Task 5: PCA → Principle component Analysis → Dimensionality Reduction

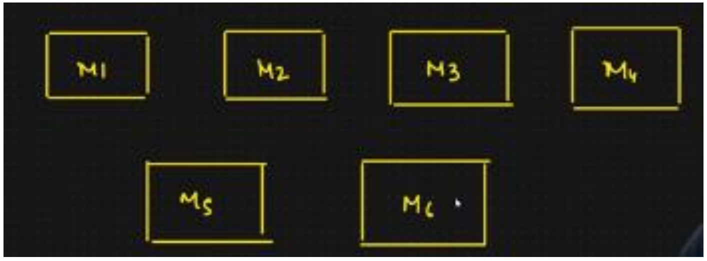
Author: Rida Aimen Mirza

Contents

Why should we use PCA ?

Curse Of Dimentionality

Lets say we have machine learning models M1 ,M2 ,M3 M4 , M5 , M6



Different models

Lets say that we have a data set in which there are around 500 features/dimensions

Suppose this data set is used to determine the size of the house.
So it may have many features including house size , no.of bedrooms ,no.of bathrooms etc
Suppose I provided 3 very important features to M1 , 6 very imp features to M2 , 15 features to M3 , 50 features to M4 , 100 to M5 and all 500 to M6 for training.
Lets suppose that I got accuracy 1 for Model M1 . Accuracy for model 2 would still go up because number of important features on which its is training has also increased.

Acc1 < Acc 2

Same goes for M3

Acc1 < Acc 2 <Acc3

But then we see that we have given M4 50 features and not all of them are going to be extremely important . Some might be very important , others might not be important at all .
So now , when we will try to find the accuracy you will see that it will tend to decrease

Acc1 < Acc 2 <Acc3 < Acc 4 >Acc 5>Acc 6

This is actually the curse of dimensionality

Over Fed Models

So Now when you look at M4 you can see that it was provided too many features , so many that it didn’t really require all those to give proper prediction so , we say that the model was OVER FEEDED . So we should give only as much features for learning to a model so that the accuracy goes on increasing.
Similarly in M5 , we see that it has been provided even more feature and it has been given too many features , so that when you give newer features to a model it learns about all the new ones even though they are not as important. So , too many input features cause over feeding and so the accuracy decreases

Degradation Of Model Performance

Model Performance Degrades because : As the number of features increase , the mathematical calculations now occur for all those features.

At some point just like a human being , our model may get confused .

Ways to Remove Curse of Dimensionality

1. Feature Selection (We will try to take most important features and then we will train our model)
2. Dimensionality Reduction (many ways , for example PCA) → this process is called features extraction . It says that we will try to derive a feature from a set of features where we will be capturing much essence of the previous features . Suppose we have features f1 ,f2, f3 , f4. In PCA we would derive two features D1 and D2 from these four features and then use these to
   Find out our output.

Difference between Feature Selection And Feature Extraction(Dimensionality Reduction)

Why should we use Dimensionality Reduction ?

1. To prevent curse of dimensionality
2. To improve performance of the model
3. To visualize the data → Visualization will lead to good understanding

Feature Selection

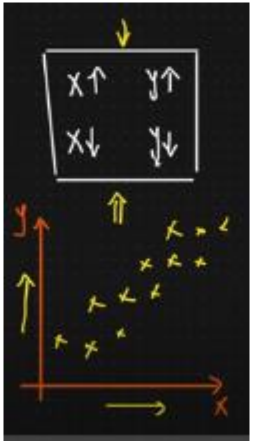
Suppose you have two features X and Y and by looking at the data you come to some type of conclusion for example that :

X is directly proportional to Y

Or maybe

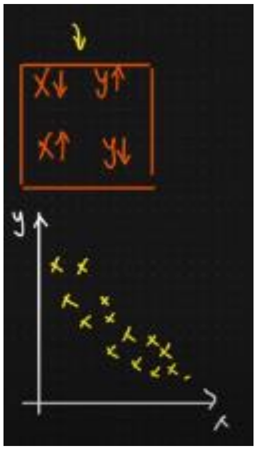
X an Y are indirectly Proportional

If we plot the above relation we would see that all points will be plotted linearly like this :



Graph of directly proportional relation

And if the second relation is plotted . our graph would look like this :

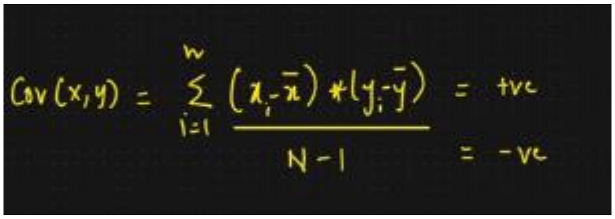


Graph of in-directly proportional relation

So you clearly know that in such linear relations . The X values are extremely important in telling what the Y value will be
and mathematically we can also find out a way to quantify this relationship . The technique used in this is

CO- VARIENCE

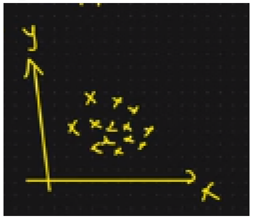
Its formula is :



$$\mathrm{Cov}(x,y) = \frac{\sum_{i=1}^{n} (x_i-\bar{x}) * (y_i-\bar{y})}{N-1} \quad \begin{matrix} = +ve \\ = -ve \end{matrix}$$

Covariance formula

When Covariance is positive it shows that there is linear directly proportional kind of relation
When Covariance is positive it shows that there is linear indirectly proportional kind of relationship
When Covariance is approximately = 0 . It means there is no relationship between X and Y
Such relation have graph that may have points in a circle



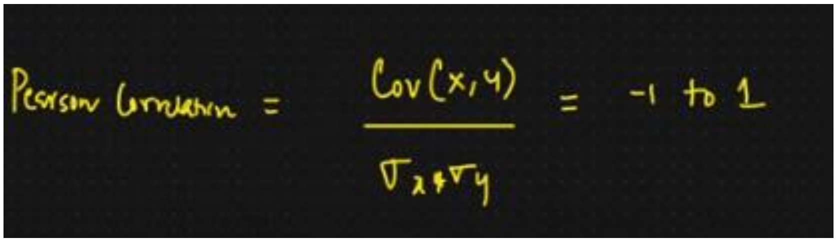
A scatter plot/ representing no relation between X and Y

And by looking at such a plot that shows that there is not much relation between X and Y

There is another relation called Pearson Correlation

Pearson Correlation

Its formula is :



$$\text{Pearson Correlation} = \frac{Cov(x,y)}{\sigma_x * \sigma_y} = -1 \text{ to } 1$$

And this gives output in the range of -1 to 1

The more , the value is towards +1 . The more positive correlated it is .

The more , the value is towards -1 . The more negative correlated it is .

The more , the value is towards 0 . No relationship .

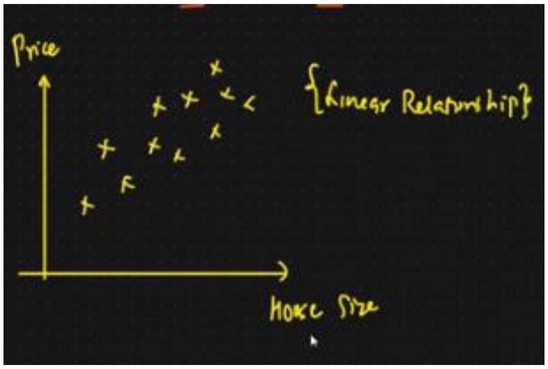
The above is just and idea about how the X and y are actually related .

There are other techniques in Feature selection also

Now lets suppose we have a data set (Housing data set) . In this data set , there are things like House Size , Fountain Size , Price . The first two are independent features and we use it to predict price .

If we use common sense we can see that fountain size is not really an important feature .

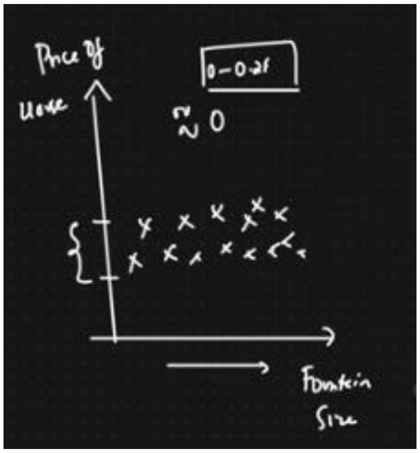
So now if a plot a graph between Price and House size . We see a graph as follows :



Now this shows that house size is an important feature to determine the price . because it has linear relationship . And how do you identify it ?

By using Covariance or Correlation

Now What about the other feature (Fountain Price) . Following is what its graph looks like



So we can see that with the increase in fountain size. The house price is stagnant . So it means that Foundatin size is not an important feature . So We can drop it

So now this is how we perform Feature selection process

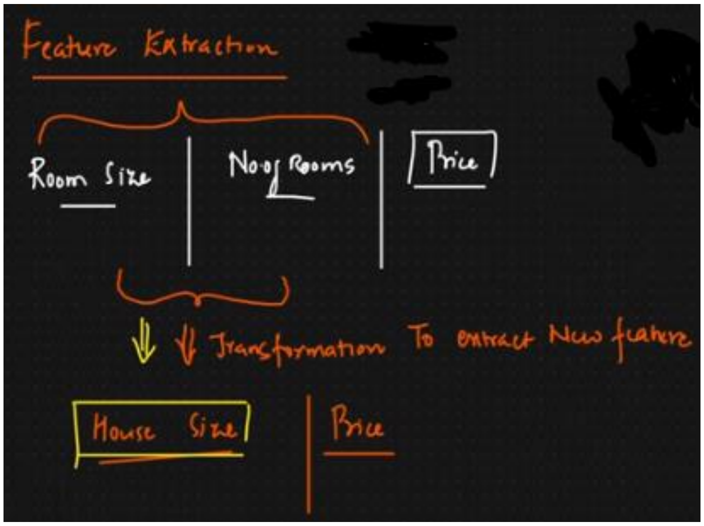
FEATURE EXTRACTION

Now consider we have the some housing data where we have features such as Room Size , No.of Rooms and Prices in this .

Now let’s suppose you want to reduce you features from two features to one feature .

Now why am I Not using feature selection ?? Because both of the above two features are extremely important

So how do we perform feature extraction ? . We the features on which we have to work , apply some transformation to extract a new feature . Lets suppose that the new feature that I extracted Is called house size . and now we are going to use this one to predict the price.

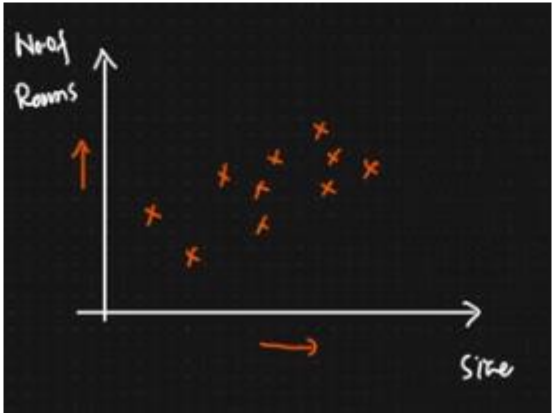


For a domain experty

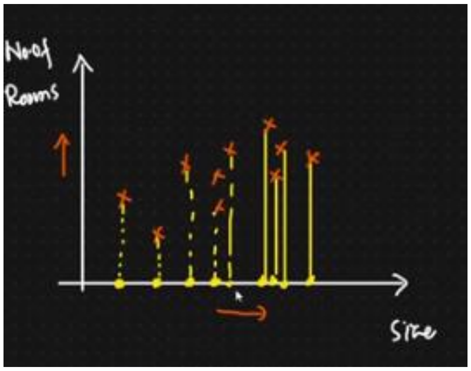
If he is give two features then he will predict the price but if only House Size is give , He will still Predict the price but with some difference because obviously some data has been lost .

PCA Geometric Intuition

We know that this is used for dimensionality reduction
Lets say I have features such as Size of the house and No of Rooms and we are using that to predict the price.
Now here if we plot a graph between Size of House and and price . I get :

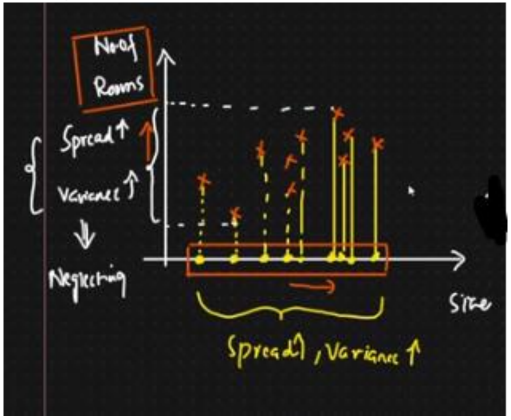


Now suppose that with the help of PCA I want to **covert 2dimension of features into 1dimension** .
For this , one way is feature selection (select one , ignore other)
A simple way of converting 2 dimension to one dimension is that I can project each point on X-Axis



So now clearly we can see that now we have our data points in one dimension
That is : I have converted 2D → 1D

One important point :
If I see the first and last data point . The area between this is the "Spread of data points"
If my Spread is huge → Variance will also increase (Directly proportional)
Disadvantage of this approach :
Here the size information is getting captured but the rooms information is getting lost . So this information that you had is getting lost .

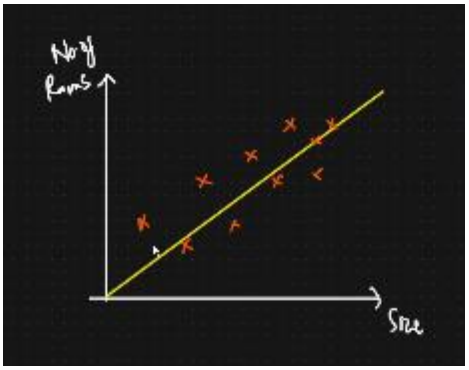


Now you see that . there is also a spread in the Y axis . and we are completely neglecting it . so There is loss of information occurring about No. of rooms
So once you convert from 2D to 1D you are loosing much information about one specific feature. So once you loose all this information. Your model may not word as efficiently.

Now here you are doing feature selection with lots of information loss . You don't want that .

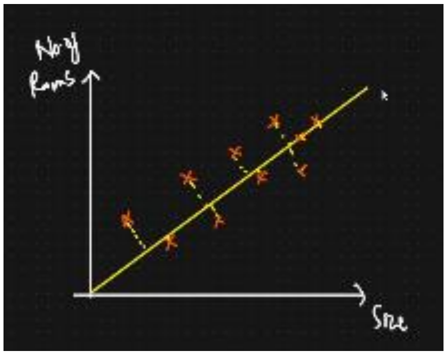
HOW TO PREVENT THE INFORMATION LOSS?

For this, you apply some transformation on the X and Y axis . For that you **apply Eigen Decomposition** on **some matrix** and this gives us a new axis that looks like this (the yellow line):

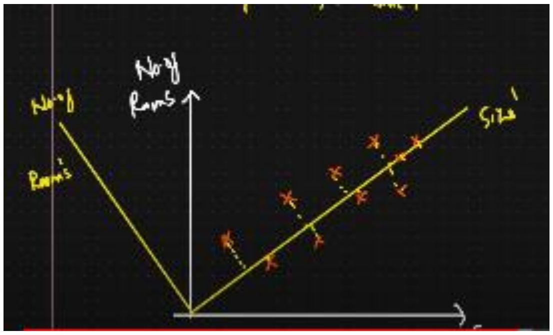


And when you get this axis . You will try to project all the information on this axis

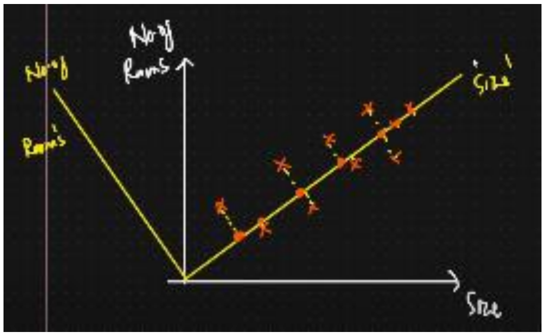
Like this :



Since you have 2 features . You will try to apply some transformation to change size axis to one more axis e.g size' and one more axis will get created which with be exactly perpendicular to this first one . No. of rooms'



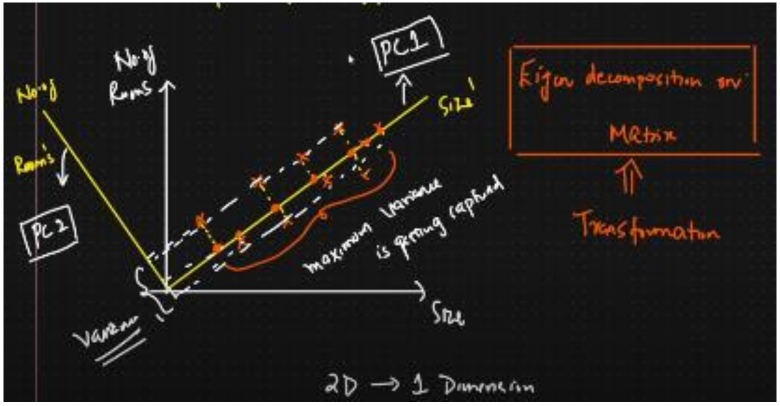
And then u will project all these data points . as follows



The difference between the two projections is that . in size' . we see that the spread has properly been captured in the new axis .

When you project the points onto the no.of days' axis you say that here you will be able to see that the spread will be very very less. (Maximum variance is getting captured).

And now you see that this is better because you are not loosing too much of information . now here you are converting 2D to 1D without too much information loss .



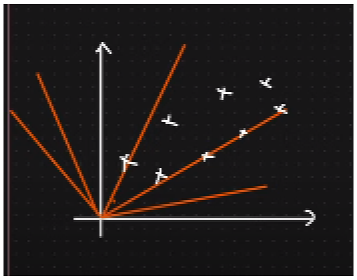
Main aim in PCA is to find out the new axis that helps us get minimum information loss. These yellow lines are what PCA's goal is . and we will call them PC1 and PC2 (Principle component 1 and principle component 2) . PC1 will be capturing maximum amount of variance . PC2 will be capturing 2nd maximum amount of variance .

At the end of the day we need to find one line that should be able to capture maximum amount of variance.

If you have 2Ds . You will get PC1 and PC2 . If you have 3Ds ,you will get PC1 , PC2,PC3 . And variance w.r.t PC1 > Variance w.r.t PC2 > Variance w.r.t PC3.

Lets say you have a task where you have a task where you have 2D points
Where you have to convert 2D to 1D .

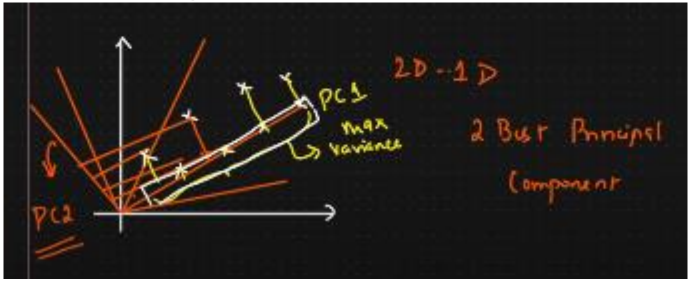
So what you will be doing ? You will apply PCA algo and it will be finding the best PC line and it can be either of the following orange lines.



And then it will try to select two best principle component lines and how do we decide on which one is the best ?

Only when we find the line with maximum variance. we will call it → PC1
When we find line with second maximum variance we will call it → PC2

And then at the end we will be taking all the points that we have plotted over the PC1



To Convert 3D to 1D
What will you do ?

Apply PCA

Get PC1 , PC2 , PC3

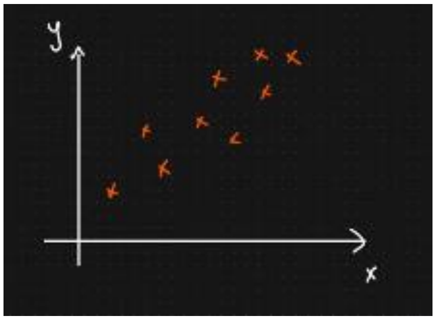
Take all points in PC1 and consider it as the 1 final dimension

To Convert 3D to 1D
Apply PCA

Get PC1 , PC2 , PC3

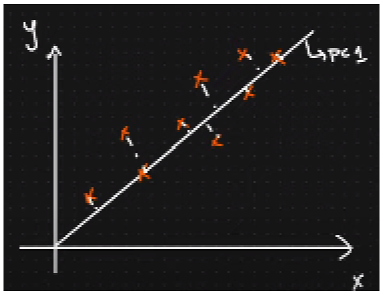
Take all points in PC1 and consider it as the 1st final dimension take all points in PC2 and consider it as 2nd final dimension

MATH INTUITON BEHIND PCA ALGORITHM



Lets consider I have 2 dimensions here and my aim is convert 2D to 1D in such a way that max variance is captured.

Lets suppose this white line is the PC1



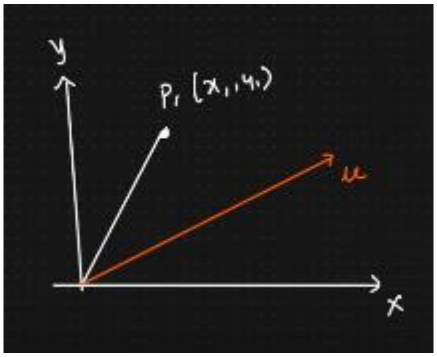
HOW DOES PCA DECIDE ON WHICH LINE IS THE BEST ?

That is based on two things:

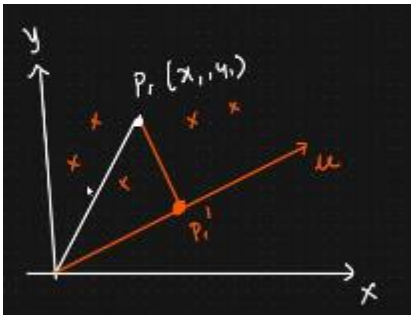
- Projection
- Cost function → related to variance

PROJECTION
Lets take one point P1 (x1,y1)

And consider it a vector and lets consider we have a unit vector represented by u



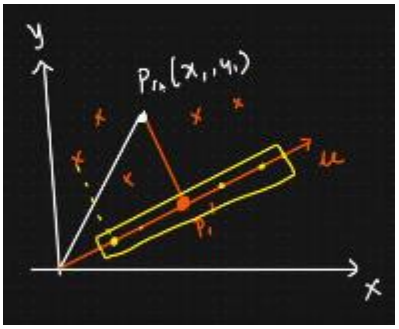
So lets say if u want to project P1 into u . Then the new projection that you get is P1’



Now what is the main aim ?

“To get maximum variance”

That is only possible if we do projection for all of the points. Then only we will be able to take all the points and calculate the variance.



The projection P1 on u is given by an equation

projection P1 on u = P1 vector * u vector/ magnitude of u

Proj$_{p1}$u = P1 * u / |u| (Output is a scalar value)

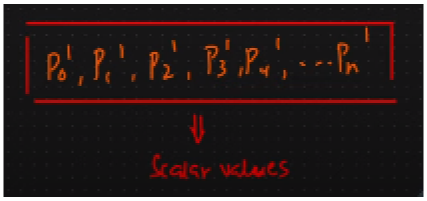
Since magnitude of u = 1

We can write

Proj$_{p1}$u = P1 . u

P1 = P1’

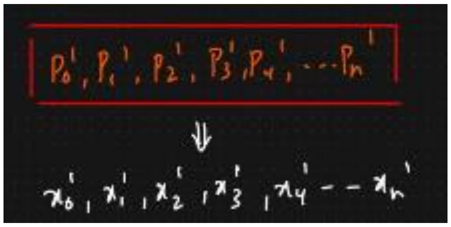
We are going to do projection for all the points



And what is the projection even talking about ? It’s the perpendicular distance from the point to the unit vector

After getting the scaler values for each projection we can now calculate the variance

Lets suppose the scaler values are represented as



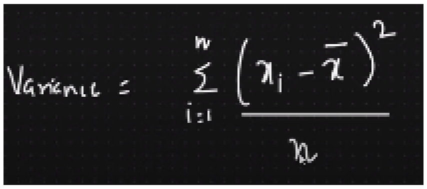
$$P_0', P_1', P_2', P_3', P_4', \cdots P_n'$$

$$\Downarrow$$

$$x_0', x_1', x_2', x_3', x_4' -- x_n'$$

To calculate variance

Variance = $\sum^{n}_{i=1}$ = $(x_{i\,-}\ x')^2 / n$



Varience = $\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n}$

Goal = find best unit vector (which captures maximum unit variance)

So this is basically our cost function

Eigen Vectors and Eigen Values

This concept is used to decide which unit vector is the best.

STEP 1 : Co-Variance Matrix between features

STEP 2: Eigen vectors and values will be cpputed from this co variance matrix

STEP 3: Largest Eigen vector (where value is highest) → this will capture the maximum variance

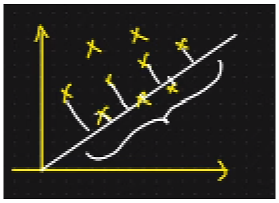
Eigen values and eigen vectors can be found out using simple equation :

$AV' = \lambda v'$ (A * vector v = λ vector v)

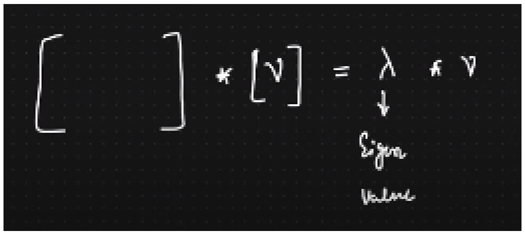
This equation is Linear Transformation of matrix

This equation will return us eigen values . One with highest value will be cohse as best line as it is mathematically proved that this gives max variance.

EIGEN VECTORS AND EIGEN VALUES (LINEAR TRANSFORMATION OR EIGEN DECOMPOSITION OF COVARIANCE MATRIX)

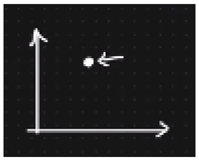


Lets consider we have some data points
And I have to find best line.
-
Suppose I have a specific matrix and a vector V .
If I apply linear transformation on the matrix . Ill get a lambda value multiplied by the same vector V



This lambda is your eigen value .

Lets say you have a point like this



On this if we apply linear transformation
Linear transformation means
If I have data that looks like this , in grid manner this can be moved in different different directions
Like from white to orange