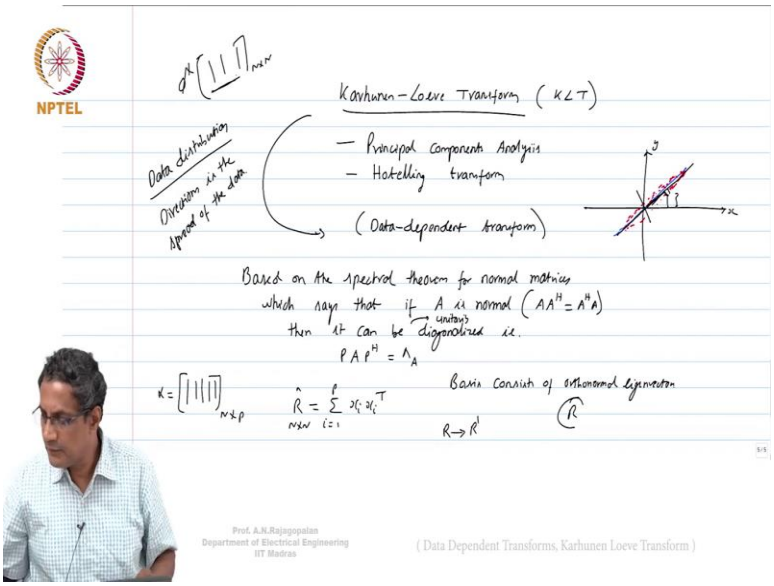
Image Signal Processing
Professor A. N. Rajagopalan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Lecture - 51
Data Dependent Transforms, Karhunen-Loeve Transform

(Refer Slide Time: 00:19)



So now what I want to do is, I want to move on to a KL Transform now, what is called a Karhunen-Loeve Transform, because until now whatever we have done is all kind of data-independent. We did DFT, we did DCT, we did WHT we can go on but each one of them now you know, you know the drill.

So Karhunen-Loeve is, Loeve transform is the one that is actually a data-dependent transform, SVD is another, so I think these are the two things that we want to do, at least I want to start today this Karhunen-Loeve transform, it is also called KLT.

This is not a Kanade-Lucas tracker, another one that is called, referred to as KLT is really a Kanade-Lucas tracker, this is KL Transform. It goes by different names, one is a Principal Components Analysis, principal components analysis and it also goes by the name Hoteling transform, I think it is a double l, Hotelling transform. It goes under all these names.

Now this one, as opposed to the earlier ones that we had till now, this is a data-dependent transform. So which basically means that we do not know a priori what is the basis and so on. Sometimes, of course, when you have an equivalence like there is a data-independent transform which becomes a KLT then maybe a priori you know the basis, but unless you look at the data you will not even know what kind of a basis you have.

And therefore, and all the separability and all, it need not automatically happen, it depends upon whether your data structure has that or not. Therefore, there are no fast implementations for this, generally, there are no fast implementations and also the fact that because you need to know the basis, it is not like in data-independent case.

Therefore it people seldom use it for real coding but there are some very nice applications of this which I will show, but as far as coding is concerned people do not really use it but as I said earlier a data-dependent transform is more like a benchmark, something that can be done in an optimal way if you knew something about the data and then you can look at a data-independent transform and try to see how close it comes or how far away is it from or whether something can be called KL transform, even one wants to do that. Like we said a DFT and sometimes for certain matrices becomes the KL transform and so on, so this KL transform really hinges on this.

This is something that we have already seen before and we have said before. This is that, this is the, now this is based on the spectral theorem for normal matrices. So this comes from based on the spectral theorem for normal matrices, I have kind of mentioned it earlier, for normal matrices according to which, which says, it says that if A is normal which, means in our, normal means A A Hermitian A, that is what we said it is like a normal matrix, then it can be diagonalized.

We are not going to go into the proof of this just take the statement on its, as it is, can be diagonalized or it can be unitarily diagonalized, that is you can find out a unitary transform, let us say some P then P A P Hermitian which will diagonalize A. And then the nice thing about this, why is this a powerful thing because of the fact that it gives you a basis consisting of orthonormal eigenvectors. It consists of orthonormal eigenvectors that is where this its strength lies.

So now, what this actually means for us? So this is orthonormal eigenvectors, eigenvalues, and all that, I mean why they are so very important when it comes to a statistical property? So for that let us just take a take a small sort of a diagram. Suppose, let us say, suppose I have, let me just draw an imaginary line here, and suppose somebody gives me some sort of data points such that they are here.

So for example, some x, I get some y so that they are around this, somewhat like that. Some sort of an ellipsoidal, it is like spread around a line. Of course, you can have some noise and all that but then it is kind of spread along around the line.

Now when you see a data like this and suppose you go to your standard basis, which is your x and then this is your y, one thing which you notice is that as x increases y seems to increase, generally, like y is increasing x is increasing, as x decreases y is also decreasing. So you see a correlation, it is a strong correlation in the sense that when x is increasing y is automatically increasing, y is also increasing, and x is decreasing y is also decreasing.

So what this actually means is that, so one way to see is that there is a strong correlation. The other way to see it is that if you look at the spread of this (variance), I mean if you look at the way or the spread of this data or whatever you call the variance of this data, so the variance seems to be like it is very well aligned along this axis. It looks like the maximum spread in this data is occurring along that axis and there is, of course, another axis which is orthogonal to this along which again, some kind of spread is going on. Correct? You see that there is a maximum variance in this direction and there is some sort of a variance in this direction.

So the point is if you try to see, so it looks like if I want to instead of sending both x and y, see, for example, I mean here, if I have to send a, send an information about a point. suppose, I send only its x coordinate then what it means is, in this standard basis it means that so much, so for this point, the y is so much and the entire thing I will be ignoring, whereas if I look at the, look at this axis where there is maximum variance, if I see this point, and suppose I find out what is its component, so it has a component along this axis

which is that much and then along the orthogonal axis the component is much, much more smaller.

So it looks like if I were to capture the spread, I mean if I had some notion about how this data is spread out and if I can find the, so what I need is an orientation of these axes, of course, again need this orthonormal axes, x and y are again orthogonal, but then the thing is in that standard basis if I try to express this data then I am not able to express it in a sort of a neat way.

What would be nicer is some way if I could get a sense for how this data distribution is, which means that I need several points, this I cannot do with one example and all, if there is a distribution of data available and based upon that if I can make some assessment about what the variances are, in the sense that in which directions are the, is the spread of the data, directions as the spread of the data, if I can get a sense of these then what I can do is I can actually, I can then ask can I sort of change the basis now.

Instead of using my standard basis like x and y, instead of that can I change my basis such that along the directions of maximum variance I will try to use them as a basis and those are the eigenvectors. The eigenvectors are the ones along which you have this, you have the maximum spread of the data.

So all of this boils around to, which is why, now which is why all through whenever we did a diagonalization we were always looking at eigenvectors, eigenvalues, eigenvectors eigenvalues, and so on. Now what happens is because this is a covariance, so what happens? Suppose, let us say, suppose, for example, somebody gives me faces let us say, human faces. Now imagine that I have a human face and suppose I stack it up as one sort of a vector.

Now if you give me several such examples, suppose, let us say now these days it is so easily available. So let us say each face is n-dimensional so I just stack it up as one vector and suppose, I have N cross P, where let us say P is much higher than N. P is like whatever ten thousand, twenty thousand, one million maybe, that many examples I have.

Now if you call this as your matrix X, wherein all my examples are there so all this I need, I mean without which I cannot, so the KLT that is why it requires a lot of data and all because you need to understand what is the kind of data that you are actually looking at. You need to have an idea about what is the, what kind of correlation exists, and so on.

And therefore, it if you try estimate R hat, because generally, nobody gives you an analytical R, typically for the world examples we do not have an analytical R unless, of course, you model it such like for example, that first order. You could impose a model but then if you have enough examples you could even try learning the covariance.

So what you do? You do something like expectation or just assume that I make it all zero mean, then what I can do is I can simply do a summation xi xi transpose with i running from 1 to P. Now what this actually means is that, so what this will mean is that I will have R hat which is actually N cross N.

Now in this N cross N matrix if I try to see what or what kind of a correlation exists, and suppose I can transform it to a kind of a different basis wherein, so it is like saying that suppose, I go from R, suppose I have an R and suppose I do an R dash, suppose I go from R to R dash by doing some kind of a transformation, wherein R dash I suddenly see that this kind of a de-correlation is going, de-correlation happens.

Like here with respect to x y I see a lot of correlation, but suddenly when I transform my axis, I suddenly find that this kind of a correlation almost, I mean if it can go down to 0, it is very good but then at least, if there is a lot of de-correlation going on I would like that instead of going with some correlated data.

That is the idea behind actually, behind this PCA, it is called principal components because which are all, principal means, which are all the significant eigenvectors, principal components to be equivalently what it means is which are the significant eigenvectors and you basically decide significance by looking at the eigenvalues. So those eigenvectors that have a large eigenvalue will be the ones which will be significant, so that is the idea.

So now, so all this means that you need to access to data, you need to be able to learn all this unless you, of course, know R a priori analytically, most likely you do not have an analytical R, therefore you will have to estimate R which means you should have a lot of examples. But once you have all of that, then you can actually start to kind of think about, think about what kind of approximations can I make now.

Now if you go back to your DFT, DCT and all, and suppose I had a sequence, let us say U, I gave you a sequence U and suppose I said instead of using all the eigenvectors of in that Fourier basis, suppose you had a Fourier, there you had, you had some phi star which was, whose columns are the basis.

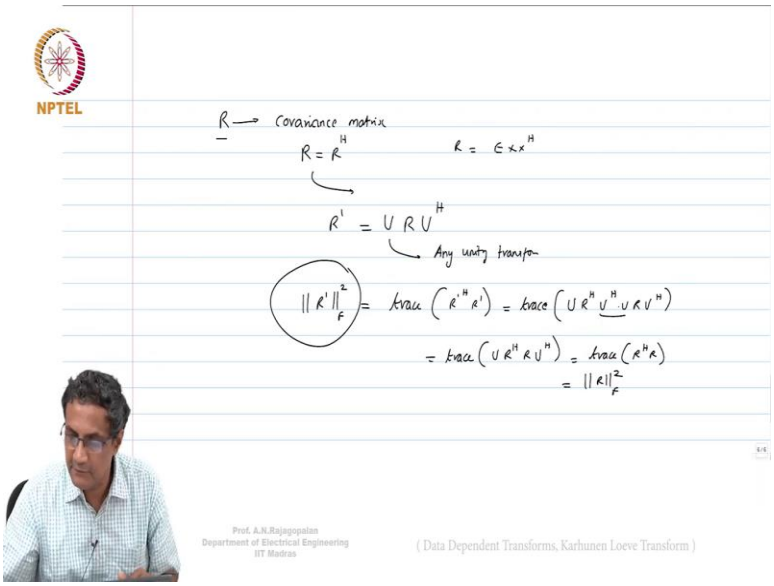
Now, if you choose to use only a few of them, see typically, this is like, it is like N cross N, now if you choose only P of these columns, not all N and then suppose you ask what will be the error that I will incur? Then in general, there is no answer. There is no answer in general that I can give because depends on what that U is. For each U you can get some error and there are no kind of general statements that I can make that you will incur an error within this range and all. I do not know because such statements I cannot even make because it is all data-independent.

Now with respect to KL transform, the nice thing is that if you can identify what those eigenvectors are, which you can if you have this covariance matrix, then what you can do is if I now say that from these Eigen, so think of this as an Eigen basis, so they are again orthonormal vectors. Now if you try to just pick a few of them and suppose I ask what will happen if I try to reconstruct with only some of them?

Then you can have a notion about how well you will do, that means in general you can claim that you can try, you can claim that if some example comes from that class, of course, provided it comes from that class, for example, for face; if I have computed the covariance matrix and suppose I just use some of the eigenvectors which are significant but leave some out and suppose I ask how much of an error am I likely to make on the average, then what you can say is suppose I give you a new face image I can a priori tell how well I will do. This you cannot do with your data-independent kind of transform.

So this is the power that this KL transform brings in, so in a sense, you get know how well you will do, in general, provided the data samples come from that class. If it is a different class then, of course, you will have a different covariance, you will have a different set of eigenvectors, again the same argument holds. Now in order to do this what we, okay let me just go one step forward.

(Refer Slide Time: 14:14)



And so, the point is when you look at this one, a distribution, talk about correlation, de-correlation and all, so it all has to do with a covariance matrix. R is a covariance matrix which is what kind of captures what kind of a correlation you have across these elements of x.

Now this covariance matrix R, we know that even if you take a complex case then we know that R is equal to R, R this is Hermitian. I mean, typically it is, of course, a symmetric matrix but in general, you can think of R as coming from some expectation x x Hermitian if you have complex data, it can even come like that. Typically, we will write it as x x transpose but I do not want to bring in real numbers as yet, the images are all real so they definitely would not get complex entries but let us just keep it general.

Now the point is which then actually means that we know that if a matrix is symmetric, then, of course, its eigenvalues are real, on top of this, if it is a positive semi-definite

matrix which is R, so then, it also means that all your eigenvalues are basically greater than or equal to 0, these things you know.

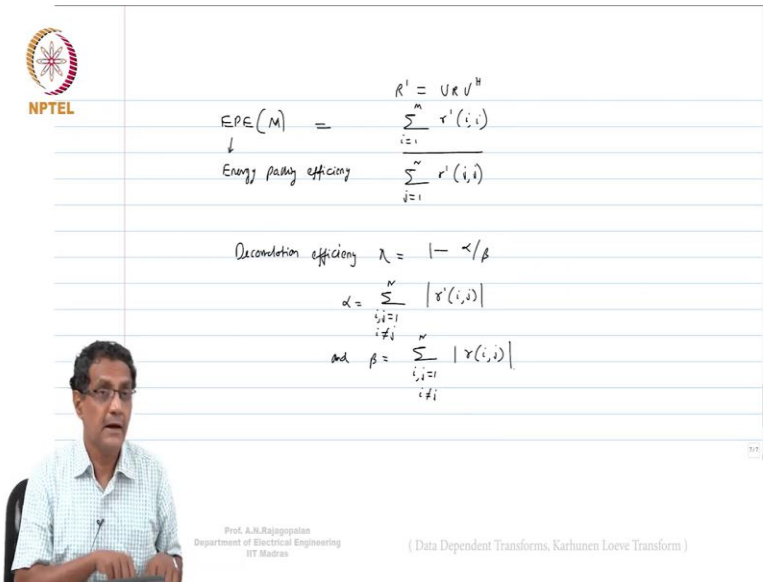
Now, what you can do next is, so suppose we do a transformation, suppose I take my R, suppose I multiply with some U and some U Hermitian and then in order to get an R dash. Now, this is, let us say any sort of a unitary transform. I am not saying that this transform will actually diagonalize R, I am just saying suppose I applied a unitary transform that is U U Hermitian as identity is equal to U Hermitian U, and suppose I acted it on R, then if you look at the Frobenius norm of R dash, suppose I look at the square of the Frobenius norm that will be trace of R dash Hermitian R dash.

So which will be a trace of, R dash Hermitian is what U R Hermitian U Hermitian and then R dash is U R U Hermitian. U Hermitian U is identity, therefore this is just trace of U R Hermitian R U Hermitian. And because of the fact that U is a unitary transform, this will, this can be shown that this is equal to R Hermitian R trace of, which is nothing but norm of Frobenius norm of, Frobenius norm square of R. So the point is this.

So when you take a U and suppose we act it on a this one, covariance matrix, then irrespective of whatever happens, whether you de-correlate or whether you do not de-correlate we do not know or how much de-correlation you do, then the fact is whatever is the norm in R dash, the same sort of an energy is also, what was in R is still intact in R dash, except that there is a kind of a redistribution.

Now this redistribution, you want it in a manner that ideally you want your off-diagonal elements to go to 0. Suppose R has off-diagonal elements, ideally if you want to de-correlate it means that your off-diagonal element should all go to 0. And then the other thing that you would like is you would like to pack as much energy as you can in the first few coefficients along the diagonal. You have the variances right along the diagonal and you would like to pack as much as you can in the first few coefficients. Now, these are two properties that are kind of, that are defined in the following way.

(Refer Slide Time: 17:33)



What is called energy packing efficiency is EPE, this stands for Energy Packing Efficiency. Now, from now on this is all in terms of a covariance. So this is all like an ensemble property now. EPE of M that means if you take the first M coefficients, and suppose you have done R dash is equal to U R U Hermitian, suppose you have done this, then EPE of M is given as i is equal to 1 to m, where R dash i comma i divided by j equal to 1 to n, all the way R dash j comma j.

So, after transformation what can I do? You can also do it for R, R i comma i, and then R. So how many were initially packed in R when the first M coefficient, now after you have done the transform, how is it changed now?

The other thing is de-correlation efficiency or a de-correlation factor or de-correlation, I think this is typically called de-correlation efficiency, which is given as eta, and this is given as 1 minus alpha by beta, where alpha is summation i comma j equal to 1 to n mod R dash i, j i not equal to j and beta is equal to summation i, j equals 1 to n i not equal to j but this is taken over R. So we will follow this up in the next class.