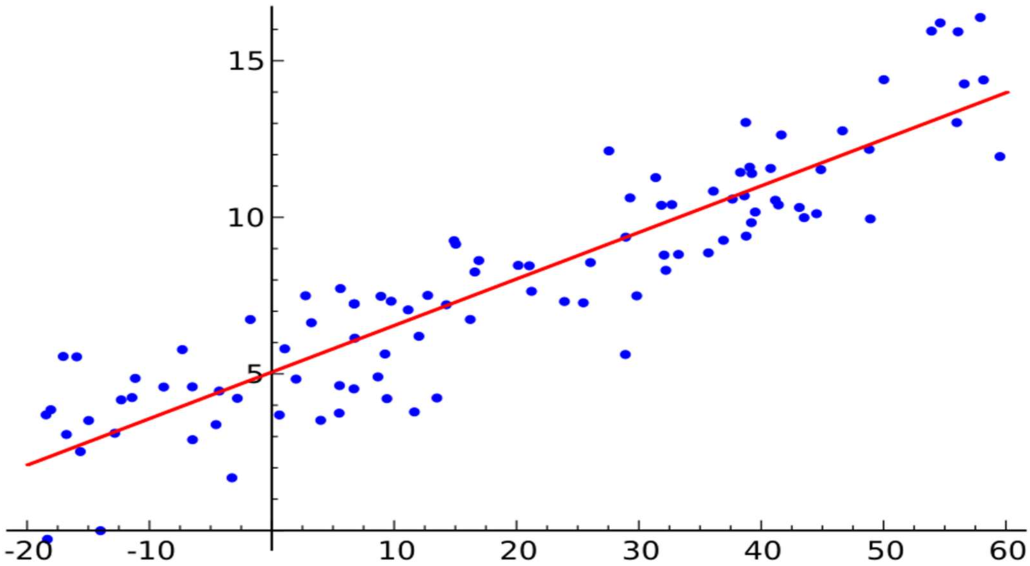
Note that vector y should have same tuple-size as column vectors of A.

How will you project a vector *z* onto row space of A?.

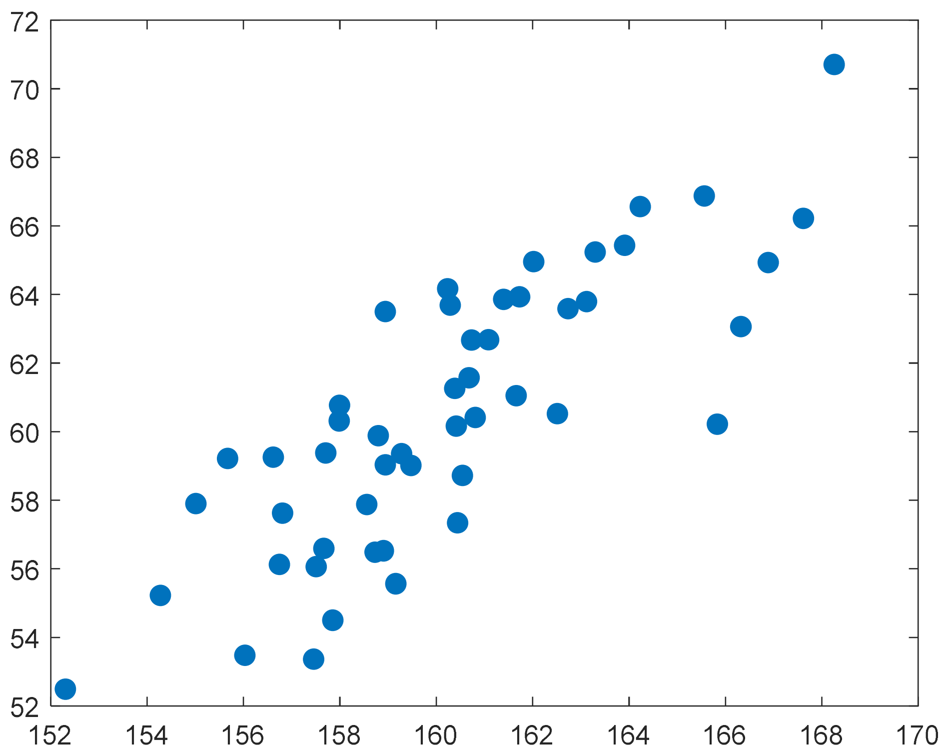
Projecting of given vector z on to rowsapce of A is same as projecting *z* onto column space of A^T **.**

$$\hat{z} = A^T \times \left(A^T\right)^{\dagger} \times z \quad \left(= A^T \left(A^T\right)^{\dagger} z\right)$$

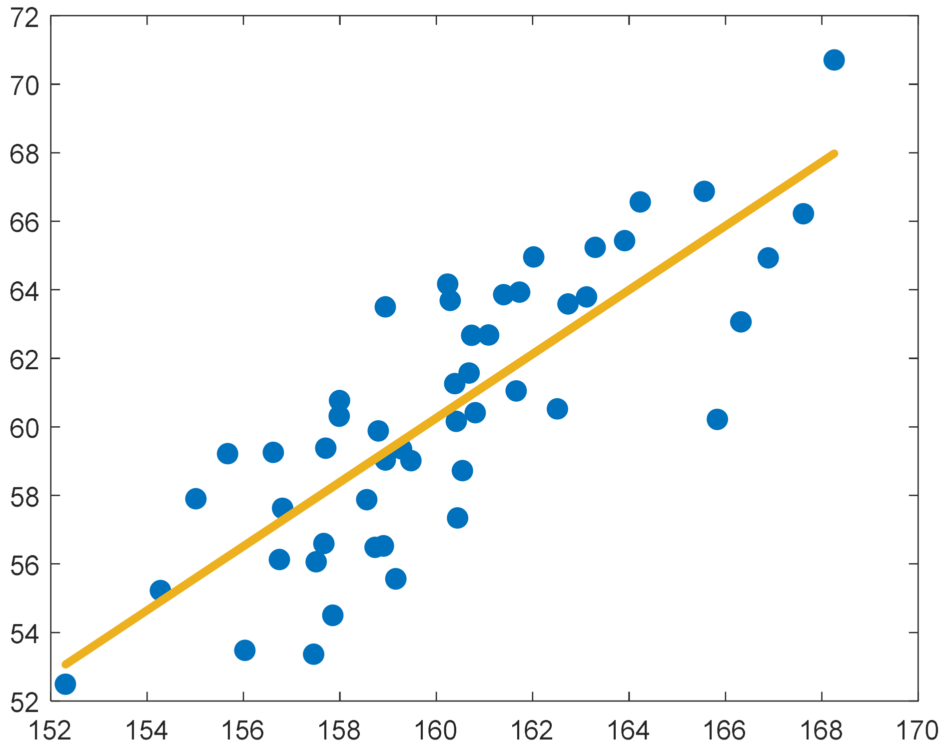
Application to fitting linear regression lines



In its simplest form, '*linear regression'* is about finding the linear relationship between a dependent and an independent variable. For example, we know there is an approximate linear relationship between height and weight of individuals in a population. Higher the height of a person, more the weight in general he/she has. Here, *weight* variable depends on *height* variable and hence, we say, *height* is independent variable and *weight* is dependent variable. If we collect data from a population and plot, what we obtain is a scattered collection of points (not all falling on a straight line) with a linear trend as in figure given below.



We fit what is called a **regression line**, i.e., a line passing through the data, the equation of which can approximately predict *weight* given the *height* of the person.



Such a line can be fitted using the concept of **projection matrix**.

Assume we have collected height values in column vector **x** and corresponding weight values in column vector **y**.

We are trying to find a relation of the form y = mx+c

We formulate the problem as follows

$$y_1 = mx_1 + c + e_1$$
$$y_2 = mx_2 + c + e_2$$
$$y_3 = mx_3 + c + e_3$$
$$y_4 = mx_4 + c + e_4$$
$$y_5 = mx_5 + c + e_5$$
$$\vdots$$
$$y_r = mx_r + c + e_r$$
$$\vdots$$
$$y_{n-1} = mx_{n-1} + c + e_{n-1}$$
$$y_n = mx_n + c + e_n$$

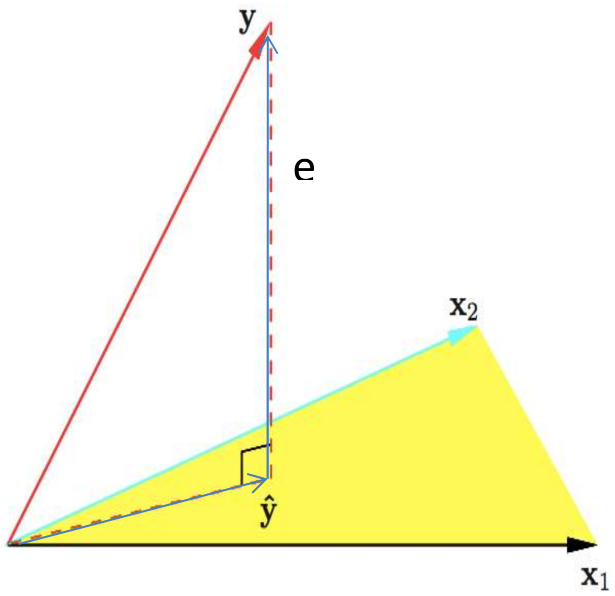
where, $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, …, $\boldsymbol{e}_n$ are deviation(it can be positive or negative depending on whether the point is above or below the fitted central line) from the y-values of the central line to be fitted.

In matrix format, above formulation is:

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} m \\ c \end{pmatrix}}_{\alpha} + \underbrace{\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}}_{e} \Rightarrow y = A\alpha + e$$

Now we are in the same situation as in the following figure:

In the figure, $\mathbf{X}_1$ and $\mathbf{X}_2$ are column vectors in A.



We obtain α (the slope and intercept) using the following formula

$$\begin{pmatrix} m \\ c \end{pmatrix} = \alpha = \left(A^T A\right)^{-1} A^T y; \quad \hat{y} = A\alpha$$

Plotting x Vs $\hat{y}$ provide the required central line, which we call regression line.

Computational Thinking.

How do we generate such data and estimate slope and intercept?

Warning: Some concepts used here are new to you. But you need to have only some intuitive idea about the concept. This may be easily obtained from Wikipedia. Make it a habit to visit wikipedia whenever you encounter a new term or concept.

Method 1

Steps:

1. We generate **n** equi-spaced points on a given(assumed) line like y=5x+10. We get an **x** vector and **y** vector.
2. Generate **n** random values from normal distribution with appropriate standard deviation and add to **y** to get $\mathbf{y_d}$. plot of x vs $\mathbf{y_d}$ will be a scatted set of data points.

MATLAB code snippet.

```
m=5; c =10 ;  % slope and intercepts
x= (-5:5)';  % x is a column vector
y=m*x+c; % y is a column vector
n=length(x); % number of data points
% let us generate n disturbing values from normal distribution with
% sigma=5 . That is,  standard deviation =5
noise= 5*randn(n,1);  % nx1 column vector
yd=y+noise;
A=[x ones(n,1)];
Alpha= inv(A'*A)*A'*yd;
ycap=A*Alpha;
plot(x,yd,'*') ;  % plot scattered data points
hold on
plot(x, ycap); % plot the regression line
xlabel('independent variable x')
ylabel('dependent variable y')
ev =yd-ycap;   % error vector.
% Verify that error vector is orthogonal to column vectors of A
Check = A'*ev ;
% print check on screen
Format bank
Check
```

Method 2.

We can generate data from bivariate Normal distribution. (get some intuitive idea from **Wikipedia** about 'normal probability distribution', 'correlation coefficient', 'variance' and 'variance-covariance matrix')

Let us consider height and weight distribution of human population.

Let the *height* be normally distributed random variable with mean 160 and variance=9.

Let the *weight* be normally distributed random variable with mean 70 and variance=16.

Let the correlation coefficient ρ between height and weight variable be 0.8. The correlation coefficient is a number which can take a value between -1 and 1. We can have positive or negative correlation.

Covariance between height and weight variable is given by the formula

$$COV(X,Y) = \rho\sigma_X\sigma_Y = \rho\sqrt{Var(X)}\sqrt{Var(Y)}$$

We can use above data to generate required 'scattered data points.

Here is the MATLAB code.

```
clear all
Mu=[160 70];   % mean values of height and weight variables X and Y.
VarX=9; VarY=16; % variance of X and Y variables
CorC=0.8; % Correlation Coefficient between X and Y
CoVarXY=CorC*sqrt(VarX)*sqrt(VarY); Covariance between X and Y
Sigma=[VarX  CoVarXY;  CoVarXY VarY]; % Covariance matrix
% Generate 500 datapoints
N=500;
Data=mvnrnd(Mu,Sigma,N); % variables X and Y are in columns
plot(Data(:,1),Data(:,2),'.')
xlabel('independent variable HEIGHT')
ylabel('dependent variable WEIGHT')
```

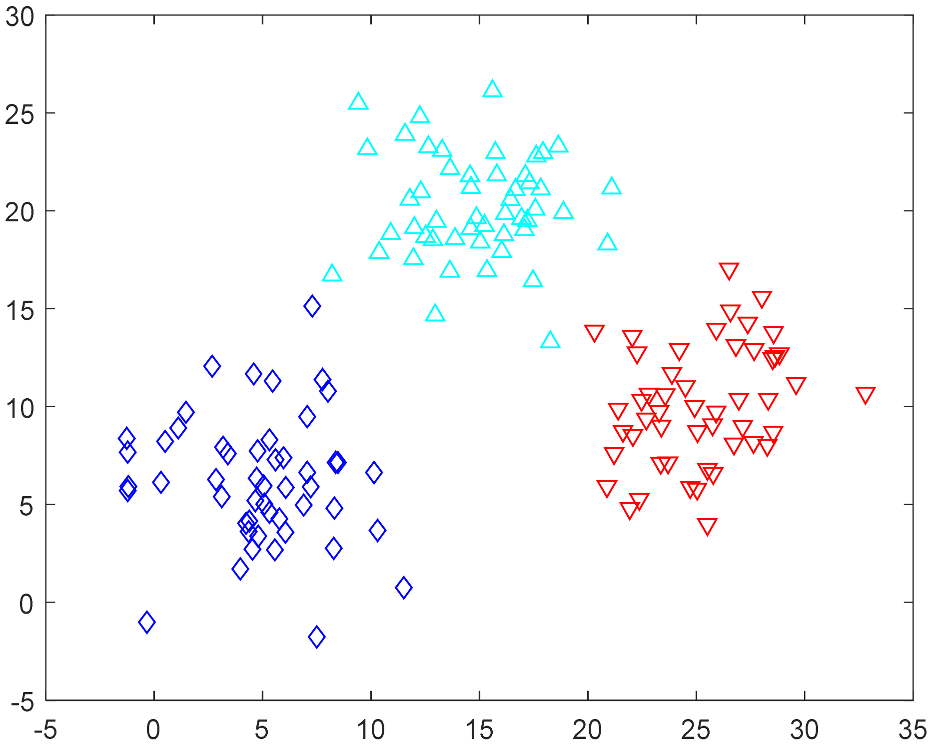
Clustering and Classification.

Read or see videos about what is **clustering** and classification in 'machine learning' from internet.

Creating data sets for clustering and classification

```
Mu1= [ 5 6]; Mu2= [ 25 10]; Mu3= [15, 20];
Sigma=2*eye (2);
N=50;
clust1=mvnrnd(Mu1, Sigma,N);
clust2=mvnrnd(Mu2,Sigma,N);
clust3=mvnrnd(Mu3,Sigma,N);
plot(clust1(:,1),clust1(:,2),'bd')
```

```
hold on
plot(clust2(:,1),clust2(:,2),'rv')
hold on
plot(clust3(:,1),clust3(:,2),'c^')
```



Assignment Questions. It is a very simple Assignment. May take only 1 hour.

1. Create projection matrix P of size 5x5 from matrix A with size 5x2.
2. Find Rank of above matrix P
3. Find P^2. What you observe? Can you prove the result analytically?
4. Find Eigenvalue of P. What you Observe? Verify the result by creating a new P from a new matrix A.

How do you relate rank and your observation, with regards to nonzero eigen values?

Fast learners may submit it on 18/9/19.
For others, submission date is on or before 23/9/19.