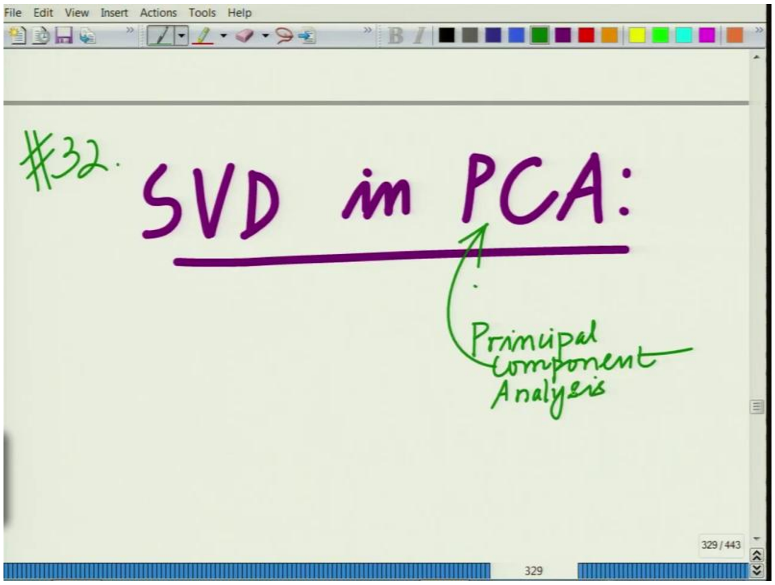
Applied Linear Algebra for Signal Processing, Data Analytics and Machine Learning
Professor Aditya K. Jagannatham
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Lecture 32
SVD Application for Machine Learning Principal Component Analysis (PCA)

Hello, welcome to another module in this massive open online course, so we are looking at the singular value decomposition, let us continue our discussion and let us look at other applications of SVD and in this module in the context of PCA that is principal component analysis, which we have already seen is a very important technique in machine learning.

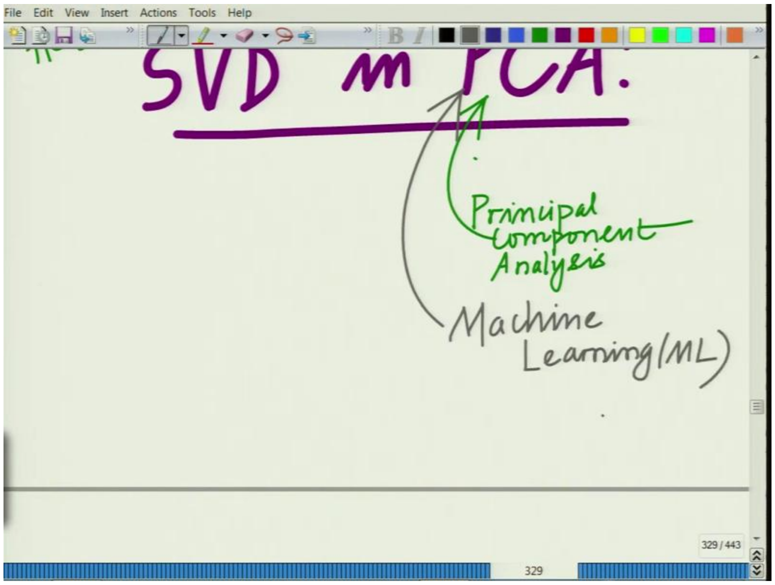
(Refer Slide Time: 00:36)



#32.

SVD in PCA:

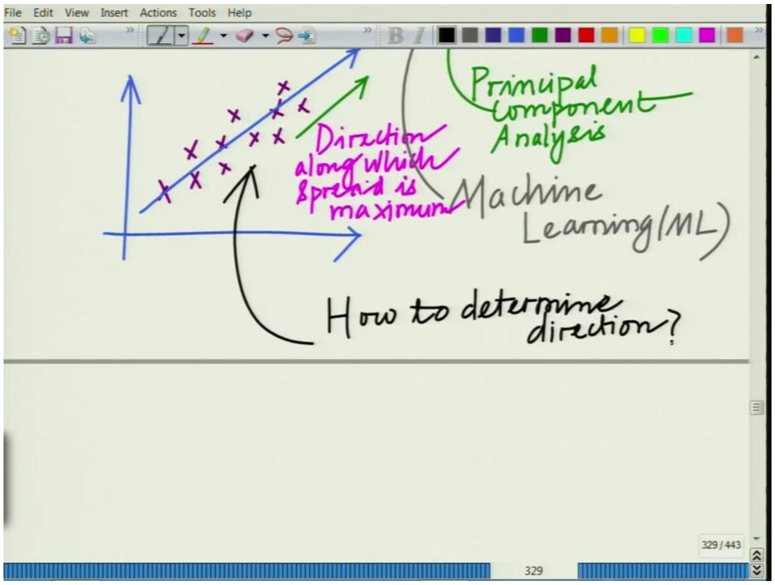
Principal Component Analysis



SVD in PCA:

Principal Component Analysis

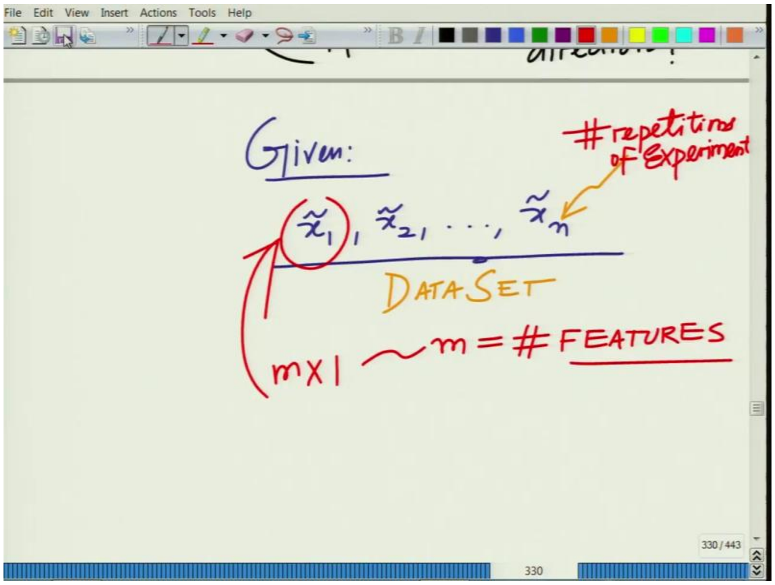
Machine Learning (ML)


File Edit View Insert Actions Tools Help
Principal Component Analysis
Direction along which spread is maximum
Machine Learning(ML)
How to determine direction?
329 / 443
329

So, let us look at an application for of SVD in the context of PCA. And as you are well aware PCA as we have discussed this before this is principal component analysis and this is a very important technique in machine learning, remember to find the principal components of the data how to be essentially compressed the data this is a very important technique in machine learning that is ML.

And what is the relevance and remember you might already remember what is PCA used for if you remember PCA is essentially used to find the principal access or the directions along which the data that is available has the largest spread for instance, let us take a very simple example, so you have data which is like this and if you look at this, you can clearly see this is the direction along which data has a large spread of the data is maximum. So, that is essentially your principal component analysis. And how to determine these directions? That is the question that you asked, how to determine this direction along which the spread is maximum?

(Refer Slide Time: 03:36)



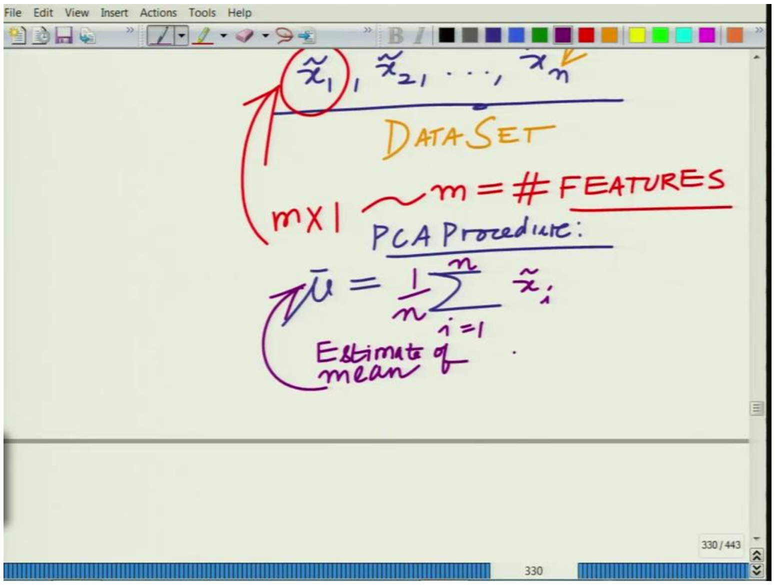
Given:

$\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n$

repetitions of Experiment

DATA SET

$m \times 1 \sim m = \#$ FEATURES



$\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n$

DATA SET

$m \times 1 \sim m = \#$ FEATURES

PCA Procedure:

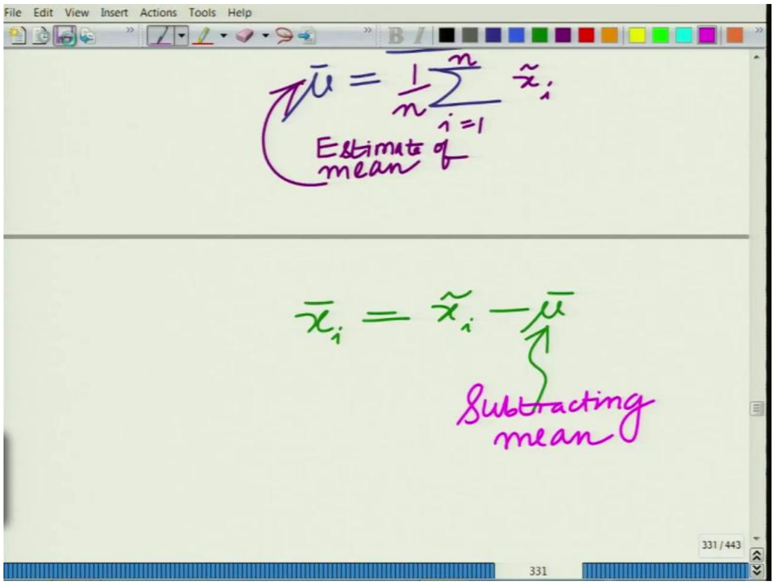
$$\bar{\mu} = \frac{1}{n}\sum_{i=1}^{n} \tilde{x}_i$$

Estimate of mean

And the answer to that is the following thing that is given the data x1 tilde, x2 tilde, xn tilde this is our data set, you can think of n as basically the number of repetitions of an experiment, this is the number of repetitions of an experiment and each vector let us say is of size m cross 1 and m is the number of features, so you have n vectors each of size m, n can be thought of as the number of repetitions of an experiment that has been carried out and m is the number of feature in each particular experiment that have been observed.

Now the PCA proceeds as follows the PCA procedure if you remember recall the PCA procedure first you determine the PCA procedure, now what is the PCA procedure? First you determine the mean the mean is summation I equal to 1 to n x tilde i 1 over n this is essentially the mean estimate of the mean rather this is the estimate of the mean and now you subtract the mean, the mean adjusted data.

(Refer Slide Time: 05:52)



$$\bar{\mu} = \frac{1}{n}\sum_{i=1}^{n} \tilde{x}_i$$

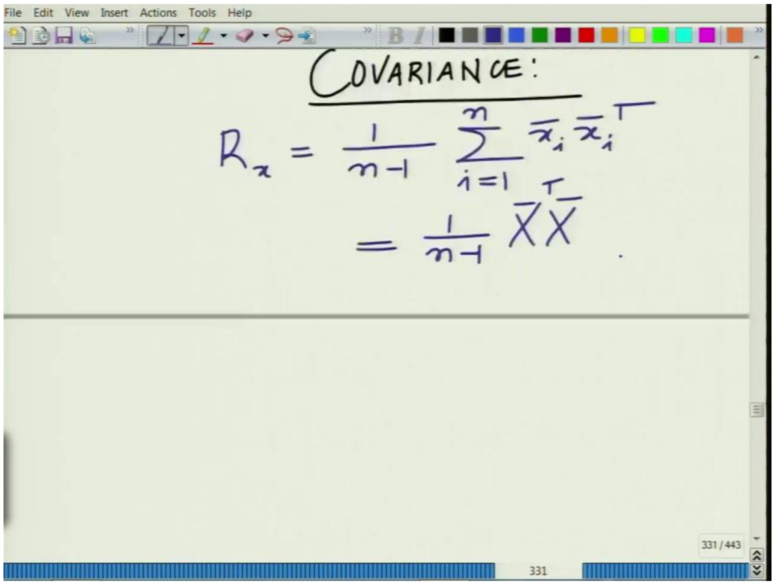
Estimate of mean

$$\bar{x}_i = \tilde{x}_i - \bar{\mu}$$

Subtracting mean

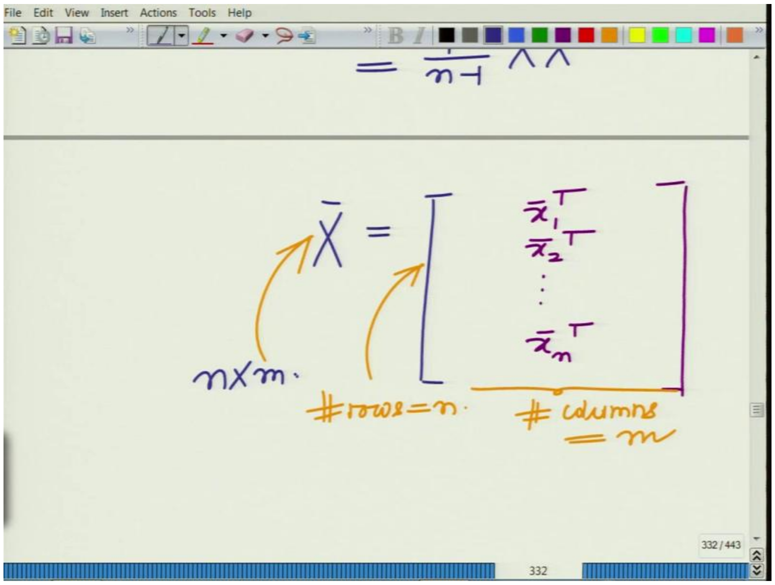
Define xi bar equal to xi tilde minus mu bar. So, what we are doing is essentially we are subtracting the mean, so we are subtracting the mean and now the next step in the PCA you might remember is to estimate the covariance of this data, so the estimate of the covariance matrix.

(Refer Slide Time: 06:31)



COVARIANCE:

$$R_x = \frac{1}{n-1}\sum_{i=1}^{n} \bar{x}_i \bar{x}_i^T$$
$$= \frac{1}{n-1} \bar{X}\bar{X}^T$$



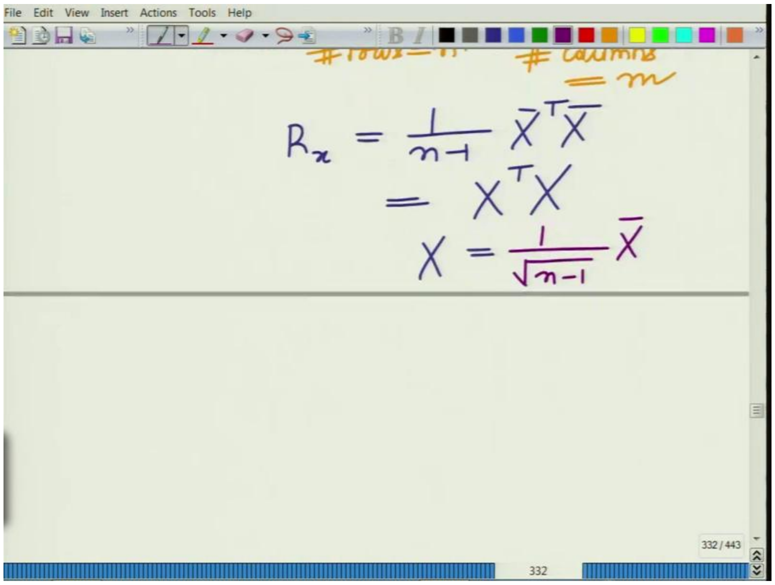
$$= \frac{1}{n-1} \wedge\wedge$$

$$\bar{X} = \begin{bmatrix} \bar{x}_1^T \\ \bar{x}_2^T \\ \vdots \\ \bar{x}_n^T \end{bmatrix}$$

n×m.

#rows = n.

columns = m



#rows = n. # columns = m

$$R_x = \frac{1}{n-1}\bar{X}^T\bar{X}$$

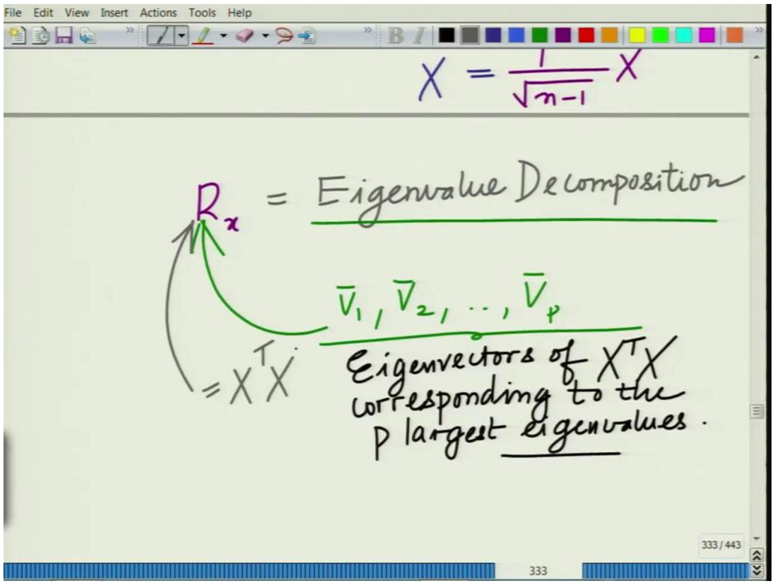
$$= X^T X$$

$$X = \frac{1}{\sqrt{n-1}}\bar{X}$$

So, the covariance this is given as the follows, this is R of x, this is equal to 1 over n minus 1 summation j or i equal to 1 to n x bar i x bar i transpose which I can write as 1 over n minus 1 the matrix X bar transpose X bar where X bar you can see is the matrix this will be the matrix x1 bar transpose, x2 bar transpose, xn bar transpose, so this is you can clearly see this is number of rows is equal to n and number of columns is equal to number of columns is equal to 2 equals basically m.

So, this is essentially what is this? This is n cross m matrix, so this is X bar and therefore I can write Rx as 1 over n minus 1 X bar transpose X bar which is I can also write this as X transpose X where X equal to 1 over n minus 1 X bar, X equal to X bar over square root of n minus 1.

(Refer Slide Time: 08:53)



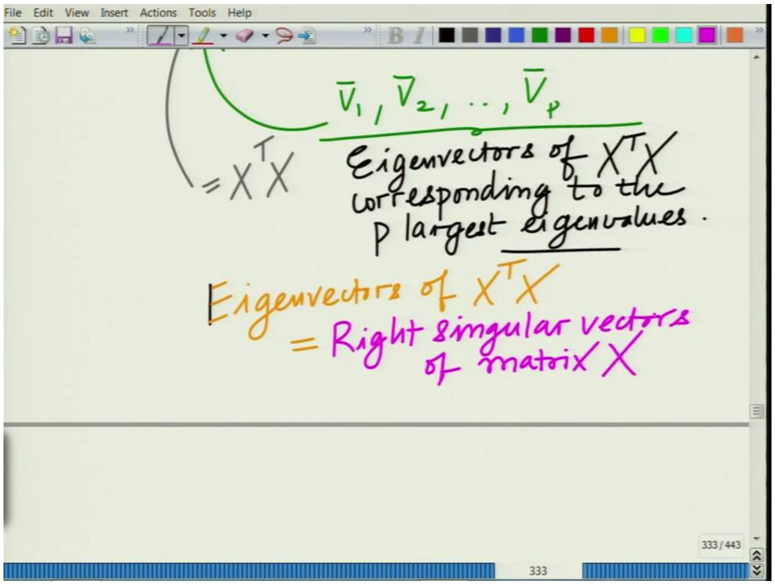
$$X = \frac{1}{\sqrt{n-1}} X$$

$$R_x = \text{Eigenvalue Decomposition}$$

$$= X^T X$$

$\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_p$

Eigenvectors of $X^T X$ corresponding to the p largest eigenvalues.



$\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_p$

$= X^T X$

Eigenvectors of $X^T X$ corresponding to the p largest eigenvalues.

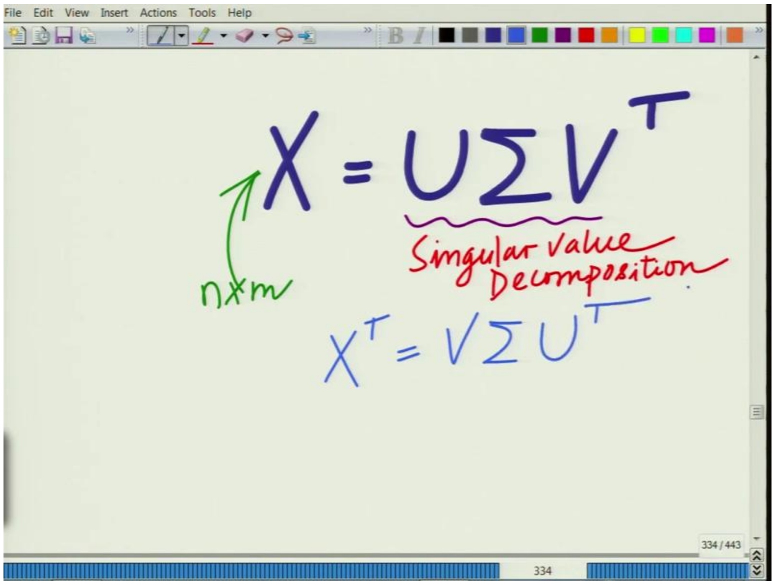
Eigenvectors of $X^T X$ = Right singular vectors of matrix X

And then what we do is essentially the next step in the PCA is to perform the eigenvalue decomposition of Rx remember recall that performed the eigenvalue decomposition of Rx and take the p Eigenvectors corresponding to the largest Eigenvalues. So, you perform the Eigenvalue, so we perform what is the next step?

Next step R we perform the Eigenvalue decomposition and consider v1 bar, v2 bar vp bar where these are essentially the Eigenvectors of X transpose X corresponding to the p largest Eigenvalue corresponding and here you will note that Rx now recall Rx, Rx is equal to X transpose X. So, we are looking at the Eigenvectors of X transpose X.

But remember the eigenvectors of X transpose X are nothing but the right singular vectors of the matrix X, this is an important property. So, the property that we want to realize here is Eigenvectors of X transpose X equal to the right singular vector these are the right singular vectors of the matrix X.

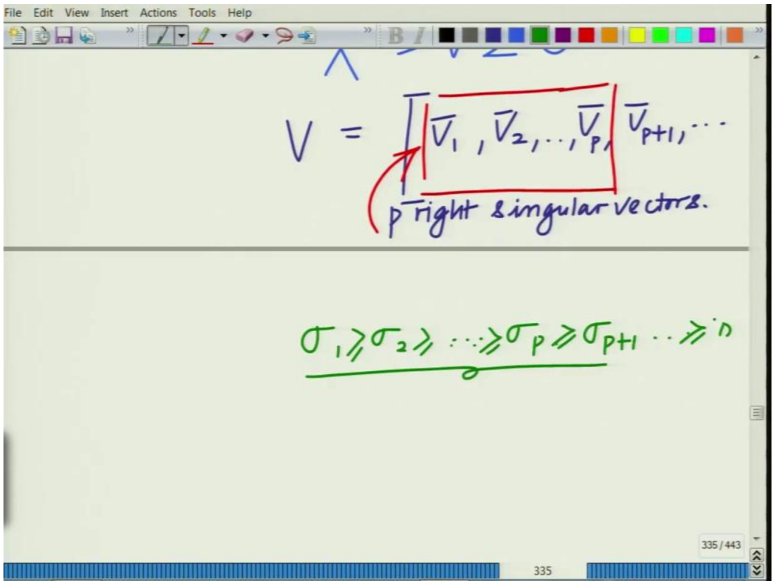
(Refer Slide Time: 11:40)



$$X = \underbrace{U \Sigma V^T}_{\text{Singular value Decomposition}}$$

$n \times m$

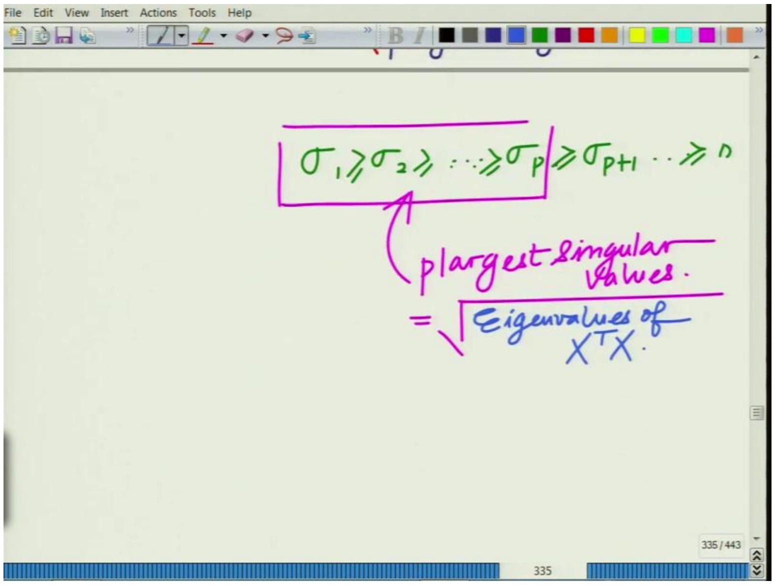
$$X^T = V \Sigma U^T$$



$$V = \left[\boxed{\bar{V}_1, \bar{V}_2, \ldots, \bar{V}_p,} \; \bar{V}_{p+1}, \ldots \right.$$

p right singular vectors.

$$\underline{\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p} \geq \sigma_{p+1} \cdots \geq \sigma_n$$



$$\boxed{\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p} \geq \sigma_{p+1} \cdots \geq 0$$

p largest singular values.

$= \sqrt{\text{Eigenvalues of } X^T X}$.

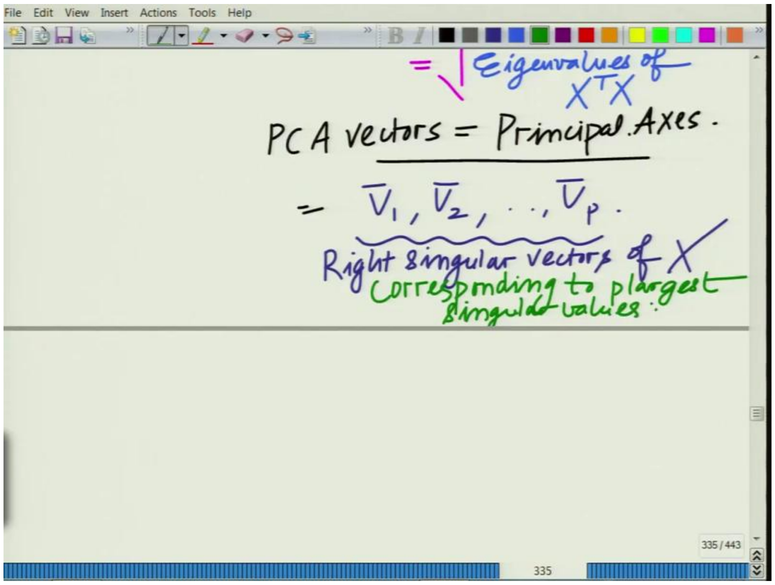
So, instead of looking at the eigenvalue decomposition of X transpose X I can take the matrix X and look at the singular value decomposition of X, so I can directly without considering X transpose X I can consider X equals U sigma V transpose that is I can directly look at the singular value decomposition of the data matrix X, remember recall, this is your n cross m matrix, this is your n cross m matrix x bar is this is an n cross n matrix and therefore now what I have is I basically have a singular value decomposition.

And now consider the right singular vectors and if you look at X transpose X transpose equals V sigma U transpose and now let us look at this matrix V which contains the right singular vectors v1 bar v2 bar so on up to v bar p then you have v bar p plus 1 so on.

And we consider now this vectors the right singular vector correspond the p right singular vectors corresponding to the largest singular values consider now the p right singular vectors corresponding to the largest singular values, because remember sigma the singular values are arranged in decreasing order so we have sigma 1 greater than equal sigma 2 greater than is equal to sigma 3 greater than equal to sigma p plus 1 and all of them are in turn greater than equal to 0.

So, we have the p largest singular values which are nothing but the Eigen square root of Eigenvalues, so these are the p largest singular value which are equal to the square root which are basically the square root of Eigenvalues of X transpose X.

(Refer Slide Time: 14:58)



$= \sqrt{\text{Eigenvalues of } X^T X}$

PCA vectors = Principal Axes .

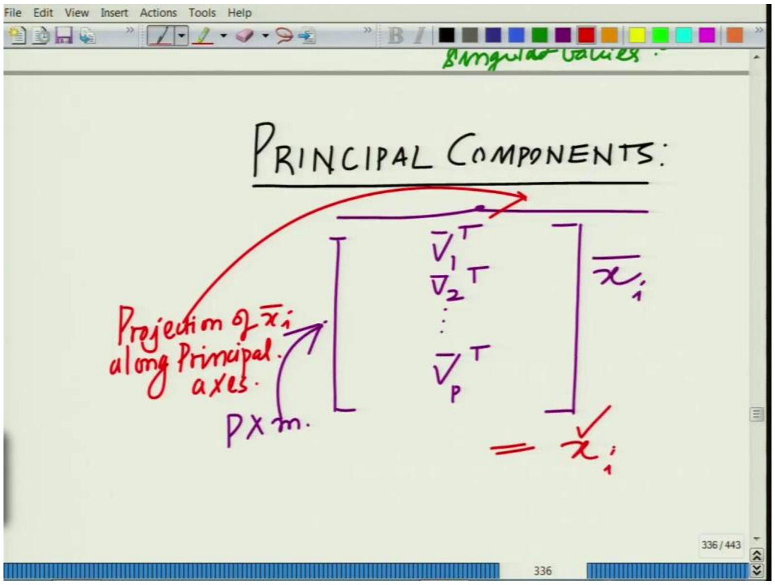
$$= \underbrace{\bar{V}_1, \bar{V}_2, \ldots, \bar{V}_p}_{\text{Right singular vectors of } X} .$$

corresponding to p largest singular values .

And therefore now the principal axis right the PCA vectors you can also the principal axis these are given by the vectors v1 bar v2 bar up to v1 bar. Now, what are these? These are the right singular vectors all of the matrix X corresponding to the larger corresponding to the p largest singular values and these are the principal axis.

And therefore the principal components are given by the projection of the data vectors along this principle axis. So, you take how do you get the principal component? So, to obtain the principal components take the projection of the data vector each data victor x bar i along these principal axis. So, how to obtain the principal components?

(Refer Slide Time: 16:36)



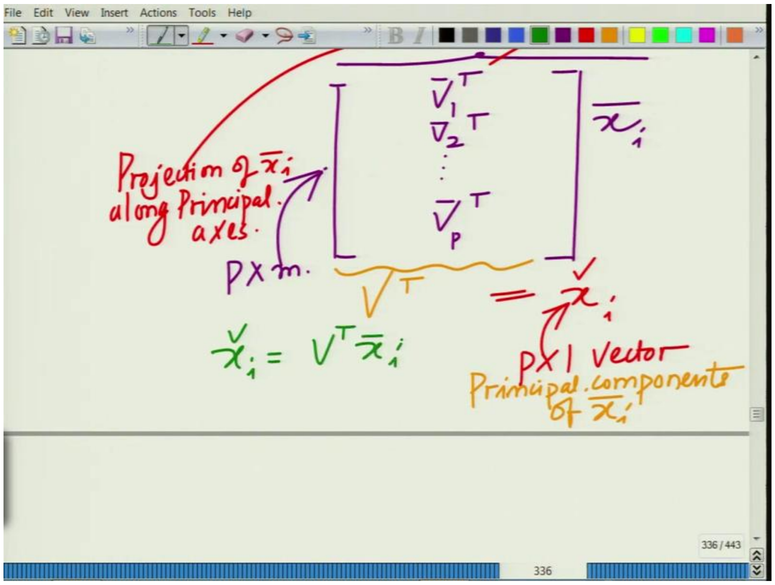
Singular values :

PRINCIPAL COMPONENTS:

$$\begin{bmatrix} \bar{v}_1^T \\ \bar{v}_2^T \\ \vdots \\ \bar{v}_p^T \end{bmatrix} \bar{x}_i = \check{x}_i$$

Projection of $\bar{x}_i$ along Principal axes.

p x m.



$$\underbrace{\begin{bmatrix} \bar{v}_1^T \\ \bar{v}_2^T \\ \vdots \\ \bar{v}_p^T \end{bmatrix}}_{V^T} \bar{x}_i = \check{x}_i$$

Projection of $\bar{x}_i$ along Principal axes.

p x m.

$\check{x}_i = V^T \bar{x}_i$

p x 1 Vector

Principal components of $\bar{x}_i$

Now, the principal components for each vector if you remember these are given by the principal components are given by you take the matrix of principal axis that is v1 bar transpose v2 bar transpose so on and so forth, vp bar transpose, this is essentially this is a p cross m matrix and take the projection or x bar i, so you are forming the projection, so this operation is essentially projection of x bar i along the principal access and this is equal to remember the notation that we was this x check i and x check i is essentially this is a p cross 1 vector containing the principal copy cross 1 vector this contains the principal components of xi.

And more specifically the p principal components of xi x bar i which are the projections, which are basically obtained by the projection of x bar i along the principal axis, the directions that have the largest spread which are basically v1 bar v2 bar vp bar. And you can call this matrix as the matrix basically you can call this as the matrix V transpose so you can write this principal component vector x check i is nothing but you can write this as x check i equals V transpose times x bar, where what is V transpose?

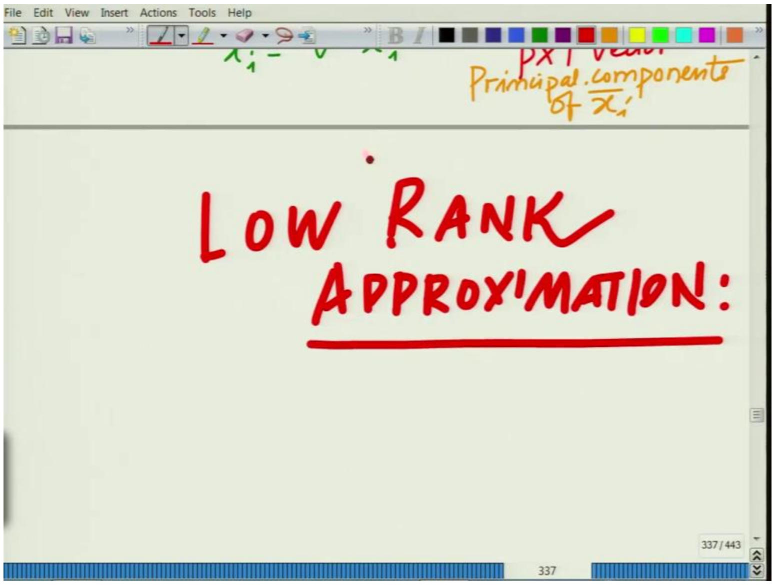
V transpose comprised of v1 bar transpose, v2 bar transpose, vp bar transpose, which are where v1 bar, v2 bar vp bar are the right singular vectors of the matrix X corresponding to the largest singular value. So, that is the use of SVD in the principal component analysis and I have already told you the principal component analysis one of the most important techniques that is used for data reduction or data compression and machine learning.

Because typically in machine learning what happens is that data vectors that you have, have an enormous size for instance you have an image, which is let us say even a modest image has 256 cross 256 pixels which means you have 2 to the power of 8, 2 to the power of cross 2 to the power of 8 that is 2 to the power of 16 pixels.

So, that is the data vector of size 2 to the power of 16, which means now to store this and processes requires huge complexity. So, to compress that for instance and application or just facial recognition one can therefore use PCA to extract the principal components and then one can directly deal with the principal components rather than dealing with the entire factor.

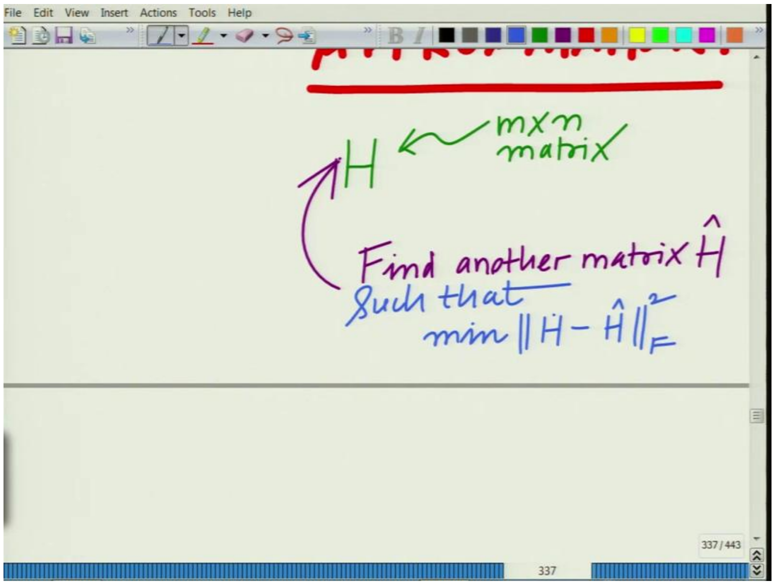
So, that is an important application of so that is so that makes PCA a very relevant in machine learning and just directly makes SVD therefore very very important in the context of PCA and naturally the context of machine learning. Let us, look at another very interesting application of SVD and that is to develop a low-rank approximation of a matrix.

(Refer Slide Time: 21:06)



p×1 vector
Principal components of $\bar{x}_i$

LOW RANK APPROXIMATION:



$H \leftarrow$ m×n matrix

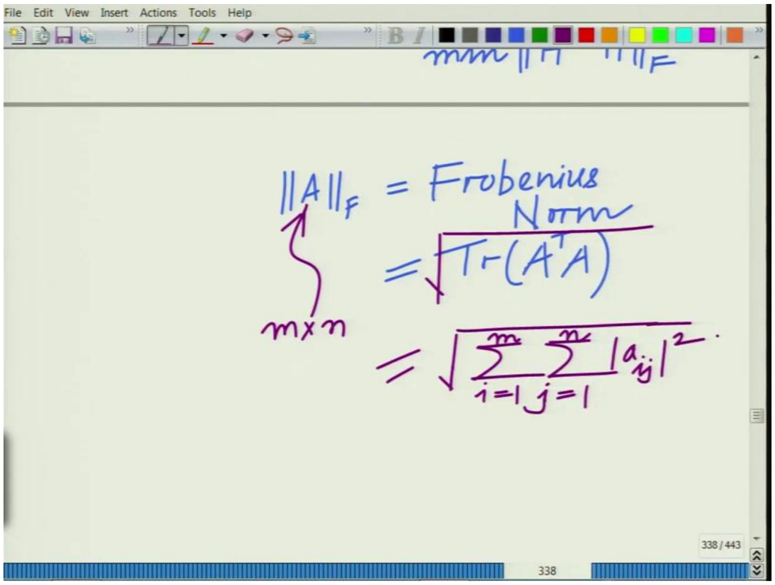
Find another matrix $\hat{H}$

Such that

$\min \|H - \hat{H}\|_F^2$

So, one of the other examples that we want to look at is basically in the context of a lower rank approximation, we want to develop a low-rank approximation, what do we mean by that? Given a matrix H, let this be an m cross n matrix. Now, we would find, we want to find another matrix H hat, such that we want to minimize the norm square and this is the Frobenius norm minimize norm of H minus H hat square.

(Refer Slide Time: 22:25)



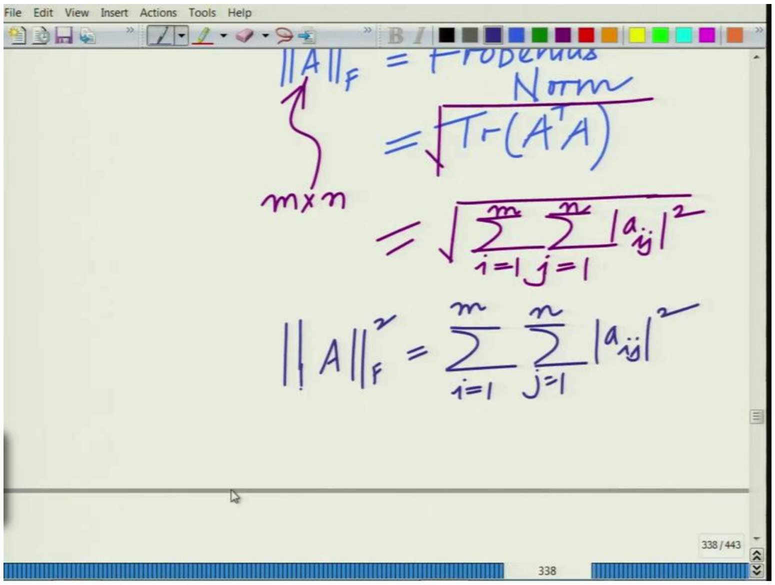
$\min \| H \quad \|_F$

$\|A\|_F$ = Frobenius Norm

$m \times n$

$= \sqrt{Tr(A^T A)}$

$= \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$.



$\|A\|_F$ = Frobenius Norm

$m \times n$

$= \sqrt{Tr(A^T A)}$

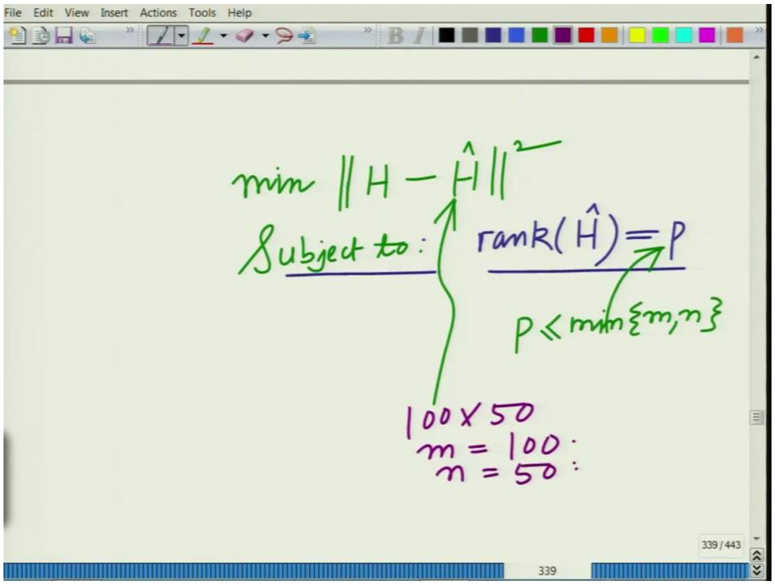
$= \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$

$$\|A\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2$$

Now, in case you are wondering what is the Frobenus norm this norm of A bar F square this is the matrix Frobenius norm which for a matrix real matrix we have already seen this is nothing but trace of A transpose A that is essentially the extension of those Euclidean norm, sorry square root of trace of A transpose A which is essentially if you look at it basically the Euclidean norm to the matrix is that is square root of i equal to 1 for an m cross n matrix, square root of i equal to 1 to m square root of j equal to 1 to n magnitude aij square take the sum of the magnitudes squares of the all the elements and then take the square root.

Naturally the square of the Frobenius norm will be simply the square magnet some of the magnitude square all elements of the A so the A F square this will be sum sum i equal to 1 to m j equal to 1 to n magnitude aij square. So, therefore we want to develop now we want to develop the low-rank approximation.

(Refer Slide Time: 24:03)



$\min \|H - \hat{H}\|^2$

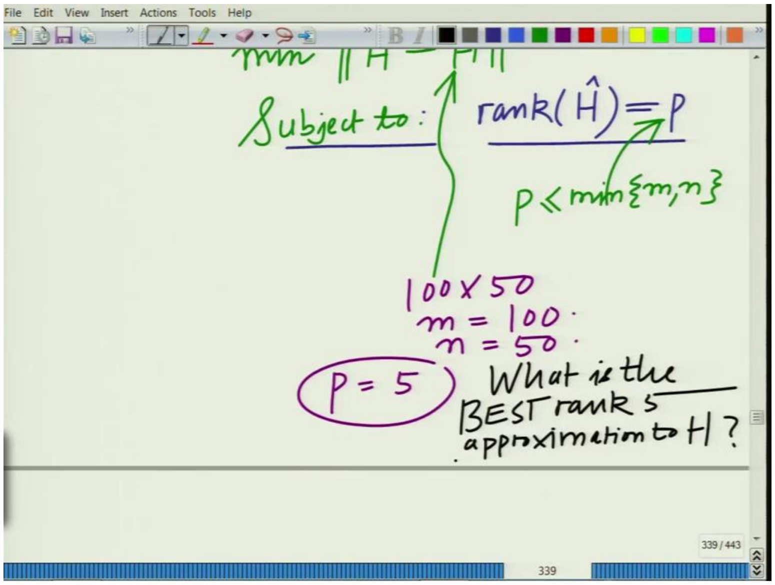
Subject to: $\mathrm{rank}(\hat{H}) = p$

$p \ll \min\{m,n\}$

100×50

$m = 100$

$n = 50$



$\min \|H - \hat{H}\|$

Subject to: $\mathrm{rank}(\hat{H}) = p$

$p \ll \min\{m,n\}$

100×50

$m = 100$

$n = 50$

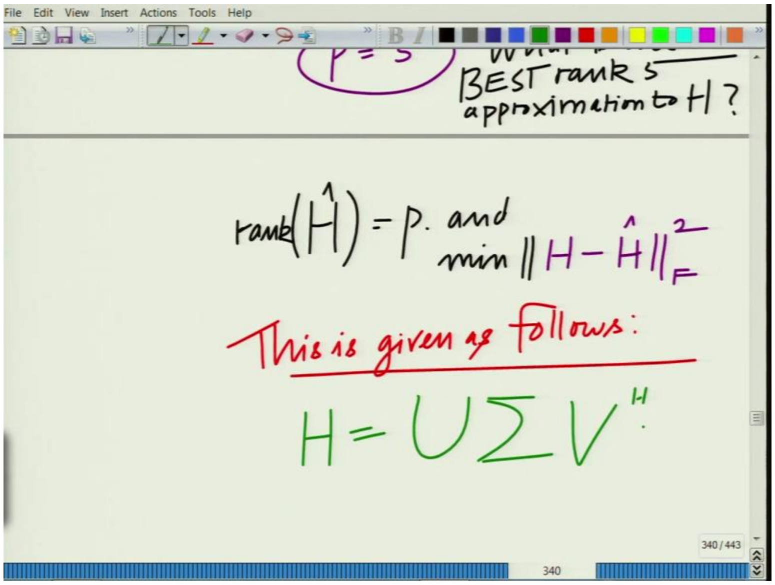
$\boxed{p = 5}$ What is the BEST rank 5 approximation to H?

What we mean is we want to find another matrix such that given a matrix H such that H minus H hat is minimum so minimize norm H minus H hat subject to the constraint, what is this constraint? And this is a very interesting constraint and like some anything we have seen before rank of H hat we want to restrict the rank of H hat to p that is subject to the constraint and p is less than equal to of course to make this problem meaningful p has to be less than equal to minimum of m common n, reason being because the rank is anyway minimum of m comma n less than equal to minimum of m comma n.

If p is greater than minimum of m comma n then H itself is the best p rank approximation is the best low rate of H itself is the best low-rank approximation to I mean H itself is the best low-rank approximation to H. So, we want to consider it consider the cases where the rank of H is greater than p for instance, let us say we have a 100 cross 100 matrix, example let us take a simple example. Let us, take a simple example, let us say we have a 100 cross 50 matrix that is m equal 100 and n equal to 50.

And we want to ask the question, what is the best at now let us set p equal to 5 and we want to ask the question what is the best rank 5 approximation to H? What is the best prank p? So, p equal to 5. So, what is the best rank 5 approximation that is what is the best matrix H hat such that H hat has rank 5 and norm H minus H hat square is minimum or norm H minus H hat, one of the same.

(Refer Slide Time: 26:47)



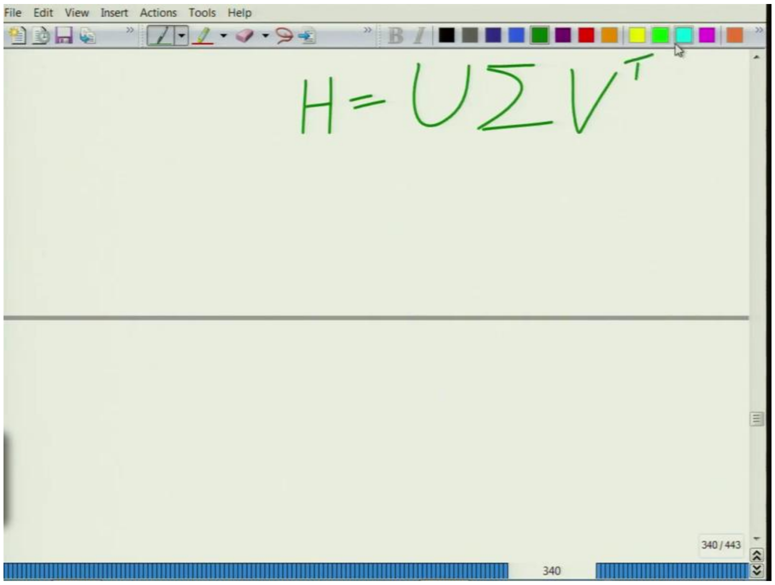
$p = 5$

BEST rank 5 approximation to H ?

$\text{rank}(\hat{H}) = p.$ and $\min \|H - \hat{H}\|_F^2$

This is given as follows:

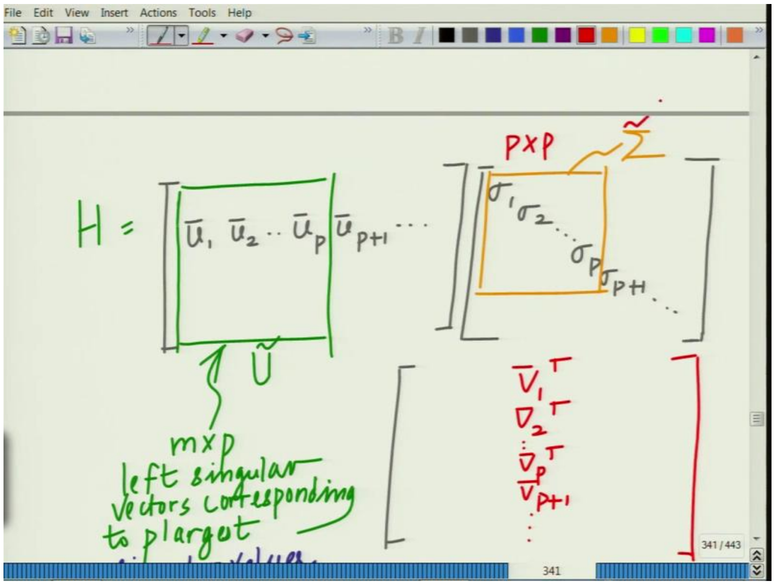
$H = U \Sigma V^H$:



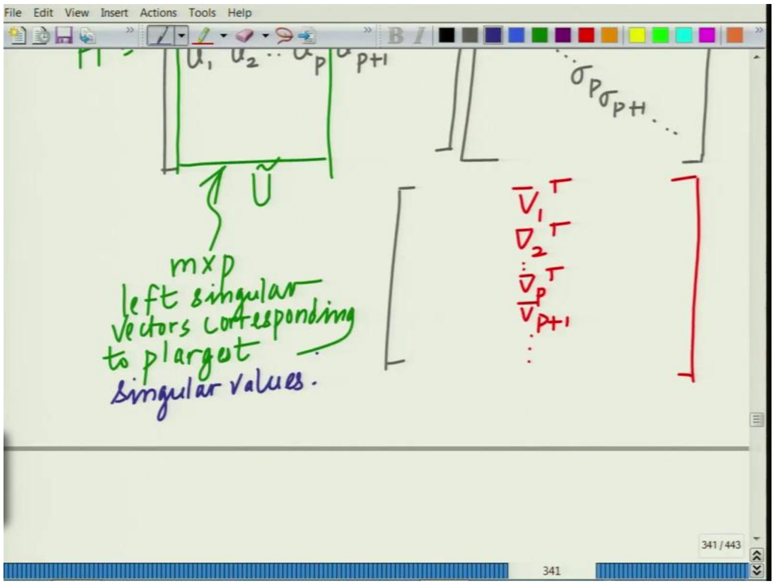
$$H = U\Sigma V^T$$

So, H hat has so you have rank H hat equals p and you have minimum norm H minus H hat or another way to look at it is think of from the set of all matrices of rank 5, through the set of all matrices of rank 5 find the matrix that has that is the best approximation to H. And the answer to that is very simple the answer to that is incredibly simple, so we take this is given as follows this can be achieved as follows start with the SVD of H. So, once again we start with the SVD of H U sigma V Hermitian or if you want to look at real matrices U sigma V transpose.

(Refer Slide Time: 28:05)


File Edit View Insert Actions Tools Help
p×p
$\tilde{\Sigma}$
$\tilde{U}$
m×p
left singular vectors corresponding to p largest singular values
341 / 443
341

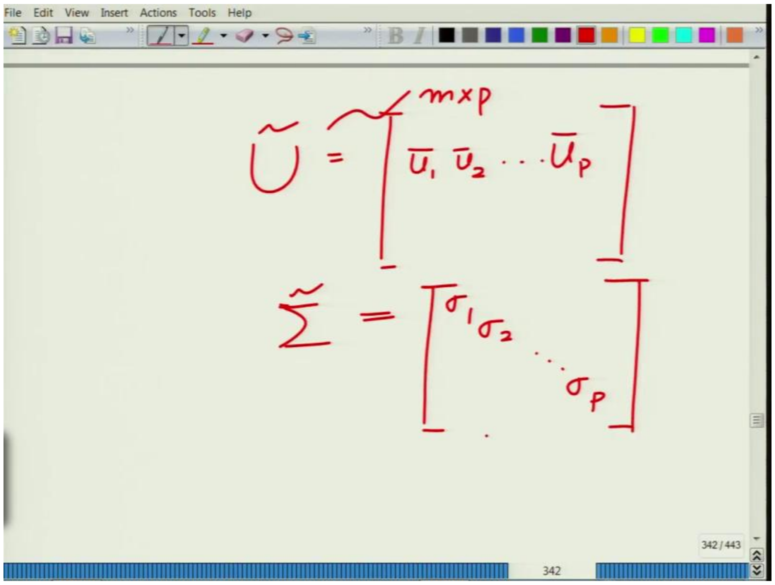
$$H = \begin{bmatrix} \bar{u}_1 & \bar{u}_2 & \cdots & \bar{u}_p & \bar{u}_{p+1} & \cdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \sigma_p & \\ & & & & \sigma_{p+1} \\ & & & & & \ddots \end{bmatrix} \begin{bmatrix} \bar{v}_1^T \\ \bar{v}_2^T \\ \vdots \\ \bar{v}_p^T \\ \bar{v}_{p+1} \\ \vdots \end{bmatrix}$$


File Edit View Insert Actions Tools Help
$u_1, u_2 \cdots u_p \mid u_{p+1}$
$\sigma_p \sigma_{p+1} \cdots$
$\tilde{U}$
$m \times p$
left singular vectors corresponding to p largest singular values.
$\bar{v}_1^T$
$\bar{v}_2^T$
$\bar{v}_p^T$
$\bar{v}_{p+1}$
341 / 443
341

And essentially you can write it in this fashion you can write it as the columns u1 bar u2 bar u bar p u bar p plus 1 and so on and then we have the singular value sigma 1 sigma 2 sigma p sigma p plus 1 and so on, we do not care what about the what the rest are and we have v1 bar transpose v2 bar transpose vp bar transpose v bar p plus 1 so on and now the best approximation rank p approximation is given as follows.

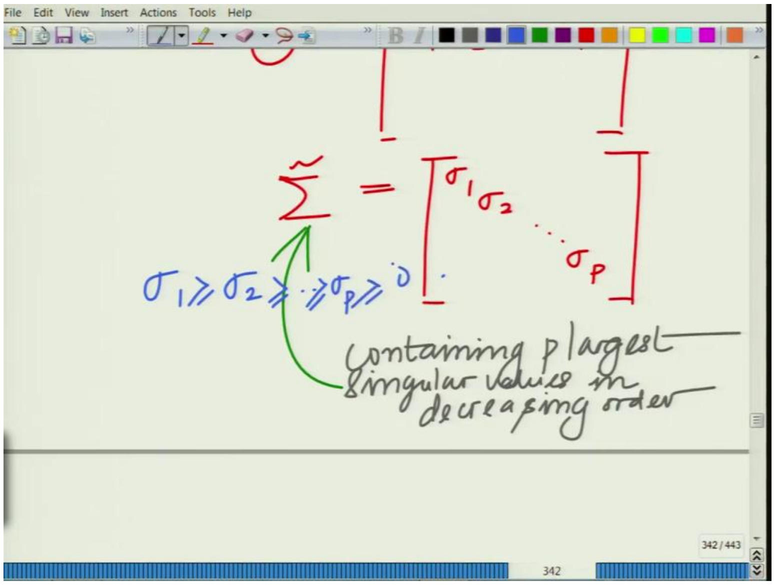
We take the left sub matrix of H that is we take the left, we can consider the left m cross p sub matrix let us call this as U tilde, what is U tilde? It contains the left singular vectors corresponding to the largest singular values, corresponding to the p largest singular values and then we take the matrix the diagonal matrix this is sigma this will be your p cross p matrix containing the largest singular value sigma 1 sigma 2 sigma tilde, let us call this sigma tilde.

(Refer Slide Time: 30:34)



$$\tilde{U} = \overset{m \times p}{\begin{bmatrix} \bar{u}_1 & \bar{u}_2 & \cdots & \bar{u}_p \end{bmatrix}}$$

$$\tilde{\Sigma} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_p \end{bmatrix}$$



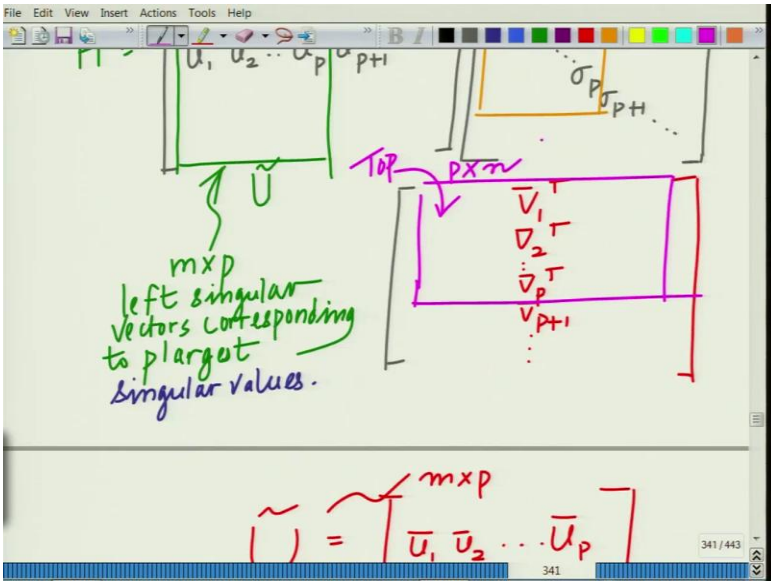
$$\tilde{\Sigma} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_p \end{bmatrix}$$

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$

containing p largest singular values in decreasing order

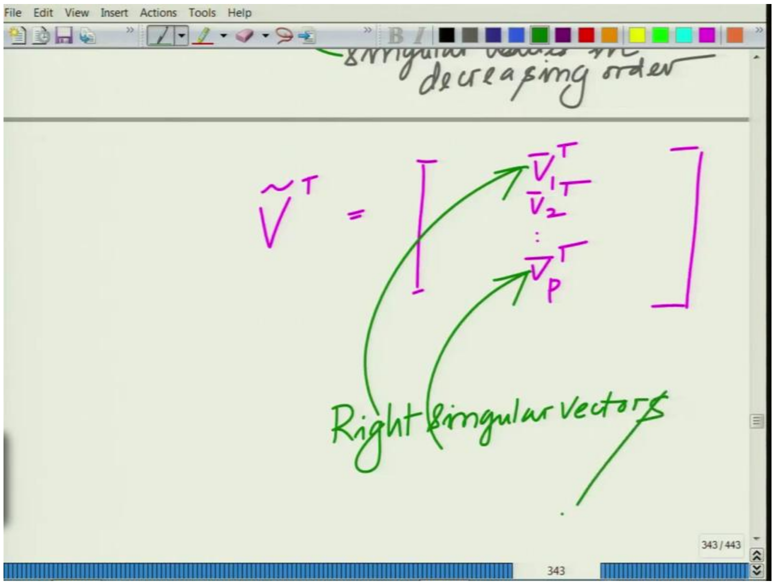
So, sigma tilde is basically so your U tilde is basically your u1 bar u2 bar up to up bar this is your m cross p matrix corresponding to the right singular vector containing right singular left singular vectors corresponding to the p largest singular values sigma tilde is your matrix that is diagonal matrix sigma 1 sigma 2 sigma p, containing the p largest singular values in decreasing order remember we already have sigma 1 greater than equal to sigma 2 greater than equal to sigma p which is in turn all of these are non-negative intern greater than equal to 0, greater than equal to sigma p intern greater than equal to 0.

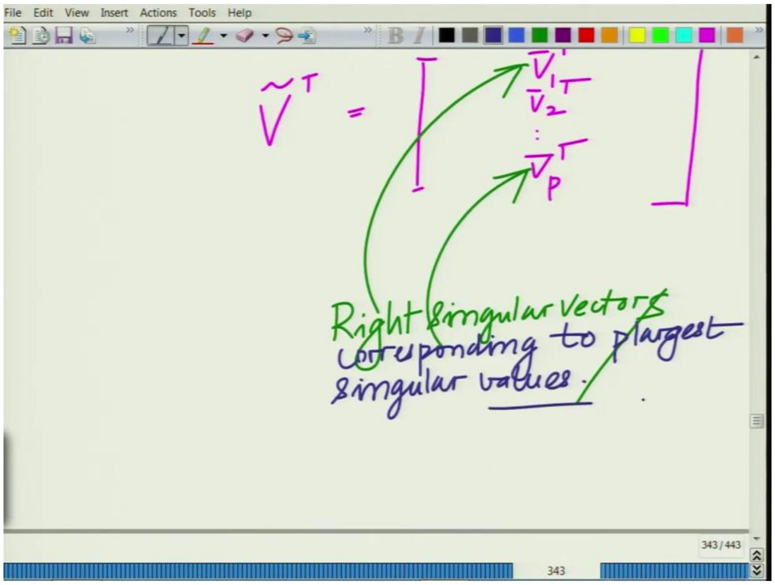
(Refer Slide Time: 31:59)


File Edit View Insert Actions Tools Help
$u_1 \; u_2 \cdots u_p \; u_{p+1}$
σ_p
σ_{p+1}
$\tilde{U}$
Top
$p \times n$
$\bar{v}_1^T$
$\bar{v}_2^T$
$\bar{v}_p^T$
v_{p+1}
$m \times p$
left singular vectors corresponding to p largest singular values.
$m \times p$
$\tilde{U} = [\bar{u}_1 \; \bar{u}_2 \cdots \bar{u}_p]$
341 / 443
341

And V tilde is the top remember V tilde is this is the top p cross n you can think of this as the top p cross n sub matrix.

(Refer Slide Time: 32:19)


File Edit View Insert Actions Tools Help
decreasing order
$\tilde{V}^T = \begin{bmatrix} \bar{v}_1^T \\ \bar{v}_2^T \\ \vdots \\ \bar{v}_p^T \end{bmatrix}$
Right singular vectors
343 / 443
343

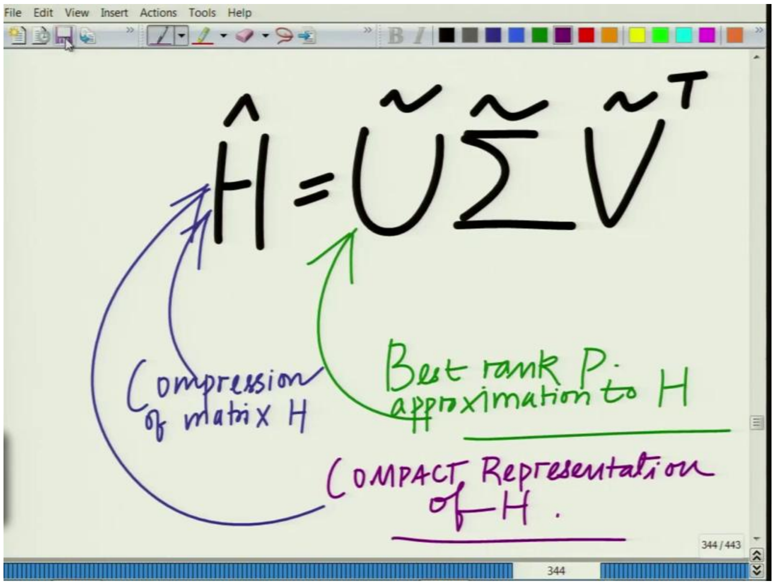


$$\tilde{V}^T = \begin{bmatrix} \bar{v}_1^T \\ \bar{v}_2^T \\ \vdots \\ \bar{v}_p^T \end{bmatrix}$$

Right singular vectors corresponding to p largest singular values.

So, V tilde or V tilde transpose you can think of this as the top p cross n sub matrix comprising of V1 bar transpose V2 bar transpose Vp bar transpose and this contains the right singular vectors corresponding to the p largest singular values.

(Refer Slide Time: 33:26)



$$\hat{H} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$$

Compression of matrix H

Best rank p approximation to H

COMPACT Representation of H.

And therefore, the best rank p approximation to H is now given by H hat now the best rank p approximation is given as H hat equals U tilde sigma tilde V tilde transpose. And this is the best rank p approximation to H. And essentially why do we need this best rank p approximation? This is nothing but a very compact representation of the matrix H because a matrix H can be of very large size remember 100 in 50.

But when you develop a rank 5 approximation it represents that essentially H can be compressed and can be represented the column space can be represented as a linear combination of essentially 5 columns, the row space can be represented as a linear combination of 5 rows and essentially the dominant axis corresponding to correspond to this singular value sigma 1 sigma 2 sigma the gains along these axis correspond to the singular value sigma 1 sigma 2 sigma of 5.

So, H hat is essentially if you think about this, it is a compact representation of H and I has several applications again it has basically very similar to PCA again can be used to represent essentially compact ways to essentially save transmit like store this matrix H transmit matrix H and manipulate this matrix H. So, in case you are wondering why we need this low-rank approximation, it is essentially a compression of matrix H or also you can think of it as a very compact representation of the matrix H.

So, that essentially gives us the interesting applications of SVD. So, we have seen two very interesting applications of a SVD, one is in the context of PCA principal component analysis that is instead of going to the eigenvalue decomposition of X transpose X we can directly look at the singular value decomposition of X and look at the right singular vectors, the right singular vectors corresponding to the rather signal values of X and that gives us the principal axis.

And the other is to develop all the best low-rank approximation right best compressed version of a given matrix H of very large size once again you perform the singular value decomposition take the left the sub matrix corresponding to be the left singular vectors corresponding to the p largest singular values, right singular vector right singular sub matrix of the right comprising of the right singular matrices singular vectors corresponding to the p largest singular values that is what we are calling as V tilde transpose.

And then we have the diagonal matrix sigma tilde containing the p largest singular value sigma 1 sigma 2 sigma p along its principal diagonal and then that is sigma tilde and then you have U tilde sigma tilde V tilde transpose that gives the best p rank approximation to H that is H hat which is the best p rank approximation.

So, as you can see the SVD is a very efficient technique a very important interesting and a very efficient technique and it has a lot of applications again we have seen in terms in MIMO wireless communications, machine learning, principle component analysis, compression, you think of signals representation signal compression and of course a cross fields just like the Eigen very similar to the Eigenvalue decomposition.

And probably in fact much more because Eigenvalue decomposition remember is define only for positive symmetric positive symmetric I mean for I mean it is defined only for square matrices singular value decomposition is defined for any matrix of arbitrary dimensions, so therefore can be used and has very very significant applications and therefore it is definitely one of the most important concepts once again in all of linear algebra and matrix algebra.

So, please go through this again try to understand it and try to appreciate the concept along with the examples and the applications. We will continue this discussion in subsequent modules. Thank you very much.