Artificial Intelligence for Economics

Prof. Dripto Bakshi

Humanities and Social Sciences

Indian Institute of Technology Kharagpur

Week – 08

Lecture - 36

Lecture 36 : Dimensionality Reduction (Principal Component Analysis) – The Technique

welcome to ah the second lecture of on principle component analysis in the first lecture we had ah dealt with or I try to give you a glimpse of the mathematical prerequisites the mathematical tools which ah we would we would employ in ah this lecture so now that we are armed with them lets begin Let's start with, before we get into the technique, let's start with a little story of an early stage financial institution. Imagine a financial institution which is lending out small loans in the rural areas. And it has just started off. And what is it that a financial institution, a lender is usually bothered about? minimizing defaulters, be it large banks or be it smaller financial institutions who are lending out money, default is what people are bothered about and any such institution would want to look at customer data and try to understand what might lead to defaulting and thereby try to come up with red flags or red flag a customer who is likely to be a defaulter. So let's take a quick look at how such a data set might look like. So let's say something like this.

do not get bogged down by the code that is not important. So, let us say every customer is characterized by spending, advance payments, delay in the proportion of what they have to pay, current balance, credit limit so on and so forth. And at the final final column we have defaulter or not defaulter. So, if defaulter status is 1 it means that that particular customer has defaulted, if it is 0 have not defaulted.

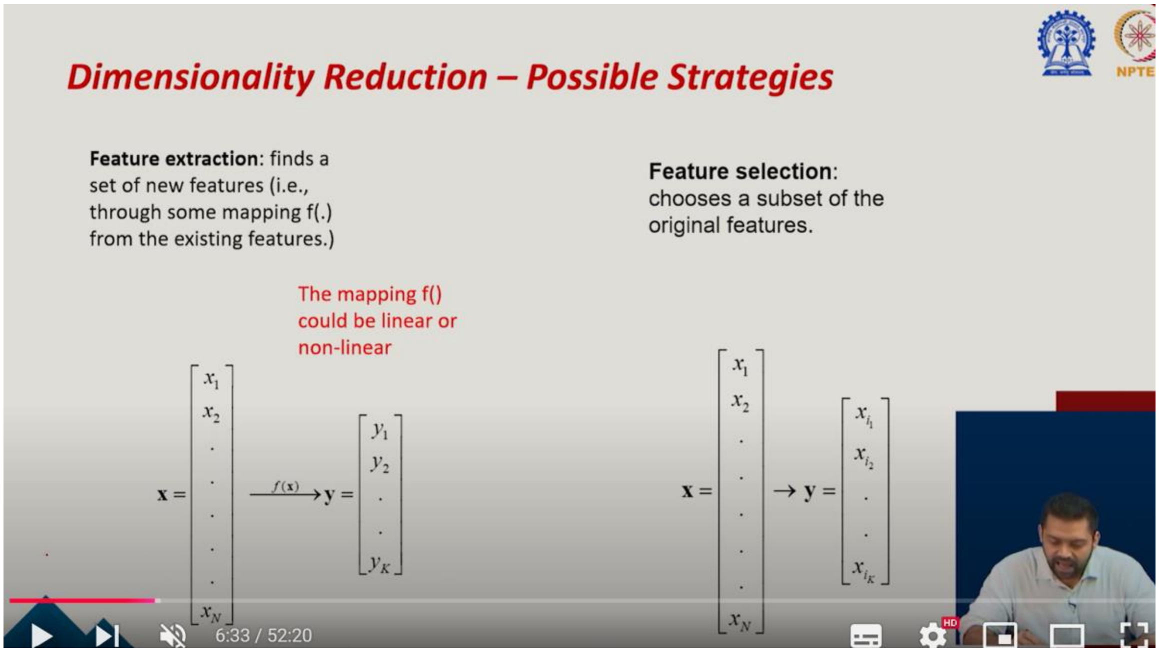
So, we see how many KYC features are there 1, 2, 3, 4, 5, 6, 7, 7 features for every customer. But how many customers are there? See only 210 rows. So, we have 7 features characterizing a customer and finally the output whether he is a default or not, but then we have only 210 customers that is not much. So the number of features are way more compared to the number of customers we have. So if we try to employ a classification algorithm on this, let's see how that works.

So what do, what do we typically do? So this is a Python notebook as you can see. So we'll, uh, divide the data frame up into two parts, test and train, sorry, train and test. This is DF train and this is DF test. So I have taken 163 observations in my training set, 47 in my test set. Then what do I do? I employ, I am not getting into the nitty-gritties of the code.

I simply apply a logistic regression model on this data, on the training data. And finally, I take my training the model which I have trained on to the on to the test data and I calculate accuracy of prediction and it turns out that it is only 68 percent that is not impressive that is not very bad, but that is not impressive either ok. And one reason for that it seems like is that the number of customers is too less ok. So, the bottleneck it seems in this analysis which the financial institution will probably encounter is there are too few customers and too many KYC and transaction features, too many features characterizing a customer, okay. So, very few observations, many explanatory variables and this leads to what is called the overfitting problem, okay.

And in case of unsupervised learning it leads to other problems, but let us not get there. So, in case of supervised learning this is the problem it leads to, it leads to low bias and high variance. So, what do we mean by that? It means that if we have many explanatory variables in the data and there are very few observations in the data, it leads to capturing the noise or idiosyncrasies of the data. ok. So, of the training data because we are training the model on the training data.

So, it captures the noise in the training data and so when we take this model to the test data it leads to very poor predictive performance. So, this is called the curse of dimensionality and dimensionality reduction usually is bail this out, bail this out of the scenario. Now, let us see. So, what is the objective? The objective is to reduce the dimensionality such that it improves performance. And usually there are two ways of



Dimensionality Reduction – Possible Strategies

Feature extraction: finds a set of new features (i.e., through some mapping f(.) from the existing features.)

The mapping f() could be linear or non-linear

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} \xrightarrow{f(\mathbf{x})} \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_K \end{bmatrix}$$

Feature selection: chooses a subset of the original features.

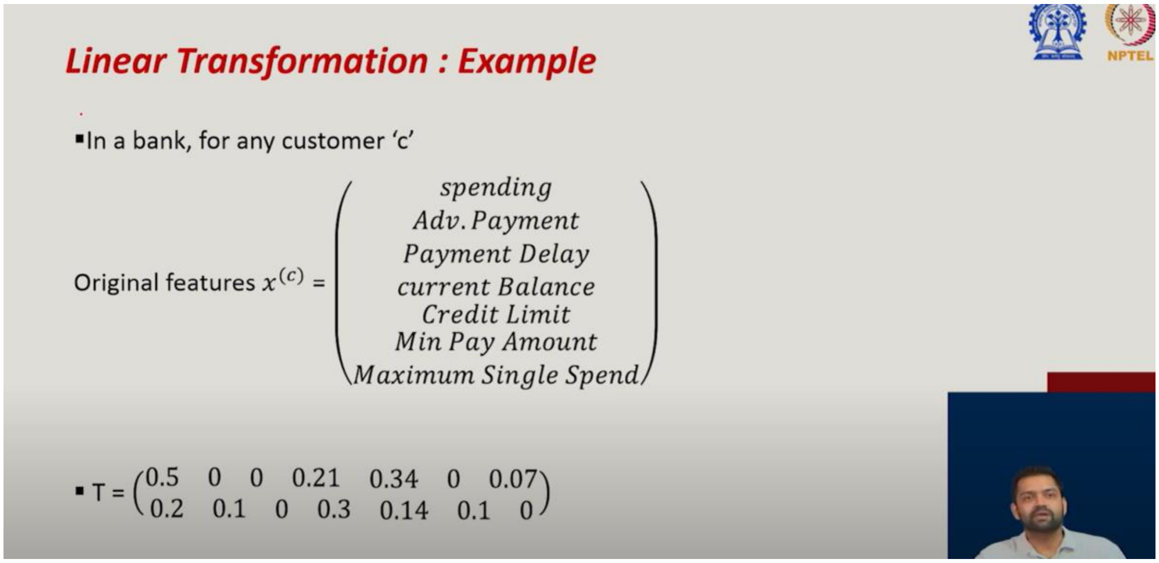
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N \end{bmatrix} \rightarrow \mathbf{y} = \begin{bmatrix} x_{i_1} \\ x_{i_2} \\ \cdot \\ \cdot \\ x_{i_K} \end{bmatrix}$$

reducing dimensionality.

One is feature extraction and the other one is feature selection, ok. Let us understand what they are. What is feature extraction? A feature extraction is, first let's try to understand what is feature selection. A feature selection is simply, as you can see here, let me use the pen. I have a bunch of features, $x_1, x_2, \dots, x_n$.

I'm selecting a few features amongst these. Amongst these features, I'm selecting a few. Let's say I select this and this and this and this. and I get my new set of features so I'm just choosing a subset of the original features like in the example which I was talking about the new financial or the young financial institution example we had seven features characterizing a customer so instead of seven we may simply choose four features from those seven that is feature selection what about feature extraction and this is what we are bothered about in this lecture. What are we doing? We are not selecting a few features from the existing set of features.

We are taking the existing set of features and we are creating a few new features, extracting new features from the existing set of features. And of course, the number of new features we are extracting is far lesser compared to the ah number of original features which were there ok. So, that is feature extraction in feature selection I am selecting a few features from the original set in feature extraction I am extracting out a few new features and the number of ah such new features which are constructed are far lesser in number compared to the original number of features fine. And feature principal component analysis is one such feature extraction technique. So, what are we doing? So, feature extraction in this particular lecture we will only talk about linear combinations.



Linear Transformation : Example

- In a bank, for any customer 'c'

$$\text{Original features } x^{(c)} = \begin{pmatrix} spending \\ Adv. Payment \\ Payment\ Delay \\ current\ Balance \\ Credit\ Limit \\ Min\ Pay\ Amount \\ Maximum\ Single\ Spend \end{pmatrix}$$

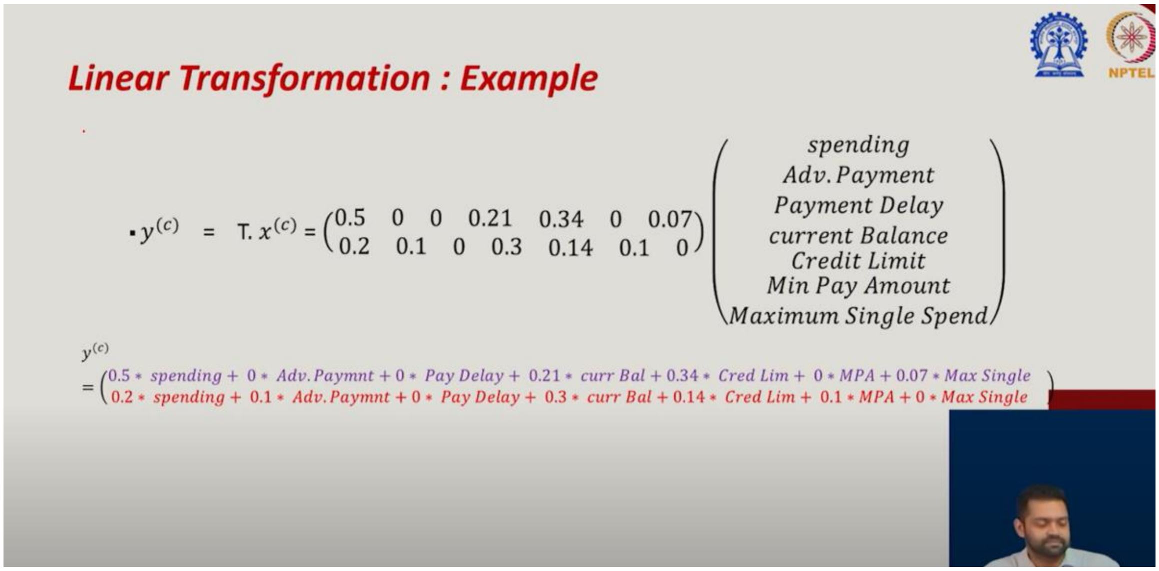
- $T = \begin{pmatrix} 0.5 & 0 & 0 & 0.21 & 0.34 & 0 & 0.07 \\ 0.2 & 0.1 & 0 & 0.3 & 0.14 & 0.1 & 0 \end{pmatrix}$

Of course, there are many ways of extracting features. What are we doing? We are constructing new features from the existing features. So the new features could be any sort of function of the existing features ok. But here we are just we will live in a simple world here. We will simply talk about linear combinations or linear maps that is if $x_1, x_2, \ldots, x_k$ are my original features then a new feature I am extracting y_j is simply a linear combination of the x_i 's which we had ok.

So, let us say x is a n dimensional vector of features. So, in case of a customer, a customer was characterized by 7 features. So, that is a 7 dimensional vector. Now, if we simply pre multiply it with a matrix capital T, where capital T has dimensions $k \times n$ and k is much lesser than n, then $t \times t$ into x will have will be a $k \times 1$ vector. So, y is a $k \times 1$ vector and y happens to be my set of new features.

I will explain it in a second. So, this is my, so this is basically a projection from the n dimensional space to the k dimensional space. I had a 7 dimensional vector representing a customer, now I have a 3 dimensional or a 2 dimensional vector representing a customer. let's see let's take an example in a bank let's consider any customer C what were my original set of features well spending advance payment dot dot dot dot maximum single spend ok so any customer C is characterized by the seven features so I call it $x^{(c)}$ now consider any matrix capital T ok. These numbers mean nothing they are just concoction of my imagination ok .

So, they mean they mean nothing and I just populated it with some numbers ok. So, this is a 2 cross 7 matrix T. If I multiply T with $x^{(c)}$ what will I get? Well I will get 2 new features ok. The purple feature and the red feature which are nothing but each of those features are nothing but a linear combination of the original features. So, now xc was the



Linear Transformation : Example

$$y^{(c)} = T.x^{(c)} = \begin{pmatrix} 0.5 & 0 & 0 & 0.21 & 0.34 & 0 & 0.07 \\ 0.2 & 0.1 & 0 & 0.3 & 0.14 & 0.1 & 0 \end{pmatrix} \begin{pmatrix} spending \\ Adv.Payment \\ Payment\ Delay \\ current\ Balance \\ Credit\ Limit \\ Min\ Pay\ Amount \\ Maximum\ Single\ Spend \end{pmatrix}$$

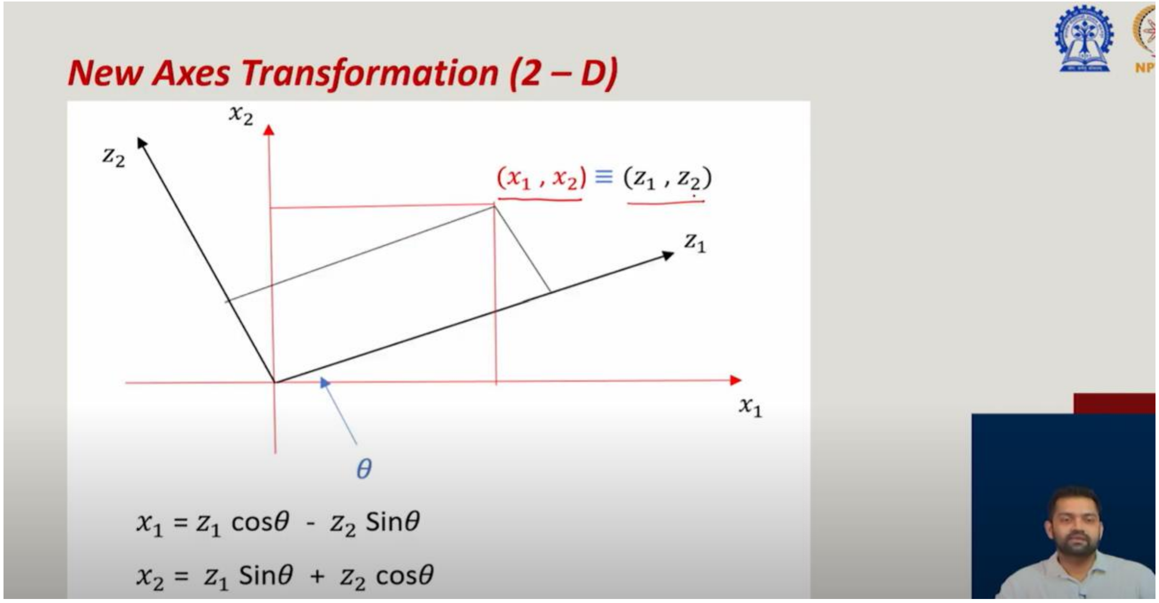
$$y^{(c)} = \begin{pmatrix} 0.5 * spending + 0 * Adv.Paymnt + 0 * Pay\ Delay + 0.21 * curr\ Bal + 0.34 * Cred\ Lim + 0 * MPA + 0.07 * Max\ Single \\ 0.2 * spending + 0.1 * Adv.Paymnt + 0 * Pay\ Delay + 0.3 * curr\ Bal + 0.14 * Cred\ Lim + 0.1 * MPA + 0 * Max\ Single \end{pmatrix}$$

vector characterizing customer c, now we have $y^{(c)}$ characterizing customer c.

And $y^{(c)}$ is only two dimensional, so there are two features. So instead of seven features, now customer C is characterized by two features. Now the question is, fine, we have understood what we are trying to do. What we are essentially trying to do is, we are trying to create new features by simply linearly combining the original features. Like in this example, we had 7 original features, we have linearly combined them and created 2 new features.

But then question is we cannot simply create features randomly. Yes, we are taking linear combinations, but what linear combinations should we take? there could be infinitely many possibilities of linear combinations, right. So, what linear combination should we take in order to satisfy or optimize, but what do we mean by that? What optimum in what sense? Optimum in the sense of information loss, in the sense of minimizing information loss. I have my bunch of features I will take linear combination and construct new features such that the information loss in this process is minimum ok. And that is exactly what we do in principal component analysis.

it is the technique of constructing new features by linearly combining the original features such that minimum information is lost or minimum or maximum variability of the data is captured, ok. Great, let us move on. Let us think 2D and let us try to understand what we mean by that. let's say we have this scatter plot given by the yellow scatter plot which you can see on the screen and we have two original variables x_1 and x_2 okay now instead of x_1 and x_2 if we rotate the axis and we construct two new variables PC1 and PC2 well it seems like PC1 explains most of the variability present in the data So, a trivial example would be something like this. So, let us say you have let me just construct a trivial example just to drive the point home.



New Axes Transformation (2 – D)

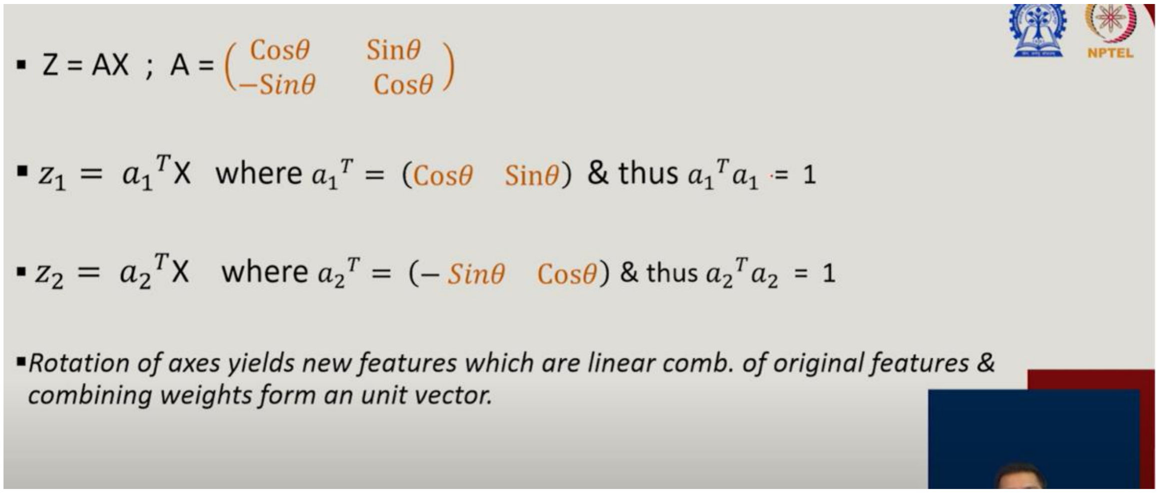
x_1 = z_1 cosθ - z_2 Sinθ

x_2 = z_1 Sinθ + z_2 cosθ

So, let us say you have this is my x_1 and this is my x_2 and let us say we have points like this 1 1, 1 2, So, these are the points. So, this is 1 1 this is 1 1 2 2 dot dot dot dot dot 10 10. Now, instead of construct instead of these 2 features if we simply have this particular feature what is the equation of this line $x_2 - x_1$ equal to 0. So, if we simply have $x_2 - x_1$ okay that gives us in fact that is enough if we know $x_2 - x_1$ instead of knowing both x_1 and x_2 that is good enough right that captures all the variability present in the data great So that's a trivial example where all the variability present in the data is captured by just one variable $x_2 - x_1$ instead of two variables x_1 and x_2. By the way, understand that $x_2 - x_1$ is also a linear combination, right? So this is plus 1 into $x_2 \pm 1$ into x_1, okay.

Now the question is this rotation of axis which we see. like like we saw here P c 1 and P c 2 are two new variables right. So, what is this rotation of axis synonymous with linear combination? Let us see let us see let us say we have let us go back to high school ok high school coordinate geometry. Let us say we have our original variables original coordinate axis given by x_1, x_2 coordinate of the point according to the original coordinate axis is given by x_1, x_2. Now we rotate the coordinate axis by an angle theta and let us according to the new coordinate axis the coordinates are given by z_1, z_2 coordinates of the same point.

Now we have learnt it and it is very easy to verify that x_1 will be $z_1 cos(\theta) - z_2 sin(\theta)$. x_2 on the other hand is going to be $z_1 sin(\theta) + z_2 cos(\theta)$. If we write these two equations in a matrix form we simply get this x_1, x_2 is this particular matrix into z_1 now look at this black matrix which we have this one if you multiply it with this brown matrix we get the identity matrix which means the brown matrix is the inverse of the black matrix right so we can multiply both sides of this equation by the brown matrix and we'll get this



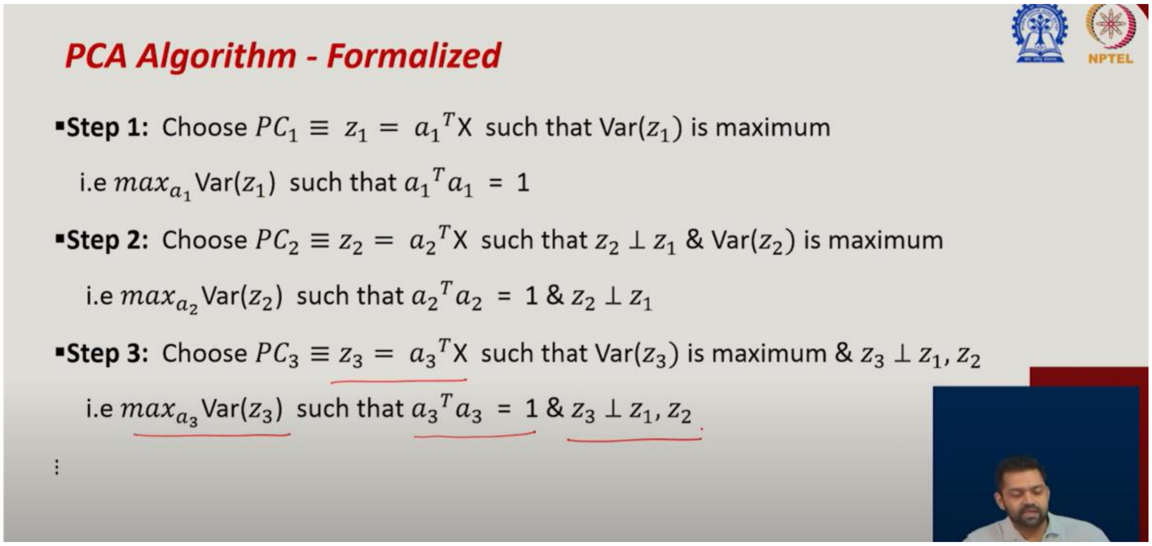
- Z = AX ; A = $\begin{pmatrix} Cos\theta & Sin\theta \\ -Sin\theta & Cos\theta \end{pmatrix}$
- $z_1 = a_1{}^T X$ where $a_1{}^T = (Cos\theta \quad Sin\theta)$ & thus $a_1{}^T a_1 = 1$
- $z_2 = a_2{}^T X$ where $a_2{}^T = (-\,Sin\theta \quad Cos\theta)$ & thus $a_2{}^T a_2 = 1$
- *Rotation of axes yields new features which are linear comb. of original features & combining weights form an unit vector.*

because the black matrix into the brown matrix will become the identity matrix. So, we simply get this or in other words what is z_1 and z_2 then? z_1 is simply as you can see here $x_1 cos(\theta) + x_2 sin(\theta)$. So, z_1 is if x is my original point which is given by x_1, x_2.

So, x is x_1, x_2 then my z_1 is simply $a_1^{tr} x$ which is a linear combination of the constituent variables of x z_2 is $a_2^{tr} x$. also something interesting what is $A_1^{tr} A_1$ and $A_2^{tr} A_2$ both will lead to sin square theta plus cos square theta right and that is equal to 1, okay. So, we have that the rotation of axis yields new features which are nothing but linear combination of the original features such that the combining weights form in unit vector, great. So, we have kind of understood the intuition of rotating axis which we saw in this picture. So, rotation of axis is synonymous or tantamount to synonymous with or tantamount to linearly combining the original features.

So, the next obvious question is what rotation is optimal? You can rotate it by any angle, but what rotation is optimal or what linear combination is optimal? How do we choose this principle components? That is what we are talking about here. So, here is the algorithm. So, how do we do it? We do it in the following manner. The first principle component is chosen such that it points in the direction of largest variance in the data. The second principle component on the other hand is chosen such that it is orthogonal to the first principle component and has the largest variance.

So, the second principle component is or points in the direction of largest variance conditional on the fact that it is orthogonal to the first principle component. What about the third principle component? The third principle component has maximum variance constrained on the fact that it is orthogonal to both the first and the second principle component so on and so forth. So, that is that. So, every principal component points to the direction of largest variance of the residual subspace which the previous such that it is orthogonal to all the previous principal components.



PCA Algorithm - Formalized

- **Step 1:** Choose $PC_1 \equiv z_1 = a_1{}^T\text{X}$ such that Var(z_1) is maximum

  i.e max_{a_1}Var(z_1) such that $a_1{}^T a_1 = 1$

- **Step 2:** Choose $PC_2 \equiv z_2 = a_2{}^T\text{X}$ such that $z_2 \perp z_1$ & Var(z_2) is maximum

  i.e max_{a_2}Var(z_2) such that $a_2{}^T a_2 = 1$ & $z_2 \perp z_1$

- **Step 3:** Choose $PC_3 \equiv \underline{z_3 = a_3{}^T\text{X}}$ such that Var(z_3) is maximum & $z_3 \perp z_1, z_2$

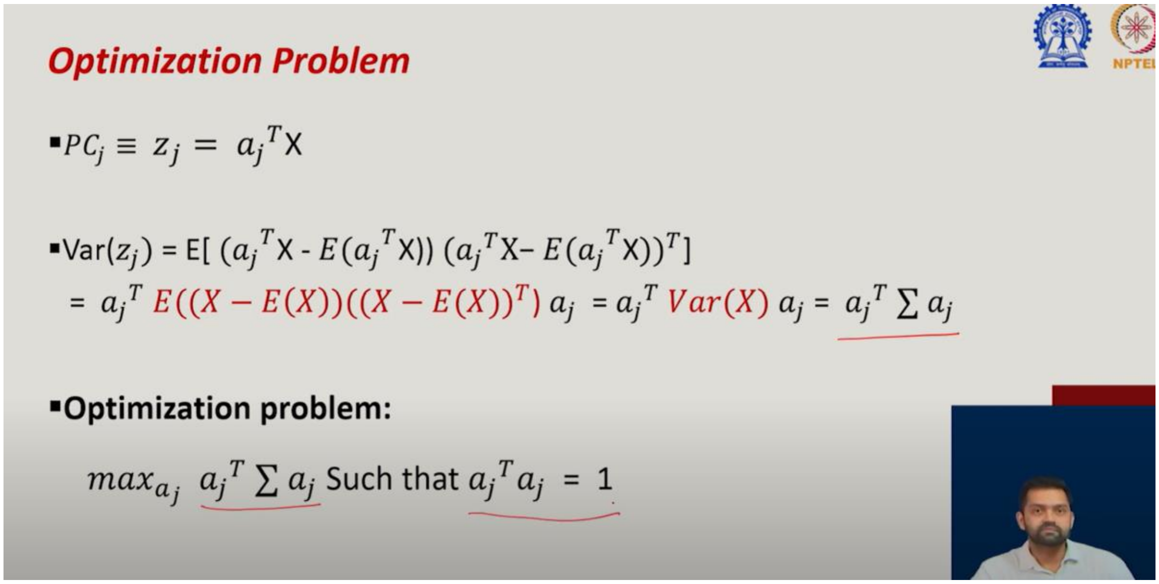
  i.e $\underline{max_{a_3}\text{Var}(z_3)}$ such that $\underline{a_3{}^T a_3 = 1}$ & $\underline{z_3 \perp z_1, z_2}$

⋮

Great, let us move on. So, if this is your data set this is your scatter diagram what would you do? You will choose the direction of maximum variance that seems to be the direction of maximum variance then I am going to choose the direction of maximum variance such that it is orthogonal to this direction which is this direction. So, that is the second principle component in this case we have two dimensions and hence we just have two principle components ok, but I hope I have been able to convey the intuition Great now let us get into the math and formalize what we have learned, let us formalize this intuition mathematically. So, what are we doing? We have our initial set of features capital X, we are finding $A_1^{tr}X$ that is nothing but a linear combination of the constituent variables of X, Z_1 is $A_1^{tr}X$ such that variance of Z_1 is maximum. So, I am choosing my A_1 the combining weights such that variance of Z_1 is maximum given that the combining weights form a unit vector. Then I will find then I choose PC 2 which is the second principle component such that variance of Z_2 is maximum and Z_2 is perpendicular to Z_1.

So, basically I am choosing A_2 such that variance of Z_2 is maximized and we have two constraints here. In case of the first principle component there was just one constraint that the linearly combining weights form a unit vector. Here we have two constraints the linearly combining weights form a unit vector and Z_2 is perpendicular to Z_1 or their covariance is 0. Third principle component I am finding z_3 equal to $A_3^{tr}X$ such that the variance of z_3 is maximized, the linearly combining weights form a unit vector and the third principle component is orthogonal to both the first and the second principle component so on and so forth.

So, let us understand, let us do the math now. consider the j-th principle component. So, what is the jth principle component? It is simply $A_j^{tr}X$. So, if x is $x_1 x_2 A_j$ could be $a_{j1} a_{j2}$ and z_j then becomes ok great. So, we have so this is what z_j is so that is the j principle component. So, this is $z_j a_j^{tr} x$ now what is variance of z_j now x is a x is a vector ok and x



Optimization Problem

- $PC_j \equiv z_j = a_j^T X$

- $\mathrm{Var}(z_j) = \mathrm{E}[\, (a_j^T X - E(a_j^T X))\, (a_j^T X - E(a_j^T X))^T]$

  $= a_j^T\, E((X - E(X))((X - E(X))^T)\, a_j = a_j^T\, Var(X)\, a_j = \underline{a_j^T \sum a_j}$

- **Optimization problem:**

  $max_{a_j}\ \underline{a_j^T \sum a_j}$ Such that $\underline{a_j^T a_j = 1}$

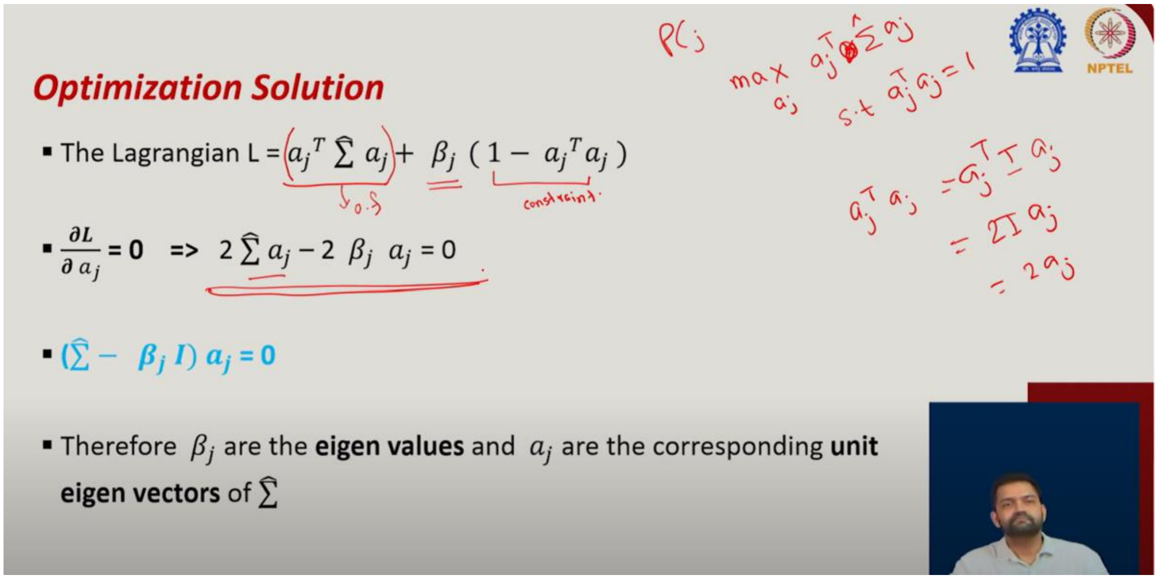
is a feature is a feature vector.

So that is a random variable because I am selecting any particular customer. So what is variance of z_j then? It is simply expectation of $A_j^{tr} x$. Remember from the previous lecture what is variance of any random variable z? It is simply expectation of $(E - E[Z])(E - E[Z])^{tr}$ into z minus expectation of Z^{tr}. Okay? So that's what we are doing here. So this is $A_j^{tr} - E\left[A_j^{tr}\right]$.

This is aj transpose x minus expectation of $A_j^{tr} x - E\left[A_j^{tr} x\right]$. Okay? Great. Now if we simplify this, what will we get? We will get this. If we take A_j^{tr} out, we will simply be left with A_j^{tr} comes out, here A_j comes out, we are simply left with this in between. Now what is this? Just by this definition, it is variance of x, okay.

And let us say variance of x is given by Σ. So variance of z_j is simply $A_j^{tr} \Sigma A_j$ where sigma is the variance matrix of x, x being my feature vectors. So, my optimization problem is simply maximizing $A_j^{tr} \Sigma A_j$ that is the variance of the j th principle component such that the linear combining weights form a unit vector. But we do not know sigma right. but we have the data on x we know all the feature vectors. So, we can compute the sample covariance and sample variance of all the features and sample covariance between any pair of features.

So, we can compute sigma hat which is the sample variance matrix of x. So, my optimization problem then becomes this maximizing the variance of the j-th principle component which is this. such that the linear combining weights form a unit vector, okay. Seems fine? No, I think we missed something.



Optimization Solution

- The Lagrangian L = $\underbrace{a_j{}^T \sum a_j}_{o.f} + \beta_j\,(\underbrace{1 - a_j{}^T a_j}_{\text{constraint.}})$
- $\frac{\partial L}{\partial a_j} = 0 \;\; => \;\; 2\hat{\sum} a_j - 2\,\beta_j\;a_j = 0$
- $(\hat{\sum} - \;\beta_j I)\,a_j = 0$
- Therefore β_j are the **eigen values** and a_j are the corresponding **unit eigen vectors** of $\hat{\sum}$

PCj

$\max_{a_j}\; a_j^T \hat{\Sigma} a_j$

s.t $a_j^T a_j = 1$

$a_j^T a_j = a_j^T I a_j$

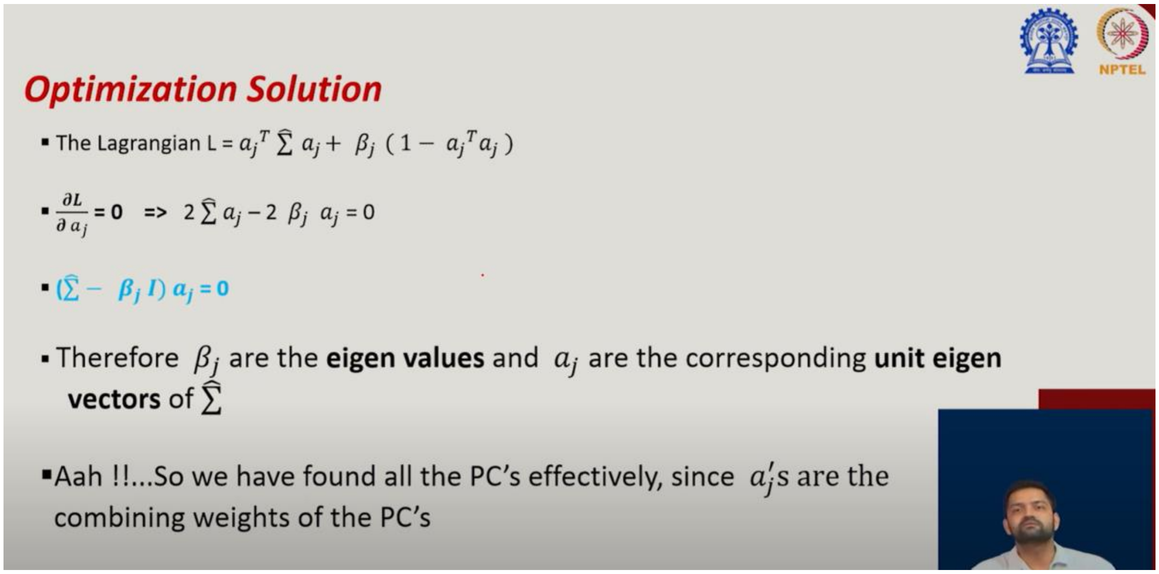
$= 2I a_j$

$= 2a_j$

What did we miss? Pause the video and try to guess. When we define this optimization problem for finding the j-th principle component, what have we missed? We have missed something important. We missed the orthogonality constraints. how did we come up with the principal components the first principal component is chosen such that sorry the second principal component is chosen such that is it maximizes variance and is orthogonal to the first the j-th principal component should be orthogonal to all other principal components right which means so let us say the i-th principal component and the jth principal component So, they should be orthogonal or in other words their covariance should be 0. So, if we simply find covariance of $z_i z_j$ what will that become? If we proceed just like we did it will simply be $A_i^{tr} \Sigma A_j$ and since we do not know Σ we will use its sample analog which is sigma hat.

So it will be $A_j^{tr} \hat{\Sigma} A_j$. All this thing will become clear at the end of the lecture because I will show all the computational steps with an example. And things will become crystal clear. So if something is bothering you, please, please hang on. Right? So coming back to the point.

So this should be 0 for all $i \neq j$. So, these are my set of constraints which should be there. So, when I am finding this j th principle component we should have these constraints in mind that it is orthogonal to all other principle components. but that will make life really difficult finding every principal component such that it is orthogonal to every other principal component. So, it will it will make me work through k or whatever many such constrained optimization problem that is mathematically intractable that is too much ok. So, what should we do? We are in a problem And at times if we land up in a problem in life it is better to just ignore it live in denial and hope that it will go away.



Optimization Solution

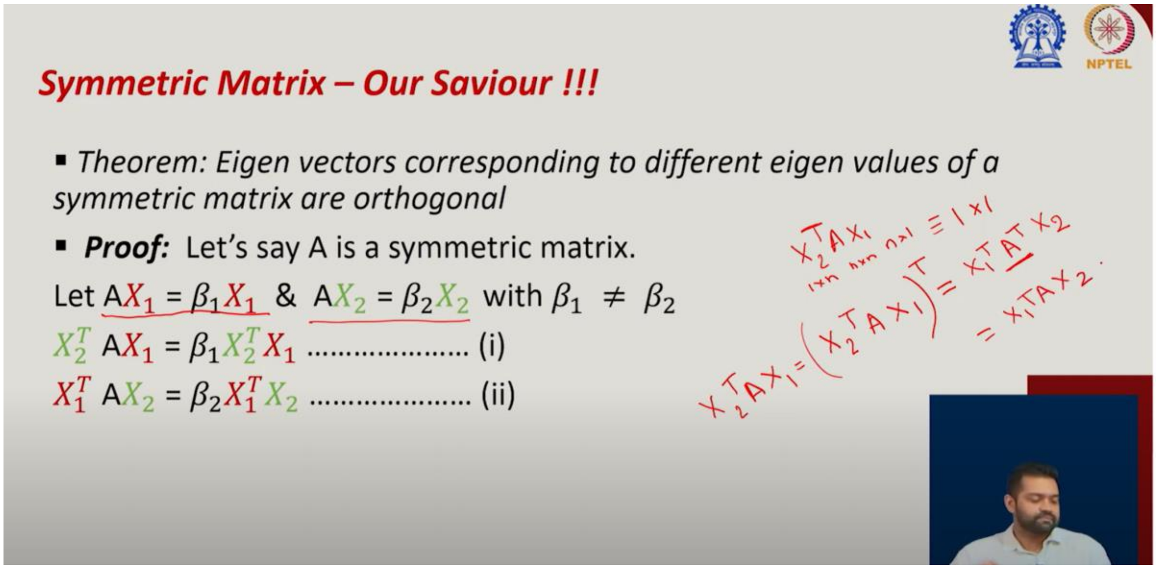
- The Lagrangian L = $a_j{}^T \hat{\sum} a_j + \beta_j \; (1 - a_j{}^T a_j)$
- $\frac{\partial L}{\partial a_j} = 0$ => $2 \hat{\sum} a_j - 2 \; \beta_j \; a_j = 0$
- $(\hat{\sum} - \beta_j I) \, a_j = 0$
- Therefore β_j are the **eigen values** and a_j are the corresponding **unit eigen vectors** of $\hat{\sum}$
- Aah !!...So we have found all the PC's effectively, since a_j's are the combining weights of the PC's

So, that is what we will do we will have faith and we will hope that this problem will go away the problem of missing orthogonality constraints will go away. So, we will go with our faulty optimization problem which is maximizing this. such that. So, our faulty optimization problem for the j th principle component was what maximizing $A_j^{tr}x$ such that sorry $A_j^{tr}\hat{\Sigma}A_j$ such that $A_j^{tr}A_j$ is 1 ok.

So, we will we will we will go with this. and this is simple see as we had talked about in the previous lecture this is an example of a constrained optimization. This is my objective function this is my objective function and this is my constraint $1 - A_j^{tr}A_j$. So, this is my constraint. So, we will define the Lagrangian in this manner l plus this where β_j is my Lagrange multiplier. Now let us differentiate let us differentiate with respect to A_j that is what we are maximizing it over right.

Now look at this do you remember this mathematical form from the previous lecture sigma hat is a symmetric matrix A_j is a vector If you go to the previous lecture, we have done exactly this, a very similar example. So, $A_j^{tr}\hat{\Sigma}A_j$, when it is differentiated with respect to the vector A_j, that should give you $A_j^{tr}\hat{\Sigma}A_j$, this can be written as $A_j^{tr}A_j$, where i is your identity matrix. So, this will become 2 IA_j , which is $2A_j$.

and then there is a β_j so that becomes $2\beta_j A_j$ okay great. Now if we simply rearrange this equation we get the blue equation which is $\hat{\Sigma} - \beta_j$ into i into A_j that is 0 in this case 0 is a null vector. Now recall the previous lecture the mathematical prerequisites lecture. this equation should remind you of something this simply tells you that β_j are the eigenvalues and A_j are the is the corresponding unit eigenvector of $\hat{\Sigma}$ ok great and if that is the case then we are really happy why because we have found the A_j which is the unit eigenvector remember we will have a family of eigenvectors and in that family there is only one unit



Symmetric Matrix – Our Saviour !!!

- *Theorem: Eigen vectors corresponding to different eigen values of a symmetric matrix are orthogonal*
- ***Proof:*** Let’s say A is a symmetric matrix.

Let $\underline{AX_1 = \beta_1 X_1}$ & $\underline{AX_2 = \beta_2 X_2}$ with $\beta_1 \neq \beta_2$

$X_2^T AX_1 = \beta_1 X_2^T X_1$ .... (i)

$X_1^T AX_2 = \beta_2 X_1^T X_2$ .... (ii)

$\underset{1\times n}{x_2^T}\underset{n\times n}{A}\underset{n\times 1}{x_1} \equiv 1\times 1$

$x_2^TAx_1 = (x_2^TAx_1)^T = x_1^T\underline{A^T}x_2$

$= x_1^TAx_2.$

eigenvector right. So, we have found the unit eigenvector A_j and that is that constitutes the linear combining weights of the j th principle component. Wow that is that is pretty good we have found the linear combining weights of the j th principle component which means we have effectively found the j th principle component that is fantastic.

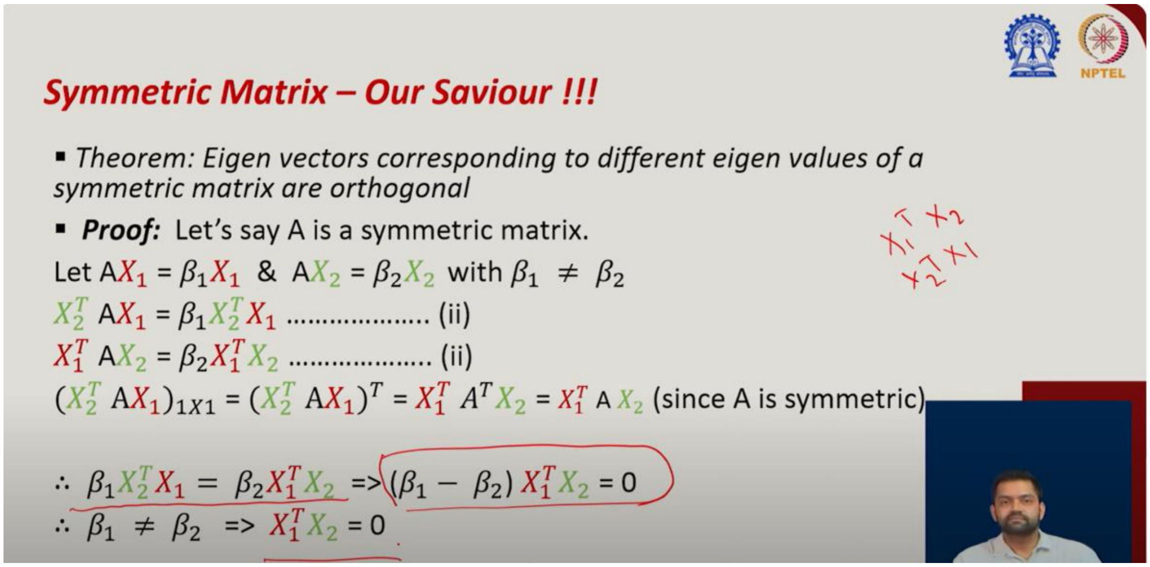
right which is what this is what we set out to do, but we still have that little niggle little confusion little dissatisfaction what we had ignored the missing orthogonality constraints. So, this solution might be erroneous ok, but let us see our savior is linear algebra. There is one particular result about symmetric matrices which will turn out to be our savior.

Let us see. This is the theorem. It says that eigenvectors corresponding to different eigenvalues of a symmetric matrix are orthogonal. Let us quickly breeze through this proof, very simple, hence I included it in the slides. Let us say A is a symmetric matrix. and let us say Ax_1 is $\beta_1 x_1 Ax_2$ is $\beta_2 x_2 \beta_1$ and β_2 are my two eigenvalues and x_1 and x_2 are the corresponding unit eigenvectors let us say. Then if I multiply the first equation by x_2 transpose and the second equation by x_1 transpose we get equation 1 and equation 2, okay.

So this equation I have multiplied both sides by x_2 transpose, this equation I have multiplied both sides by x_1 transpose and we end up with equation 1 and 2. Now look at this $x_2^{tr}Ax_1$. What is the dimension of this? Well $x_2^{tr}A$ is let's say $n \times n$, then x_2 transpose will be $1 \times n$, x_1 is $n \times 1$, so this is 1×1. So $x_2^{tr}Ax_1$ is a scalar. What about transpose of this? Remember if I take transpose of a scalar, I get the scalar itself.

So, this means $x_2^{tr}Ax_1$ should be equal to $x_2^{tr}Ax_1$ transpose. Now, if you take transpose of this you will get $x_1^{tr}A^{tr}x_2$, but A is symmetric. So, A transpose is A. So, this is $x_1^{tr}A^{tr}x_2$

that is what we see.



Symmetric Matrix – Our Saviour !!!

- *Theorem: Eigen vectors corresponding to different eigen values of a symmetric matrix are orthogonal*
- ***Proof:*** Let's say A is a symmetric matrix.

Let $AX_1 = \beta_1 X_1$ & $AX_2 = \beta_2 X_2$ with $\beta_1 \neq \beta_2$

$X_2^T AX_1 = \beta_1 X_2^T X_1$ .... (ii)

$X_1^T AX_2 = \beta_2 X_1^T X_2$ .... (ii)

$(X_2^T AX_1)_{1X1} = (X_2^T AX_1)^T = X_1^T A^T X_2 = X_1^T A X_2$ (since A is symmetric)

$\therefore \underline{\beta_1 X_2^T X_1 = \beta_2 X_1^T X_2} \Rightarrow \boxed{(\beta_1 - \beta_2) X_1^T X_2 = 0}$

$\therefore \beta_1 \neq \beta_2 \Rightarrow \underline{X_1^T X_2 = 0}$

$X_1^T X_2$

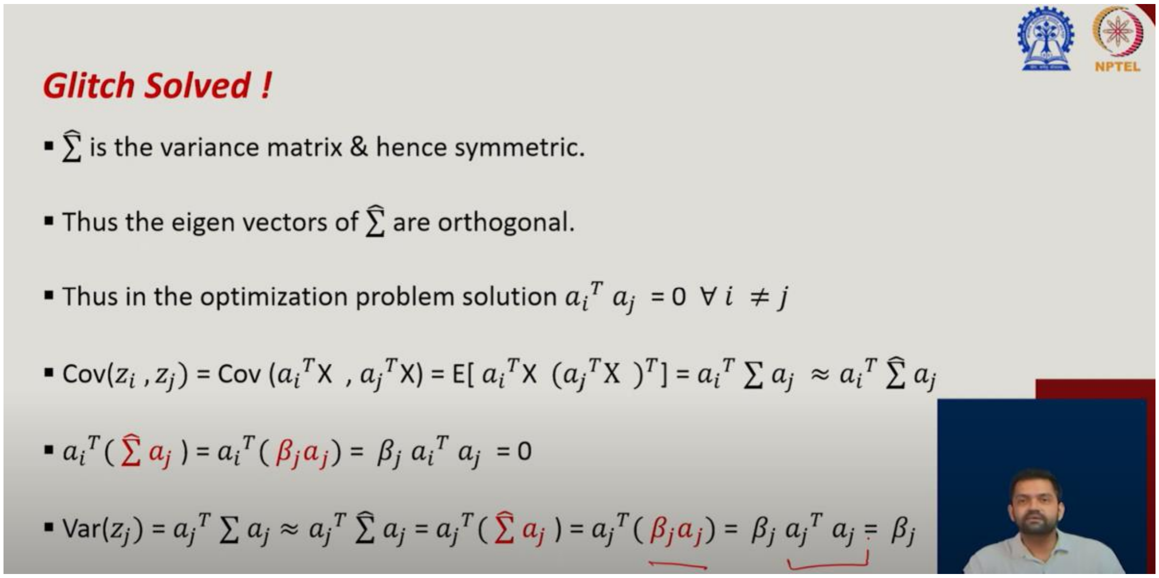
$X_2^T X_1$

So, $x_1^{tr} A^{tr} x_2$ is equal to $x_2^{tr} A x_1$. So, the left hand side of the two equations are equal which means that the right hand sides are also equal. If the left hand sides are equal it means that the right hand sides are equal which means $\beta_1 x_2^{tr} x_1$ and $\beta_1 x_2^{tr} x_1$ these two are equal. but again $x_1^{tr} x_2$ and $x_2^{tr} x_1$ very similarly they are also equal. So, it simply simplifies to this. Now, we assume we started with the assumption that what is the theorem saying eigenvectors corresponding to different eigenvalues different eigenvalues.

So, β_1 and β_2 are different if β_1 and β_2 are different look at this equation β_1 and β_2 are different, when can this equation lead to 0? Only if $x_1^{tr} x_2$ is 0, which in turn implies that x_1 and x_2 are orthogonal, ok. Very good, let us move on. Now, great we learnt this theorem. Now, let us see how it is going to help us resolve the problem which we had. What was the problem remember? We solved the optimization problem for the j th principle component ignoring the constraints, orthogonality constraints that is ignoring the fact that it has to be orthogonal to all other principle components.

Let us see. Now, what is the first important point? $\hat{\Sigma}$ is a variance matrix and hence it is symmetric. Remember we did it in the last class last lecture what was Σ_{ij} it is covariance between x_i and x_j so what is Σ_{ji} covariance between x_j and x_i again so these two are equal so Σ_{ij} and Σ_{ji} are equal right. which means so since $\hat{\Sigma}$ which is the variance matrix since it is symmetric the eigenvectors corresponding to the different eigenvalues of sigma hat are orthogonal, okay. Thus in the optimization problem what was A_i and A_j what did it turn out they were orthogonal and they were the eigenvectors of sigma hat, so they should be orthogonal, okay. Now what is covariance between the ith and the jth principle component? If we proceed in the previous manner it is simply $A_i^{tr} \hat{\Sigma} A_j$ exactly like we did before.

Now what is $\hat{\Sigma} A_j$? A_j is the, A_j is a eigenvector of sigma hat for which the corresponding eigenvalue is β_j that is what we have seen during the Lagrange optimization. so $\hat{\Sigma} A_j$ is simply $\beta_j A_j \hat{\Sigma} A_j$ is simply $\beta_j A_j$ because A_j is an eigenvector and β_j is the corresponding eigenvalue β_j is a scalar it comes out and i have simply $\beta_j A_i^{tr} A_j$ but we have just seen in the theorem which we proved that the eigenvectors corresponding to different eigenvalues are orthogonal. Hence $A_i^{tr} A_j$ is 0. So, this entire thing becomes 0 which in turn means covariance of $z_i z_j$ becomes 0.



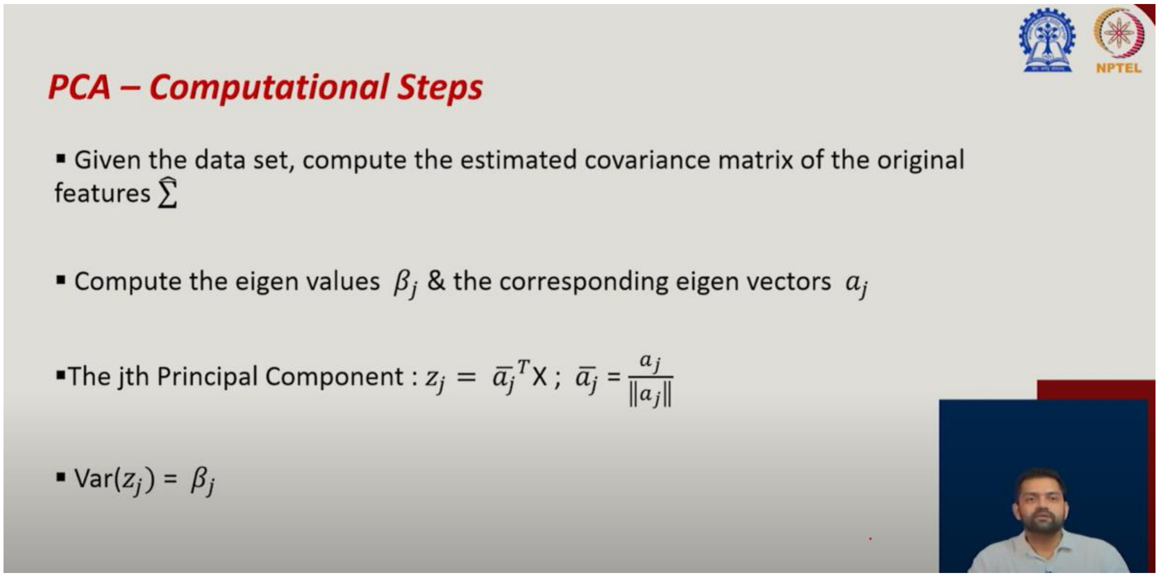
Glitch Solved !

- $\hat{\sum}$ is the variance matrix & hence symmetric.
- Thus the eigen vectors of $\hat{\sum}$ are orthogonal.
- Thus in the optimization problem solution $a_i{}^T\ a_j\ = 0\ \forall\ i\ \neq j$
- $\mathrm{Cov}(z_i\ , z_j) = \mathrm{Cov}\ (a_i{}^T\mathrm{X}\ ,\ a_j{}^T\mathrm{X}) = \mathrm{E}[\ a_i{}^T\mathrm{X}\ \ (a_j{}^T\mathrm{X}\ )^T] = a_i{}^T \sum a_j\ \approx a_i{}^T \hat{\sum} a_j$
- $a_i{}^T(\ \hat{\sum} a_j\ ) = a_i{}^T(\ \beta_j a_j) = \ \beta_j\ a_i{}^T\ a_j\ = 0$
- $\mathrm{Var}(z_j) = a_j{}^T \sum a_j \approx a_j{}^T \hat{\sum} a_j = a_j{}^T(\ \hat{\sum} a_j\ ) = a_j{}^T(\ \underline{\beta_j a_j}) = \ \beta_j\ \underbrace{a_j{}^T\ a_j}\ = \ \beta_j$

Wow that is wonderful that is what we wanted right that is what we wanted. We wanted the i th and the j th principle components to be orthogonal to each other. and it and we those were constraints which had to be imposed, but it turns out that they automatically become orthogonal, the constraints are automatically satisfied. So, we need not bother about them when we are solving the optimization problem, okay. Now, one more what is variance of Z_j the jth principle component remember it is $A_j \hat{\Sigma} A_j$.

that is what we were maximizing if you remember. Now sigma hat a j again by the same thing $\beta_j A_j$, β_j comes out $A_j^{tr} A_j$ remember $A_j^{tr} A_j$ is 1 because A_j is a unit vector we are only talking about unit Eigen vectors. So, it is simply β_j which means that the Lagrange multipliers which we found out in the Lagrange optimization the Lagrange multipliers are the variances of the principal components. And we have found out the Lagrange multipliers right they are the Eigen values of sigma hat great. So, which means we have found out not only the principal components which is which is simply knowing the A_j's once we know all the A_j's we have effectively found all the principal components. We have also found out the variance of every principal component which is given by β_j which is nothing but the Eigen values of the $\hat{\Sigma}$ matrix.

Which means if we know the sigma hat matrix the Eigen vectors will give me the combining weights for the different principle components and the corresponding eigenvalues will give me the variance of that particular principle component right excellent so now we look at an example and we will try to see how to go about this so what is the first step when you have a data set let us summarize what is the first step which you have The first step is you are given the original features from that compute the estimated variance covariance matrix sigma hat find sigma hat. Then find the eigenvalues and eigenvectors of sigma hat in the previous lecture I have explained clearly how to find eigenvalues and eigenvectors of a matrix. So, find the eigenvalues and eigenvectors of sigma hat which are β_j and A_j's. then find all the unit eigenvectors and once you have done that you have found the principal components and the variances of those principal components. The principal components are simply $\bar{A_j}^{tr} X$ where $\bar{A_j}$ is the jth unit eigenvector.



PCA – Computational Steps

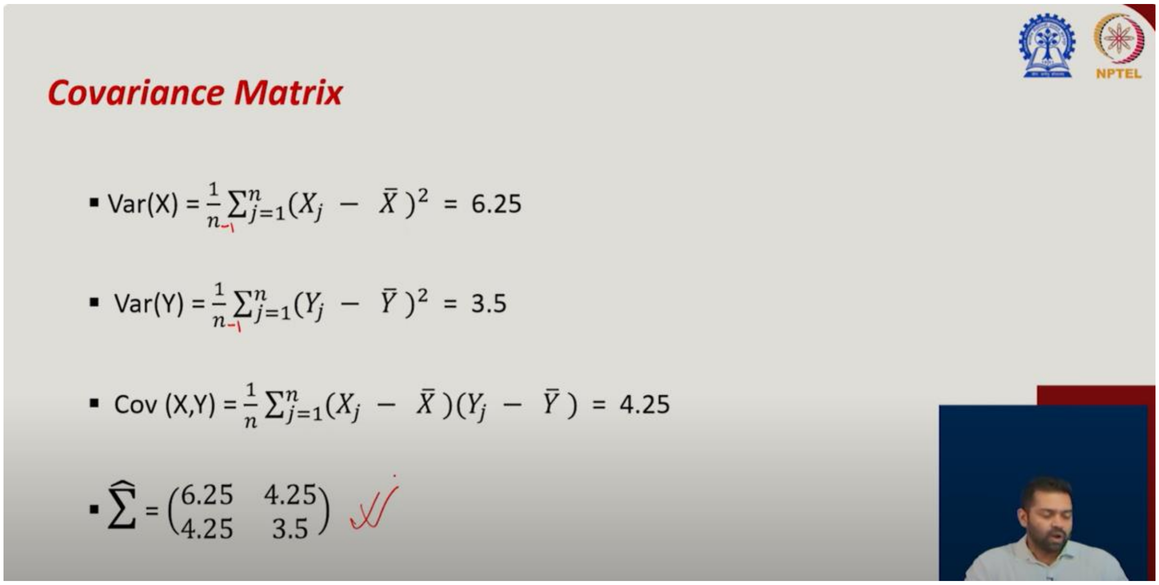
- Given the data set, compute the estimated covariance matrix of the original features $\hat{\Sigma}$
- Compute the eigen values β_j & the corresponding eigen vectors a_j
- The jth Principal Component : $z_j = \bar{a}_j^T \mathrm{X}$; $\bar{a}_j = \frac{a_j}{\|a_j\|}$
- $\mathrm{Var}(z_j) = \beta_j$

Great, let us move on. Let us take an example and let us see instead of doing let us first take an example and see and then we will come back. Let us say we have two variables x and y, x is given by 1, 3, 3, 5, 5 whatever and then we have another variable y 2, 3, 5. If we plot a scatter plot this is how it looks like. Now, can we find principal components? We have two variables in the data set.

can we find principal components let us see. So, what is my first step if you remember I should first compute the covariance matrix let us do that I know the x vector. So, I can find variance of x the sample variance of x which is given by this the sample variance of y given by this actually ideally you should have $n - 1$ here. ok instead of n you should have $n - 1$. So, you have found the variance the sample variance sample covariance ok and thus you have your variance matrix sigma hat. Once you know $\hat{\Sigma}$ then life is easy all you need to do is find the Eigen values and Eigen vectors of $\hat{\Sigma}$.

So we have seen how to find that $\hat{\Sigma}A$ is βA, this is what we have done in the previous lecture sigma hat minus beta into the identity matrix into A is 0. Now we will have a non-null solution only if the determinant of $\hat{\Sigma} - \beta_i$ is 0 or determinant of this matrix is 0 and it turns out we have 2 values β_1 is 9.34 β_2 is 0.41. Now what are these betas we have just proved in the in the previous slides β_j is the variance of the j th principal component ok.

Now what is explained variance of PC 1 it is simply what portion of the total variability of the data is captured by principal component 1. what portion of the total variability of the data is captured by principal component 2, okay. And we see that principal component 1 captures an overwhelming 96 percent of the variability of the data. So, which means that instead of taking x and y, if we just take principal component 1, PC 1, 1 variable. that is that is almost equivalent to taking two variables x and y, just taking PC 1 captures 96 percent of the information or variability present in the data set constituting x and y.

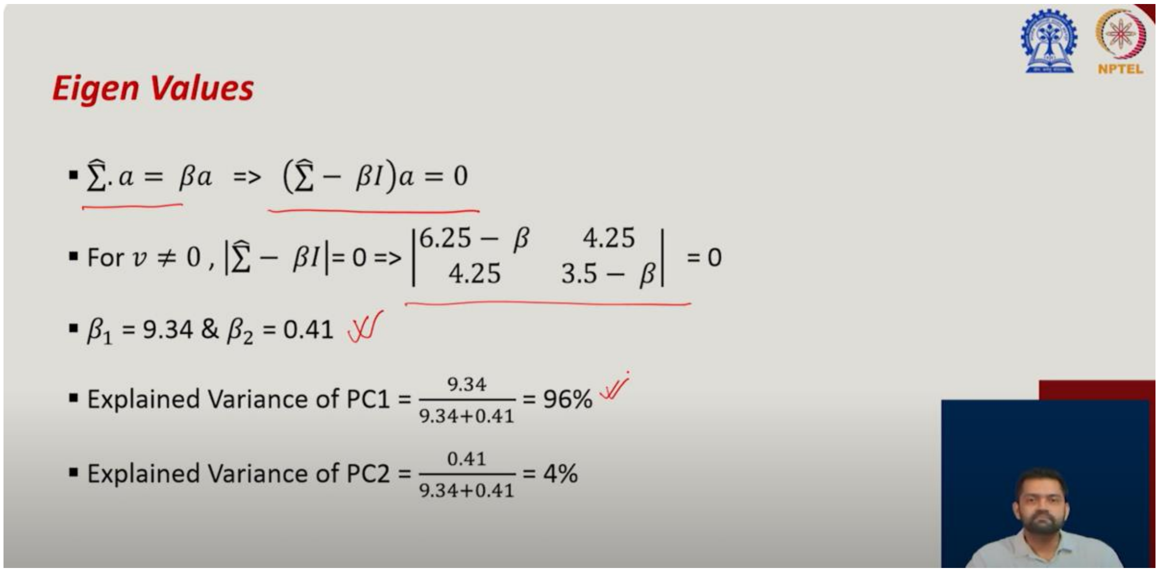


Covariance Matrix

- Var(X) = $\frac{1}{n-1}\sum_{j=1}^{n}(X_j - \bar{X})^2 = 6.25$
- Var(Y) = $\frac{1}{n-1}\sum_{j=1}^{n}(Y_j - \bar{Y})^2 = 3.5$
- Cov (X,Y) = $\frac{1}{n}\sum_{j=1}^{n}(X_j - \bar{X})(Y_j - \bar{Y}) = 4.25$
- $\hat{\Sigma} = \begin{pmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{pmatrix}$

So, let us go ahead and find PC 1, remember we have found the eigenvalue. So, we can find the eigenvectors of sigma hat, hence we can find the unit eigenvectors of sigma hat. once we have found that I can find P c 1, these are my combining weights and this is P c 1, okay. And this is how it looks like, this is the this is the P c 1 vector. So, which means instead of taking x and y 2 variables if we just take P c 1 that is good enough, it captures 96 percent of the information. Now, so we have reduced the dimension from 2 to 1 great now let us go back to that data set which we had in mind which we started off with remember this.

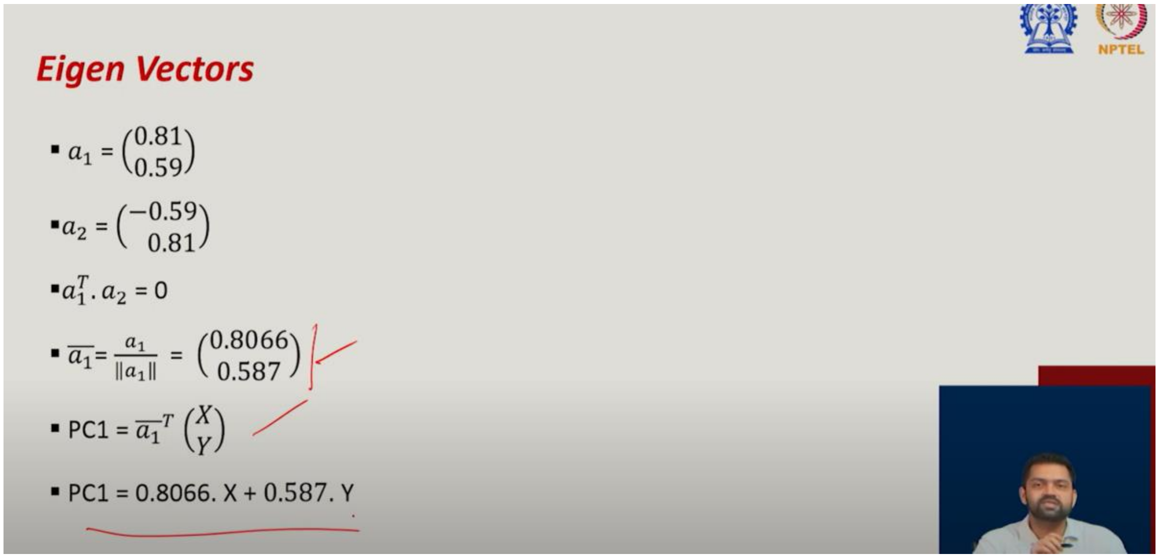
So, let us try to see what we are doing here I will I will try to get into the real thing yeah so what i'm doing is so this is my original data set this is my original data set what i'm doing is the following i'm trying to find all the principal components and then try to i'll try to see taking how many principal components is good enough the choice so understand how to do this so from from sklearn i'm importing this library called pca And this is what I am doing, I have defined an defined a empty array called explain variance, let me explain variance, then I am running a loop from 1 to 7, remember there were 7 features characterizing a customer. And then what am I doing, I am finding the first principal component, see component equal to k you are given, remember. So, in the when k equal to 1, I am finding the loop enters with k equal to 1, I am finding the first principal component and storing it in YPCA and then I am also finding the explained variance ratio. Remember that is what we calculated in the example a few minutes back.



Eigen Values

- $\underline{\hat{\Sigma}.a = \beta a}$ => $\underline{(\hat{\Sigma} - \beta I)a = 0}$
- For $v \neq 0$, $|\hat{\Sigma} - \beta I|$= 0 => $\underline{\begin{vmatrix} 6.25 - \beta & 4.25 \\ 4.25 & 3.5 - \beta \end{vmatrix}}$ = 0
- β_1 = 9.34 & β_2 = 0.41 ✓✓
- Explained Variance of PC1 = $\frac{9.34}{9.34+0.41}$ = 96% ✓✓
- Explained Variance of PC2 = $\frac{0.41}{9.34+0.41}$ = 4%

The explained variance of PC 1 was 96 percent in our example which we did in the slides. So, here for every principal component, I am finding the principal component and I am finding the explained variance ratio of the principal components, ok. Now, I am doing the following. I am plotting the cumulative explained variance and the principal component the number of principal components. So, you can see here the explained variances the first the explained variance of the first principal component is 0.78 that is 78 percent that of the second principal component is 12 percent that of the third principal component is 6 percent.



Eigen Vectors

- $a_1 = \begin{pmatrix}0.81\\0.59\end{pmatrix}$
- $a_2 = \begin{pmatrix}-0.59\\0.81\end{pmatrix}$
- $a_1^T . a_2 = 0$
- $\overline{a_1} = \frac{a_1}{\|a_1\|} = \begin{pmatrix}0.8066\\0.587\end{pmatrix}$
- $\mathrm{PC1} = \overline{a_1}^T \begin{pmatrix}X\\Y\end{pmatrix}$
- $\mathrm{PC1} = 0.8066. X + 0.587. Y$

So, if we cumulatively plot it. So, what is the total way ah variability captured by the first 3 principal components that is 78 plus 12 plus 6 that is a little more than 96 percent maybe 97 percent that is what we see from the diagram 2. If we just take the first 3 principal components the cumulative variance the cumulative variability captured by them is 97 percent ok fantastic. So, which means just taking the first 3 principal components is good enough.

and that is what we do remember we were storing them in YPCA. So, I am sorry. So, this is what we do. So, I am storing these in PC 1, PC 2 and PC 3 ok and I am inserting these principle components in my data set. So now my new dataset looks like this. Initially it looked like what? It was characterized by seven features.

Like this. Spending, advance payment, dot dot dot dot dot. And finally there was, maybe you can see it here. So there were seven features here. But now, Now, instead of these 7 features, I have replaced it with just 3 features, first PC, second PC, third PC. The output variable of course remains the same, defaulter, yet either a defaulter or not a defaulter. So, instead of the explanatory variables I had 7, I have reduced it to 3. I have just found the first 3 principal components and it turns out they capture a 97 percent of the variability.

So, we can move ahead with this instead of the original data set. If we want to predict and we carry out the same thing again, we separate the training set and the test set, we take randomly take 161 observations in the training set, 49 in the test set, we run a model and we see that we get a 83 percent accuracy here. It turns out that if we do it with PCA here, it just gives a little bit of a better performance. Anyway, so I think I have managed to drive home the point of dimensionality reduction using principal component analysis.

This is where we end this particular segment. I hope it's clear, I hope it has been an exciting journey for you. I hope it was nice learning this new technique. Thank you. See you in the next lecture.