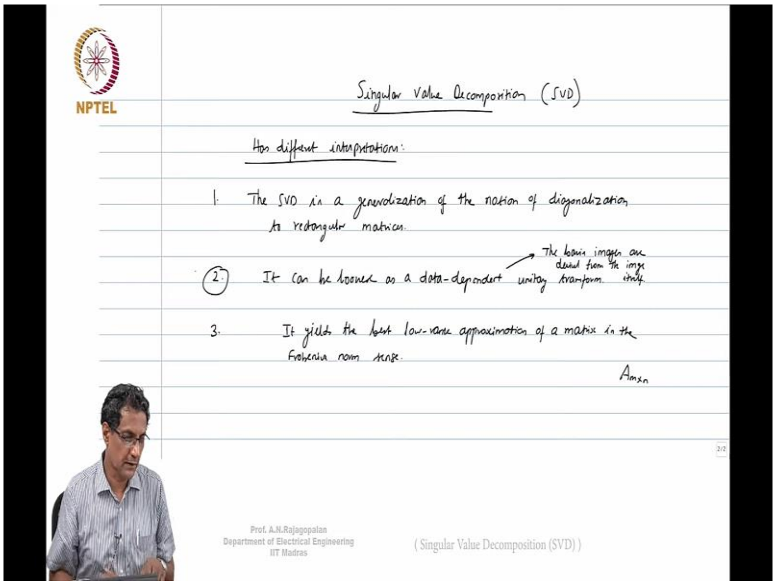
Image Signal Processing
Professor. A.N. Rajagopalan
Department of Electrical Engineering
Indian Insititute of Technology, Madras
Lecture No. 55
Singular Value Decomposition (SVD)

(Refer Slide Time: 00:16)



Now the SVD has various interpretations. Now I am going to write 4 of them here, 3 of them in fact and then it is up to and depending upon what you are dealing with you will interpret it accordingly. 1, the SVD is actually a generalization of the idea of is a generalization generalization of the idea of diagonalization of the notion of diagonalization notion of diagonalization to rectangular matrixes.

Now, until now we are only talking about diagonalizing covariant and all that, square matrixes basically. The notion of diagonalizing to rectangular matrixes. See even for circle and all we always assumed everything was square and then we talk about diagonalization does not have to be covariant, even your system matrix we assumed it to be circle and then we did. Now we want extend that notion of diagonalization to non-square matrix, rectangular or they are called non-square matrixes.

But the same will be useful if we use a square matrix but the notion is an extension. 2, it can be looked upon as a data dependent transform looked as data dependent unitary transform, looked as a data data dependent unitary transform. This is also another way to look at it is kind of a unitary transform, data dependent unitary transform which which basically means that you know you should be you should be able to expand a given image in terms of an orthonormal this one basis but the only thing is the basis images themselves would then come from the image itself just like a PCA.

But then in a PCA it was an ensemble thing you had many images from which you computed your Eigen vectors. Here it has to be something like that image a basis corresponding for, a basis that works only for that image. To change that image, you cannot use the same basis. So, this was not a case with respective PCA, I mean you could just have this kind of just justhave this sort eigen vectors. And then if I give you another phase you could you could use it; you cannot do that here.

But then the notion of unitariness, everything is fine. Because we are also talking about unitary transform so I thought we should definitely include this point. 3, another way to look at it is it yields the best it yields the best it yields the best low rank approximation low rank approximation of, of course a matrix in the frobenius norm sense. So, what it actually means is that if I actually give you so what this means is if I give you a matrix which is let us say some A m cross n.

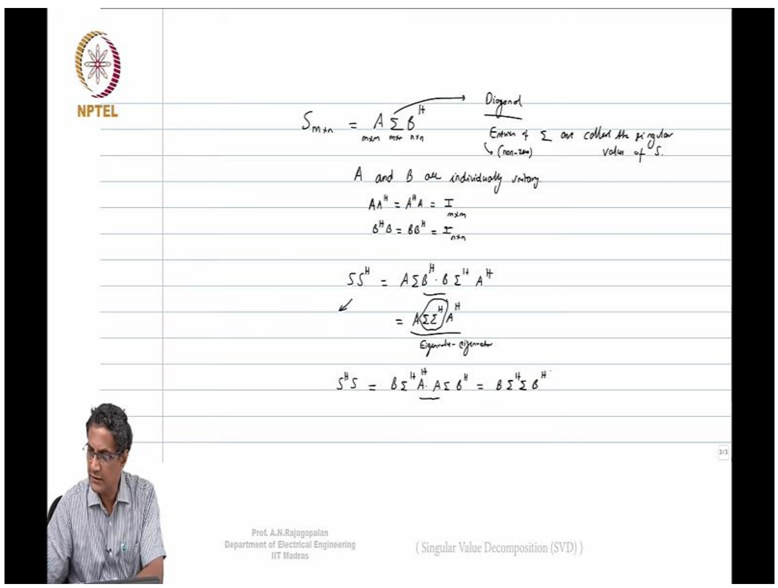
Let us say the rank is m smaller than n and the rank is m. then suppose I ask you for an A HAT having a rank less than m, let us say m minus 1 or some other rank less than m. But suppose I ask you which is which is the best approximation that I can get for this A m n with some other A hat, A hat mn of course the rank that is less than this guy's rank. Which is the one which is closest to be in this frobenius norm.

So, that means if we take the norm of A minus A hat then the frobenius error right in that sense then which one will be will that matrix be. So basically, then you would not have to run around to find out what will be that matrix the SVD will actually will directly give you that matrix. So, it can give you the best low rank approximation in the frobenius sense. Now, this one so this one I think one separate point but need not be a separate point. So, here the data dependent transform in the sense that the basis form are derived from the image itself which actually raises some interesting points which we will talk about.

And then so these are the 3 points, I am not really talking about applications and all. Applications you can, I mean SVD is all over the place you know pseudoinverse you name it right. You used it already for inverse homography, it can be used to compute the PCA so many things applications wise. So, these are really the 3 main things for us right this matter the most because what, how we started all of this

We had data independent transforms of different kind and data dependent PCA we did already, the KLT, now the second one is you know SVD. And then the other points are all related to diagonalization which we have already looked in the context of transform. The third point is included because that is actually a nice thing to know in terms of the low rank approximation. Now now what it actually says is that now according to SVD.

(Refer Slide Time: 5:44)



If you have S mn a rectangular matrix S mn then it can be expanded under you know this again is a standard form. They do not write it as A Hermitian sigma B they write it as A sigma B Hermitian. They are simply following simple text which is why if you look at the Eigen vectors earlier you say there used to be columns of A Hermitian now they will become columns of A because of the way the singular value decomposition is, it is the standard way which they write. So, I just want to go with what they write in text.

Yes, that will only mean the style way in which we interpret our Eigen vectors. Alright now the point is this and also if you see for example another thing is that when we talk of pseudo inverse and all and that. When we have a rectangular matrix, you cannot even talk about its inverses t. You know generally we do not talk about but then there is something called as the pseudo inverse which allows us to even talk about the inverses of rectangular matrixes.

And so so, that inverse is not really the real kind of inverse right that is why it is called as the pseudo inverse, you must have read it somewhere pseudo inverse you guys must have read. So those properties are not they are not exactly the same as what you would you know it is not like the real inverse. But then there are some 3 or 4 properties that satisfies which look very clean.

So, and SVD is also used actually to get the inverses of rectangular matrixes, what is called as a pseudo inverse. And it is also it is also it is also used you know when you want to compute the health of the matrix which can be which can be which can be found in terms of what is called as a conditioned number. So, when you want to compute a conditioned number it is computed in terms of the singular values of that matrix.

Again, you can you can use SVD to compute the health of the matrix and so on. That is something that will come up later, the condition number and all. When we do, when we do like you know two topics away from now but at that time that we will talk about it we will actually use SVD at that time. But right now, this menas is that m cross m, this is m cross n. And A and B are you know individually unitary.

It does not mean that you know A B Hermitian, no his identity or something. A Hermitian is identity, DB Hermitian is identity are individually unitary. Under some special circumstances A can be equal to B. but generally it means that A Hermitian is equal to A Hermitian A is equal to identity but this identity will be m cross m. And B Hermitian B is equal to BB Hermitian is equal to I, this will be m cross m. So, these two I's are not the same.

Now this guy this called kind of diagonal even though it is not really square but we call this a diagonal matrix because of because of the way it is structured. See for example and the entries of the sigma, the entries of sigma that is whichever which are kind of non-zero. These non-zero entries of sigma are called the singular values, are called the singular values of S, which is why it is called as Singular Value Decomposition. And by the way SVD also has an interpretation of encoder decoder kind of interpretation. There is so many things to it.

So, we just look at something that is that is just useful for for this course. Now what this means is that if I try S Hermitian S, all the way I am still sticking with Hermitian thing. So, if we do okay or else let us do SS Hermitian, maybe let us do that first. SS Hermitian will mean that we have A sigma, so so if you are really wondering what is A and B are?

Now another way to look at what is A and B mean is if you try SS Hermitian then there will be A sigma B Hermitian and then S Hermitian is B sigma Hermitian A Hermitian. And because we said B Hermitian B is identity it turns out to be A sigma sigma Hermitian A Hermitian. And since this matrix is of the form SS Hermitian it is automatically symmetric right. It is actually Hermitian symmetric and therefore therefore, we cannot look at this as an Eigen value in vector kind of a decomposition of say SS Hermitian Eigen value Eigen vector composition which we saw only only only see recently.

So, which then means that now the now the now the only only change is that is that the Eigen vectors are all in the columns of A now because of the way this SVD you write it. So, the Eigen vectors of SS Hermitian are but the columns of A and then the Eigen values are, Eigen values of SS Hermitian are sitting here. This matrix sigma sigma Hermitian which will now be diagonal okay which will be sqaure and diagonal.

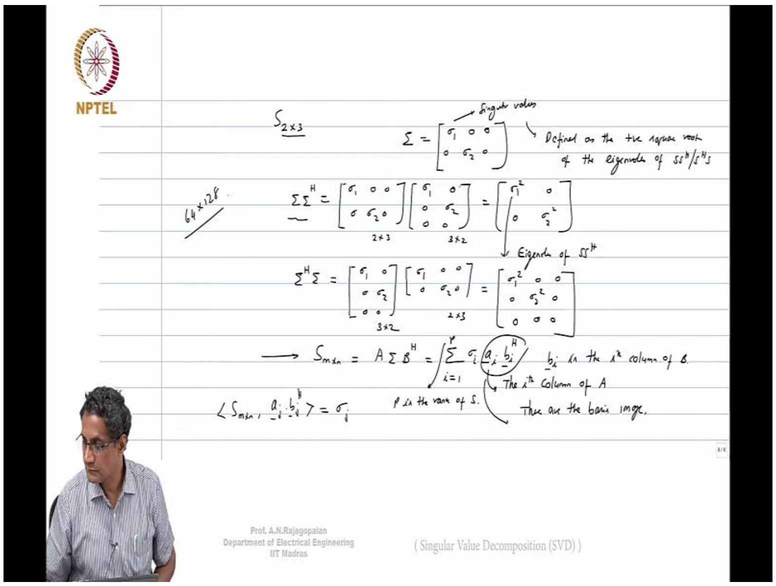
Then you can also look at what happens to S Hermitian S and then depending upon the rank, this structure will change whether you get all zeroes as the last row or whether you do not get them depends upon which is which is depends upon the ranks of S. S Hermitian is if you do you get B sigma Hermitian A. Wait a minute, S Hermitian A Hermitian, then A sigma B Hermitian that gives you B sigma Hermitian this is identity, sigma B Hermitian. And therefore, the Eigen vectors of S Hermitian S will now be the columns of B.

Now what this actually means is that, so this A and B really if we think about them A and B are really you know are really Eigen vectors. They contain Eigen vectors of in one case SS Hermitian corresponding to A and B contains the Eigen vectors of S Hermitian S. So therefore, this A and B are not something are not dropping from somewhere.

So, which is also the reason why this kind of dependent transform right when we say the basis images are all coming from coming from coming from in fact A and B because here because the basis images get formed from the columns of A and B because A and B come from S itself. In one case SS Hermitian and another case S Hermitian S. Therefore, we say that the basis is dependent on the image itself.

So, if you change the image for example if I go from S to some other image even if it has the same dimension it does not mean that A and B will have to be the same. A and B will then automatically change if I change my S which will mean that the basis images will change.

(Refer Slide Time: 12:31)



So, if I just want to let me take a small example just to illustrate how it might look like. So, if you take S let us say for example 2 cross 3, what it will mean is that your sigma will then be 2 cross 3. What this means it will look like sigma 1 0 0 0 sigma 2 0. So, these are called the singular values which then means that, then of course A and B have 2 cross 2 and 3 cross 3 as their dimension.

Then if you look at sigma, sigma Hermitian and that will look like sigma 1 0. And and one more thing these singular values they are actually defined as defined as the the positive square root of positive square root of of the Eigen values of SS Hermitian or equivalently S Hermitian S. So, if you look at it so, sigma 1 0 0, sigma sigma 2 0 Hermitian will mean that sigma 1 0 0 sigma 2 0 0. So, you got like 2 cross 3 and then you got like 3 cross 2 and this will give you sigma 1 square 0 0 sigma 2 square right.

So, so as you can see clearly these are the Eigen values of of of S what is that which was this guy SS Hermitian. So, this is so these are the Eigen values of SS Hermitian, SS Hermitian and the positive square root of these Eigen values is sitting here as the singular values which is sigma 1 and sigma 2. If you do the other way round sigma Hermitian sigma, then you will get so that is sigma 1 0 0 sigma 2 0 0 and then sigma 1 0 0 sigma 2 0 0. So, this is 3 cross 2, this is 2 cross 3. Therefore, we will get sigma 1 square 0 0 0 and then sigma 2 square which will all be zeroes.

Now I am going to leave it as an exercise for you to show that the S m n that we wrote that I wrote as A sigma B Hermitian. I will leave it to you as an exercise. Please show that this can be equivalently written as suppose let us say S has a rank R R has a non-zero singular values, then it can be written as R what should I use here I let us do P whatever it does not matter. I could do 1 to P where you know P is the rank of S. If S is full rank it will be smaller of n m or n or otherwise it can be less than less than that number.

So, I could do 1 to P is sigma i ai bi Hermitian, so ai is actually a vector now. This I am going to leave it to you ai is the ith column of A and bi is the ith so bi so, I am not going to write bi Hermitian so bi is the column of b which is why I said that there is a small sort of a difference between you know the way we did prior like when we did KLT it was like column of A Hermitian, Si Hermitian. But because of the way SVD in standard form of the SVD is A sigma B Hermitian and therefore we turn out and this will come out as columns of A and columns of B.

Now if you look at this ai bi Hermitian these are like the column of vector multiplying a row right, ai vector is coming from A, bi is coming from B and together when you try to multiply this so this kind of this sort of an outer product you have here. This is the basis image so these are these are the basis images. And again, I am going to leave it to you as an exercise to show that if I take inner product of S m n with let us a j b j Hermitian this will give me sigma j, this again follows the same thing.

So, what I can now look upon is your so if you look upon it as an image expansion problem and if you look at S as an image. Now in this case it is a square image not a square it is actually a rectangular image a n cross n, that image can then be expanded in terms of the basis images that you obtain by doing ai bi Hermitian and these ai bi Hermitian and these ai bi Hermitian.

So, what kind of a weight should you attach to ai because we need a weighted linear combination? What kind of a weight should then be there then that weight is actually the singular value itself the singular value itself that you can again obtain by either doing S m n. We can show that enough product of S m n with a j b j Hermitian will give you this. Now here is where here is where it is interesting because the basis image is coming from A and B that is the vectors of A and B. And it also means that you know what there is one more interesting thing is suppose I say S is an image. Suppose I say that S is some let us say 64 by 128.

And if I wanted to transmit S as it is, let us now look at the image expansion problem. Suppose I transmit the image itself, then how much how many bytes will I need? It is an intensity so let us assume we got like 256 grey levels, so 1 byte for each grey level. So, 64 into 128 that many bytes. Now If you are transmitting in this manner, what can you say about now 1 byte for an integer no that is if you are transmitting a grey level.

How many bytes will you need, see you would not want you will not want to send ai into bi Hermitian the whole matrix that does not make sense because then I would end up I mean why would I want to use this to send because even in one image I would have consumed my whatever the whole size. I would rather send this vector and that vector and then this outer product somebody can do at the other end right to get this to get that whatever m cross n.

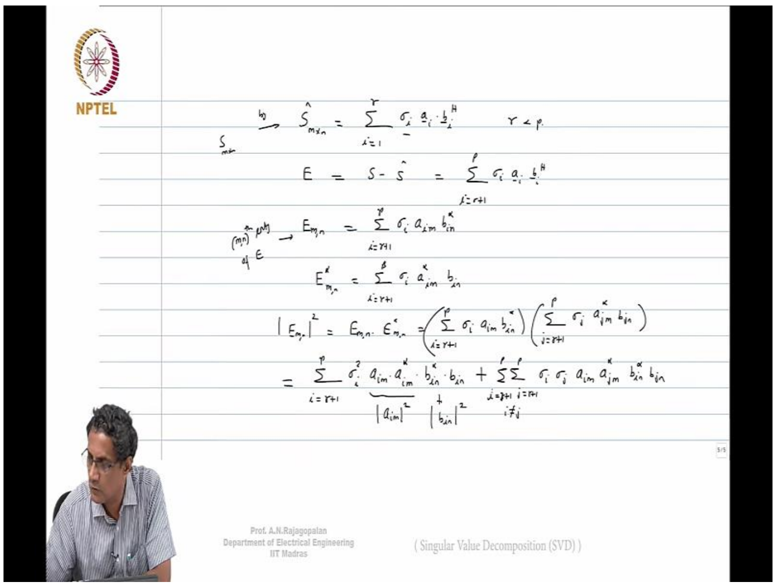
So, how many entries are there in ai? ai will have 64 cross 1 that many entries in it, bi will have bi will have 128 cross 1 or Hermitian is 1 cross 128. So, how many bytes do you think we need to send a and bi? ai is actually a unitary, ai Hermitian, A matrix is unitary therefore these numbers need not be integers. It can even be a complex number for all we know.

So, normally you would assign about what do you do float bytes of something we should take it as a float so it will be like 4 bytes minimum. 4 bytes times 64 and then4 bytes times 128 bi. If you want to send one sort of a basis image and then this is a singular value which is again a float that is like plus another 4 bytes. Now you have to see how many such basis images can you add, again we would go by the significance of the singular value, something which is less significant you may not want to transmit.

You may want to pick the most significant singular values and quickly add them up. So, you should be able to show that that beyond a point let us say I equals to 1 to R may come an R which is less than P by which time I would have exceeded the size of S itself it can happen. So, in that sense in that sense nobody uses SVD to you know do this kind of a compression at all.

But I am just saying that you know theoretically the possibility exists I mean you can ask how many turns should I get involved in this expansion before I exceed the size of S if I had to transmit S that way. You could ask that question but normally people do not do not really use it for that but then because we are doing this image expansion business, I thought I mean it is important to know at least that such an expansion is possible, whether you do it or not is another thing.

(Refer Slide Time: 21:34)



Now what you might want to ask is the following. Suppose I do I so say S Hat m cross n such that I use only let us say i is equal to 1 to r terms in this what is that sigma i , ai bi bi say Hermitian, r less than p. See if I use all of p and of course I reconstruct S exactly correct but suppose I do not use and now let us assume that it is all ordered, the singular values are all ordered.

But I just take the take the you know top r terms so r singular values. So, those are the ones that I use so I want to ask what kind of an error will I now make in terms of approximating S by S mn by this S hat. So, suppose I call that error matrix as E, let me call this E m cross n is S minus S hat. Both are of course m cross n, now this will be summation i is equal to r plus 1 is to p sigma I ai bi Hermitian. So, all the terms that we have left out, that will be the that will be the matrix the error matrix.

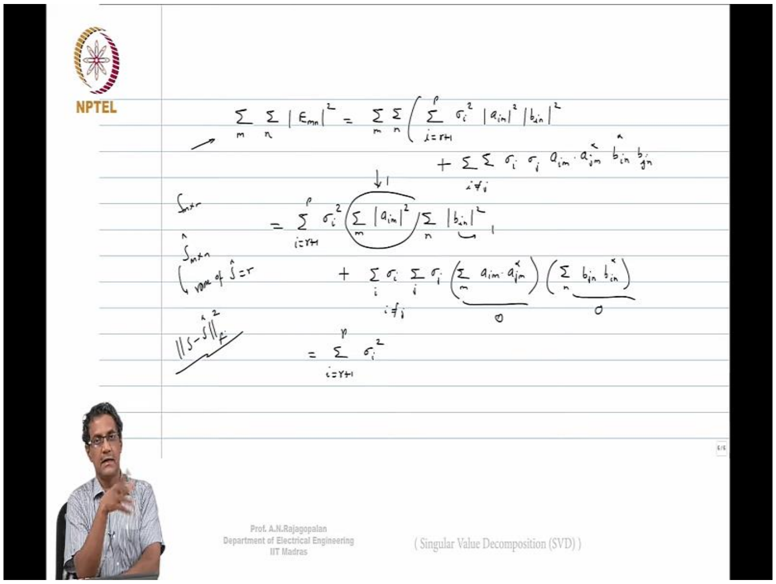
Now if I want to now yeah if I now let me write this as E, let me not write this mn. Let us we know that these dimensions but what I want to represent this, suppose I write m comma n, I mean the mnth entry of E entry of E suppose I examine the mnth entry in this matrix then that can be shown to be I just leave it to you or you already know what it is r plus 1 to p sigma i ai m bi n star, down to the scalar expression now. The star is of course a complex conjugate. Now, E start mn will be what summation i is equal to r plus 1 to p so sigma is anyway real therefore know that just stays like that.

Now, you will get the whole star so this will be the whole star. This will be a star im, b in. Now, if you look at magnitude E mn square that means at the entry the mnth entry in E what is the square error what is the magnitude square error, that we know is E mn into E star mn or in this case that would be summation i is equal to r plus 1 to p sigma i ai m bi n star into let us say summation let us use some j is equal to r plus 1 to p sigma j a jm b jn no a star b jn because E star mn.

So, then this will be what so if you just collect the all turns where i equals to j then this will be like r is equal to 1 plus to p sigma square I a im and a, I is equal to j so a im complex conjugate into b in star and b in plus a double sum i is equal to p what is that r plus 1 to p is equal to r plus 1 to p but i not equal to j sigma i sigma j, a im a jm star b in star b jn, if you just expand this. Then now for example, if you see this quantity this but magnitude a im square, this is magnitude b in square.

Now, the now if you want to compute the frobenius norm of this matrix that will be the that will be the sum total error. When I do S an approximate S by S hat. What is the sum total error that we are making across all the all the entries in the matrix that will be like for that what should I do? I should sum it up over all m comma n over all the all the entries in this error matrix.

(Refer Slide Time: 26.20)



So, suppose if I do that then summation m n over this size of the image magnitude E mn square this will be the this will be the this will be the frobenius norm of frobenius norm square of E in the error matrix. This now will be equal to summation m n and then whatever you had there, right you had summation what was that i is equal to r plus 1 to p, then you had sigma i square then what was it some magnitude a im square then magnitude b in square plus summation double sum i not equal to j, sigma i sigma j.

Then whatever a im then a jm star, what was that a jm star then b in whatever b in and then b jn star b in then b jn. No b in star and b jn b in star and b jn okay this is what you have. So now if you actually push this summation inside then if you just pull out. So, i is equal to r plus 1 to p sigma i square push this m inside. You have magnitude a im square into summation, push this n in sqaure b in square plus then summation let us keep this i.

Let us keep this i outside and then sigma i then j sigma j and then summation m a im a jm star into summation n b jn into b in star. Now this we know, this we know is simply the simply the norm of the the column of A because A hermitian is identity therefore this one. Because A and B again form this orthonormal set pf vectors orthonormal basis or orthonormal sets, this is also 1.

Now if you look at it, here i is not equal to j and therefore this is like saying that this summation will give you 0. Because i is not equal to j, so this is like taking the inner product of a ith column with a let us say a jth column which we know they are orthonormal. And similarly, here also i is not equal to j therefore right this is also 0. Therefore, this simply boils around to summation i is equal to r plus 1 to p sigma i square.

So, so which we know is basically the Eigen value of right sigma i square and all the Eigen values of you know SS Hermitian or S Hermitian S. So, in the PC also we had written something similar, we had said summation lambda i of all the Eigen values that we left out. See that you get very very similar expressions of the SVD except that now this is exact.

This is for example if you take a matrix and you approximate it by some right by whatever if you just collect the first whatever we are saying r terms and then if you just do this expansion and then you just compute it there and this is exactly what it will be. In terms of the PC it is more like more like a mean square property, this is like for that image what exactly will happen okay if you make this approximation.

Now incidentally, there is also a theorem which we would not prove, what it says is that among all matrixes that now among all the matrixes that have suppose you approximate S. In this case S m cross n with an S hat m cross n then where the rank of this guy is let us say rank of S hat is equal to r suppose. So, we say that I want the rank of S hat to be r but then it should to be as close as possible to the frobenius sense. So, that means norms minus S hat you know square is as small as possible in the frobenius sense then this is the guy.

So, if you wanted to compute that matrix you would just do exactly this SVD, we will take the first r terms of course order their singular values. Take the first r terms whatever is that matrix that you get, it will have an of course its norm error will be exactly its summation sigma E square and wherever wherever you go in the world to pick a matrix of this dimension m cross n and of this rank r.

This is the matrix that will give you the smallest you know will give you the best low rank, it is called low rank because you are going like one rank you are going lower rank than S and trying to get an approximation to S. So, the best of which is what my third point I had written this is the best low rank approximation for S because SVD can be actually used for that. This I this I derived just just to show that would be the what error that will you be making.