

Pattern Recognition and Application
Prof. P.K. Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 16
Dimensionality Problem

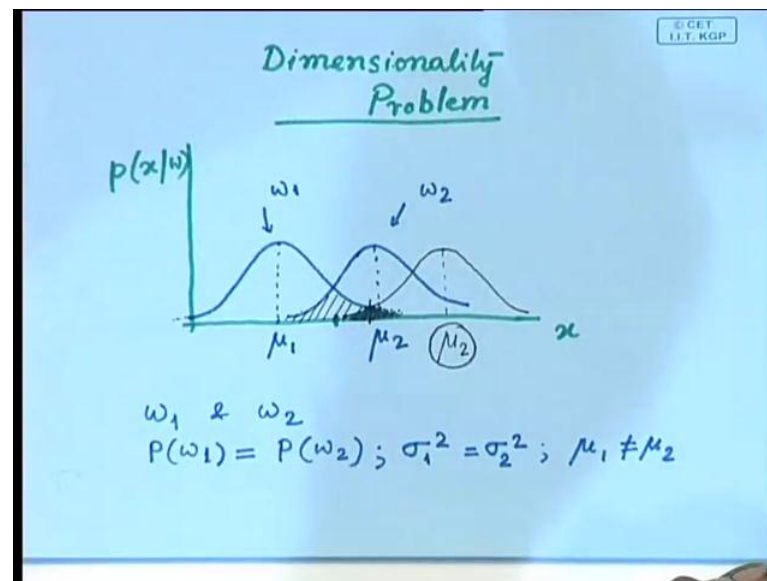
Good morning. So, in the last class we have talked about that given the probability density function which has a known parametric form, where it is not necessary that the probability density function has to be Gaussian or normal. But if it is having any other form which can be parameterized, and that is we have the known parametric form of the probability density function. Then it is possible to estimate the value of the parameters from the training samples and the kind of estimation that we have talked about is maximum likelihood estimation of the parameter vector. That is the value of the parameter which is best supported by the training samples those are given simultaneously.

We can also have non parametric estimation of the probability density function, because in many cases I do not need to know what is the exact form of the probability density function? But what we have to find out is p of x given ω or p of ω given x . So, given a value of x , I have to estimate what is the probability density function or the corresponding or what is the value of the probability density probability for the corresponding value of x . So, I need not have the knowledge of the entire parametric form of the probability density function, rather if I know what is the value of x for a value of x and this value of p x has to estimated from the given set of samples. That is what is called non parametric form. I can have both types of estimation whether it is in the non parametric form or in the non parametric form.

Now, today I will discuss about a problem, which is dimensionality problem. That as we said that the feature vectors are of dimension d or every component of the feature vector represents certain aspects of the pattern. Or it gives some description of the pattern which we discussed in our earlier classes when we talked about feature extraction processes. So, when I have d numbers of the components of the feature vector, each of these components give you some characteristics of the pattern. When all this different characteristics or d number of characteristics are taken together, I can have some sort of

discriminating power given by this feature vectors among different patterns. So, today we will discuss that, what is the problem if we have more number of features in the feature vector or if the dimensionality of the feature vector increases. So, that is what is the dimensionality problem? Now, let us go to the Baye's decision theory or minimum rate classification or we said that if I know the probability density function.

(Refer Slide Time: 03:16)



So, let me assume that we are having two classes. One is class omega 1 and the other one is class omega 2. So, I consider two classes, omega 1 and omega 2 and I also assume that say p of omega 1, that is a priori probability of class omega 1 and the a priori probability of class omega 2, p of omega they are same. So, by assuming this if I plot the probability density functions p omega 1 and p omega 2 and let me also assume that the variance of these two classes are also same. That is sigma 1 square is equal to sigma 2 square.

So, I also assume that along with this, so p omega 1 is equal to p omega. I also assume that sigma squared is equal to sigma squared for simplicity. That means both the probability density functions, they have the same size and same shape, but the difference is the means are different. That is mu is not equal to mu, obviously because mu and mu are also same. Then I have the same probability density functions, there is not different. So, given this if I plot p of x given omega and p of x given omega 1, so let me plot in this

side p of x given ω and suppose the probability density functions are something like this.

So, this is 1, this extends infinitely then the other one, something like they have the same form. So, they have the same shape and same size, only thing is one shifted from the other because the μ 's are different, μ_1 is different than μ_2 . This is for class ω_1 and this is for class ω_2 or else this is μ_1 and this is μ_2 . Then we have said earlier when we talked about base decision theory, that whenever the two probabilities are same then that is my boundary. So, for all values of x which are greater than this, that is p of x given ω_1 is greater than p of x given ω_2 , obviously it is nothing but p of x given ω_1 into p of ω_1 , which is greater than p of x given ω_2 into p of ω_2 . That was our base decision theory, but because here we are assuming that p of ω_1 is equal to p of ω_2 .

So, it comes to be p of x given ω_1 greater than p of x given ω_2 . So, for all values of x lying on the right hand side of this point, we will decide that it belongs to the class ω_1 . And for all values of x lying on the left hand side of this point we will decide that it comes to class ω_2 . Then we have also said when we talked about the base or minimum error rate classification, that the error in taking this decision because whenever I decide in favor of class ω_1 , that is a finite probability. It may belong to class ω_2 as well.

So, the error involved in taking this decision and that we said earlier is nothing but the area under this curve and it extends indefinitely. So, this is error in taking decision, whether I decide in favor of ω_1 or I decide in favor of ω_2 . So, this area under this probability curve, that gives me the total amount error. Now, we find that if I shift this μ , that is instead of μ being here, if μ is this class ω_1 probability density function is something like this.

So, suppose this becomes my μ , if this is the μ then this is my decision boundary and if this is the decision boundary then the error will be reduced to this. So, earlier my error was bounded over this curve, now the error is bounded. Error is bounded under this curve. So, definitely the total error in the earlier case is higher than the total error I get in this case when the separation between μ_1 and μ_2 becomes large. When the separation between the two means μ_1 and μ_2 is low then the total error increases. If

the separation between the two means μ_1 and μ_2 is large then the total error decreases.

So, that clearly indicates what are the features, which are best suitable for classification purpose. It is the features whose means for different classes are widely apart. Those are the features best suited for classification. If the means are quite near very near, then those features does not give you much of discriminating problem, though even if the means differ by some amount. However small it is they still have some discriminating power, but the discriminating powers of those features, where the means are closer means for different classes are closer is much less than the discriminating power of those features whose means are widely apart.

So, usual practice in pattern recognition problem is that you start with some features finding find out what is the performance of classification. If the performance of classification, that means misclassification rate is not satisfactory, that is the performance is poor. You try to incorporate more and more features because I said if the features means of the features are different, they are not same. They have some discriminating power, but the extent of the discriminating power depends upon the distance between the means. As the distance increases, the discriminating power increases. As the distance decreases, the discriminating power also decreases. If I find with certain number of features the performance of classification is not very good, you try to incorporate more and more features.

If I try to incorporate more and more features, then effectively what I am doing is the dimensionality of the feature vectors I am going to increase. So, the value of d will increase as we incorporate more and more features in the feature vector, but that leads to another problem. Should we go on increasing the dimension of the feature vector without any control? That is what is the dimensionality problem? So, let us see what kind of problem that we face if we go on increasing the dimensionality of the feature vector, ok?

(Refer Slide Time: 11:50)

$$g(x) = -\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

$$\mu_i = \frac{1}{n} \sum_{k=1}^n x_k \approx (nd)$$

$$\Sigma_i = \frac{1}{n} \sum_{k=1}^n (x_k - \mu_i)(x_k - \mu_i)^T \approx (nd^2)$$

$O(nd^2)$ $O(\underset{\uparrow}{c} \underset{\uparrow}{n} \underset{\uparrow}{d}^2)$.

So, we find that earlier we said that that function $g(x)$ or discriminating function of different classes is given by minus half x minus μ transpose Σ inverses into x minus μ minus d by log of π minus half log of the determinant of the covariance matrix plus log of $p \omega_i$. So, all these have to be $i \times$ minus μ_i Σ_i inverses into x minus μ_i and this is Σ_i . So, this was discriminating function for the higher class, right?

Now, see that what is the amount of computation involved in this particular case, out of this we said that this term minus d by log of π . This is class independent, this does not depend upon the class. So, we can as well remove it from the definition of discriminating function, because for all the classes this value will remain. So, this does not give you any discriminating power, but the other terms they really discriminate among different classes. Now, from here when I try to compute the value of μ_i , you assume that I have n number of samples, training samples belonging to class ω_i .

So, when I compute the value of μ_i , value of μ_i is nothing but 1 upon n summation of x_k , where k varies from 1 to n , right? So, I have to incorporate n number of summations followed by one division, and this x_k . That is the feature vector of dimension d for every component. I have to compute n number of summations, that means the total number of summations that I have to perform is n into d , right? I can assume that for this division it takes constant time.

So, the computational complexity to find out the mean vector is n into d , which is dimensional dependent. It depends upon the number of vectors, at the same time it depends upon d which is the dimensionality of the vector, right? Now, coming to the other term computation of this σ_i or σ_i inverse, ok? You find that this σ_i , the definition of σ_i was 1 over n into summation x_k minus μ_i into x_k minus μ_i transpose, where k varies from 1 to n , right?

So, this is vector of dimension d by n and this is a vector by d , right? This is a vector of dimension d by n . This is the vector of dimension by d , when I compute I get a matrix of dimension d by d . That is d squared and following similar manner if I try to find out is the amount of computation will be involved in this case or the computation of the σ_i . You will find that these steps are the computation of the order of n into d squared. So, to find the mean vector I need a computation of dimension n into d to find the covariance matrix, I need a computation of the order n into d squared.

Similarly, over here this also takes a computation of order n into d squared. Of course, this one is constant, not constant. It depends upon the value of n because when I want to compute ω_i , I have to take n number of samples in picture. So, this is of the dimension n complexity n . So, overall you find the dimensional complexity of this discriminating function because this is the highest one. So, I can take the overall complexity is of the order n into d squared, not only that I have seen number of classes, right? For every class I have to compute this discriminating function and I have to find out which one is maximum.

So, before I get the maximum, I have to compute the discriminating function for each and every class and have c number of classes for every class. I have a computational complexity of n into d squared. So, for c number of classes the total computational complexity will be of order c into n into d squared. So, here you find that the complete computational complexity linear in terms of c , that is the number of classes linear in terms of n . That is the number of samples in the class, but it is quadratic or square in terms of the dimensionality, right?

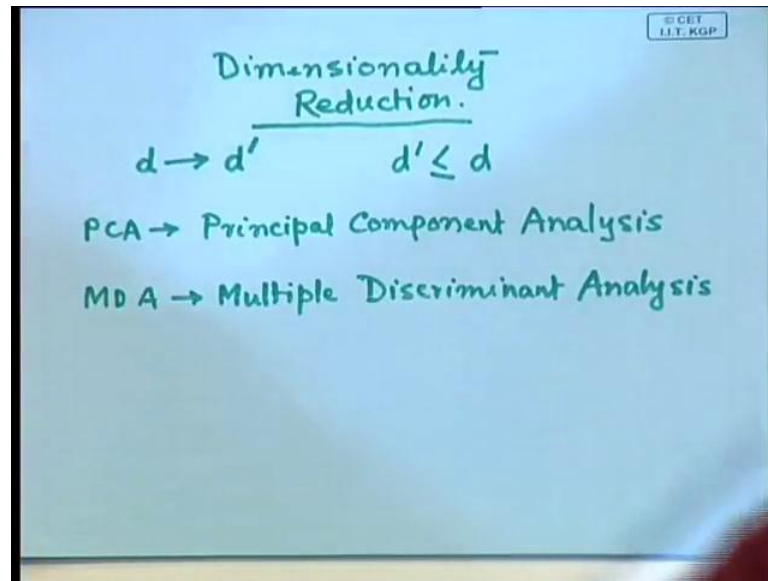
If the dimensionality is increased by a factor of 2 , the computational complexity is raised by a factor of 4 . If it is increased by a factor of 3 , computational complexity becomes 9 times, right? So, this dimensionality of the feature vector is very, very crucial determined

how efficient your classifier will be. So, it is not always good that you go on increasing the feature of the dimensional vector by incorporating more and more features. It is true that if I incorporate more and more features, the classification performance might be better, but at the same time the classifier becomes inefficient.

So, we have to think of the ways in which the dimensionality of the feature vectors can be reduced. That means whether it is possible that initially I start with feature vectors of higher dimension, but whether it is possible to reduce the dimensionality, but still maintaining the separability power of the selected features. That means what we have to do is given a feature vector of higher dimension, I have to take the projection of those higher dimensional feature vectors onto a lower dimension. But this projection has to be in such a way that the space on which, the space your dimension on which I am projecting these feature vectors, that space should be as orthogonal as possible.

That means the different features they should not be correlated because one of the factors when I incorporate, more and more features are arbitrary. I do not guarantee those features will be orthogonal, that means one feature may have some correlation with another feature. But when I try to project over to lower dimensional space, I should make sure that this lower dimensional space is orthogonal. That means whatever feature I try to capture, one feature should not depend on the other feature. So, these features should be independent. So, coming to this concept of projection with the end to reduce the dimensionality of the feature vectors, I can have two types of projections or two types of concepts.

(Refer Slide Time: 20:44)



So, what we are discussing now is the reduction of dimensionality. That means initially I have a feature vector of dimension d . I want to reduce this to feature vector dimension of d' , where d' is less than or equal to d . Obviously, I do not want d' to be larger than d . At most it can be equal to d , but I target d' less than d so that the dimension is reduced.

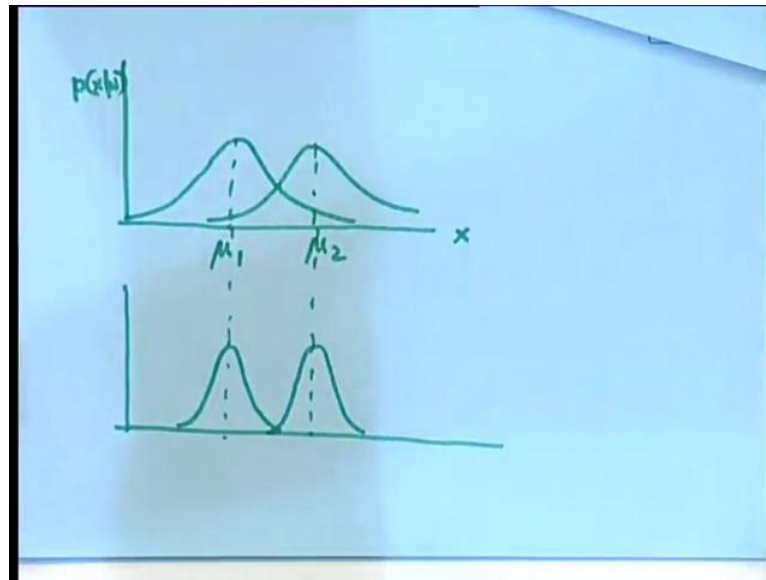
Now, while I try to do that there are two approaches, one is PCA or principal component analyses. The principal component analyses or PCA approach is that, you try to project your higher dimensional feature vectors on to a space of lower dimension in such a way that, in the lower dimension whatever feature vector I get or projected vector I get, that best represents the initial feature vector, that I have to get the original feature vector, that I have to get.

So, when I say best represents, it is the best representation in terms of list squared error. That is the list square error between your initial feature vectors or original feature vectors and the reduced feature vectors that should be minimum. The other approach is called MDA that is multiple discriminant analyses. The aim of multiple discriminant analyses is after a while we are doing or trying to manipulate all these feature vectors, because we want to separate among the feature vectors which belong to different classes.

So, the multiple discriminant analyses, what it has to do is when you type take a projection on to a different space on to a q different feature space, it tries to separate the

mean vectors of different classes and at the same time tries to make the samples belonging to the same class more compact. That means within class scatter is reduced. Whereas, between classes scatter is increased, ok? That is quite natural because here what we said is coming to this diagram or let me redraw this diagram.

(Refer Slide Time: 24:13)

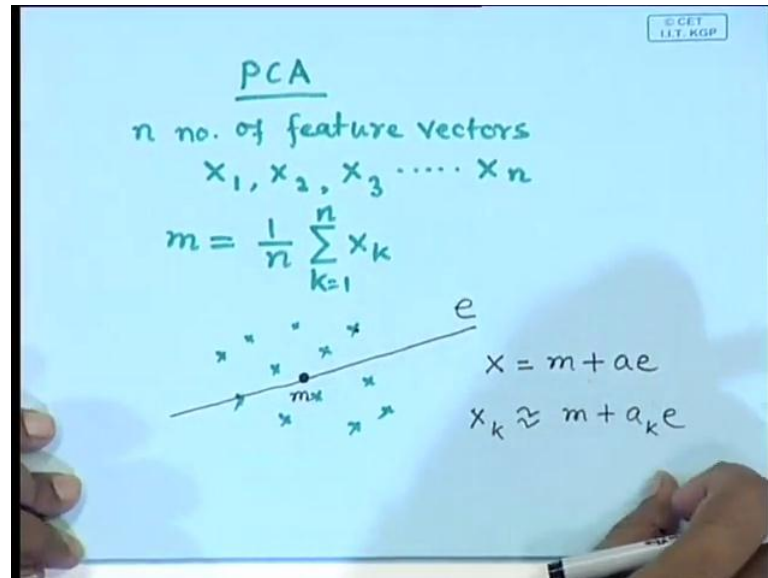


This is the x , this is $p(x|\omega)$ given ω , ok? So, suppose the distribution is something like this. So, this is μ_1 and this is μ_2 , we said that we increase the distance between the μ_1 and μ_2 . My classification error will be less, that is 1 and at the same time if I try to reduce the variance of the samples within a class, that means keeping μ_1 and μ_2 same in the same location. So, this remains μ_1 and this remains μ_2 , but now I want to have a distribution something like this. So, you find that the distance between the μ 's, μ_1 and μ_2 remains the same, but within class scatter or the variance of the samples belonging to the same class are reduced. Then also my classification will be better, that is error rate will be less.

So, this MDA approach or multiple discriminant analysis is what it does is, it tries to increase the distance between the means. That is the inter class scatter, that it tries to increase and at the same time it tries to reduce the within class scatter or tries to reduce the variance of the feature vectors on to the projected space. So, you benefit in both ways, one is the increase the distance between the means and simultaneously reduce the variance within the class so that the overall error is minimized. So, these are two

different approaches, one is principal component analyses, other one is multiple discriminant analyses, ok? Today, what we will discuss is the principal component analyses approach.

(Refer Slide Time: 26:36)



So, we will talk about the PCA or principal component analyses. So, our problem is like this, that we have been given n number of feature vectors, every feature vector is of dimension d. So, I have n number of feature vectors. The feature vectors are x_1, x_2, x_3 up to x_n each of the feature vector is of dimension d, right? Obviously, the mean of these feature vectors is given by $\frac{1}{n} \sum_{k=1}^n x_k$, k varies from 1 to n. That is the mean of the feature vector.

Now, let us try to see that initially I have the dimensionality of the feature vector that is equal to d. Let me take extreme case that I want to represent these by a zero dimensional feature vector, right? So, if I try to represent this by a zero dimensional feature vector that means within feature vectors I do not have any variance. So, that is nothing but this mean. So, this entire set of feature vectors is represented by mean vector, but while doing that what I will lose is the variability within the data which is also not good, right?

So, instead of zero dimensional features, let us assume that I want to represent this dimensional feature vectors by d feature vectors, pardon meaning the feature, but the entire n number of features is represented by only one feature vector. While doing that I lose the variance because I have a single representation. Now, what I am saying is I do

not want to do that I will represent this d dimensional feature vectors by one dimensional feature vector. I will still have n number of feature vectors, the number of feature vectors will remain the same.

So, there will be variability, but the dimension of every feature vector instead of d will be 1. So, what I am trying to do is a vector getting mapped to a scalar, because I am reducing the dimensionality to 1. Let me take step by step approach so that what should be such best type of projection. If I want to project in one dimension and then whether I can project on to a multiple dimensions. If I want to project on to multiple dimensions what should be nature of such space? So, if I want to project on to a single dimension. So, suppose I have the set of points something like this, they are betraying ally distributed. I have n number of such points and the mean m is somewhere over here, ok?

So, what I do is if I want to represent this d dimensional feature vectors by a one dimensional feature vector. That means I have to take, I have to represent this feature vectors somehow on to a line. The feature d dimensional feature vector will somehow mapped to different points on a line, right? So, let us take a line passing through the mean of this feature vectors and let me assume that unit vector in the direction of this line is given by e , right? So, I am in space x , within the space I have a line passing through the mean of sample given feature vectors and the direction of this line I am saying it is e , right?

So, when I do this, you find this the equation of this line will be given by x is equal to m plus $a e$, because it passes through m . That is the mean of the given feature vectors and equation of this line if a is equal to 0. Then x equal to m , that means I am at this point m , right? For different values of a , I am moving from m in the direction of a . That means I am moving along this line.

So, the equation of this straight line in the direction of e , where e_i is an unit vector passing through the mean m is given by x equal to m plus $a e$, right? This a now represents positions of different points on this line, given this I can represent a point x_k on this line and that x_k point, x_k I can represent by m plus $a k$ time c . We will see what will be the value of a , the best possible value of a for different points I will have different values of this a .

So, this a_k represents that a point x_k , where is it met on this line. Now, while doing this definitely I have incorporate some amount of error, because x_k is of dimension d . I am representing this x_k by a line by a point on the line. So, obviously while doing so I have incorporated some amount of error, and what is that error? This is nothing but m plus a_k times e minus x_k , x_k is the original one and this is the one that I have represented. So, what I can do is for taken all these lines together or all these points together I can define an error function or a criteria function, ok?

(Refer Slide Time: 33:27)

The image shows a handwritten derivation of the cost function J and its partial derivative. The equations are as follows:

$$J(a_1, a_2, \dots, a_n, e) = \sum_{k=1}^n \|(m + a_k e) - x_k\|^2$$

$$= \sum_{k=1}^n \|a_k e - (x_k - m)\|^2$$

$$= \sum a_k^2 \|e\|^2 - 2 \sum a_k e^t (x_k - m) + \sum \|x_k - m\|^2$$

At the bottom left, the partial derivative is given as:

$$\frac{\partial J}{\partial a} = 0$$

So, that error function or the criteria function is nothing but we represent it as this is the initial way, represent it by J . Obviously, it will be a function of a_1, a_2 up to a_n and it will also be the function of the direction of the line which is e , right? This is nothing but m plus $a_k e$ minus x_k , take the mod squared, take the summation over k varying from 1 to n . So, this is the squared error within summation, then this becomes sum of squared error, right? What we try to do is, we try to minimize this sum of squared error in terms of a , right?

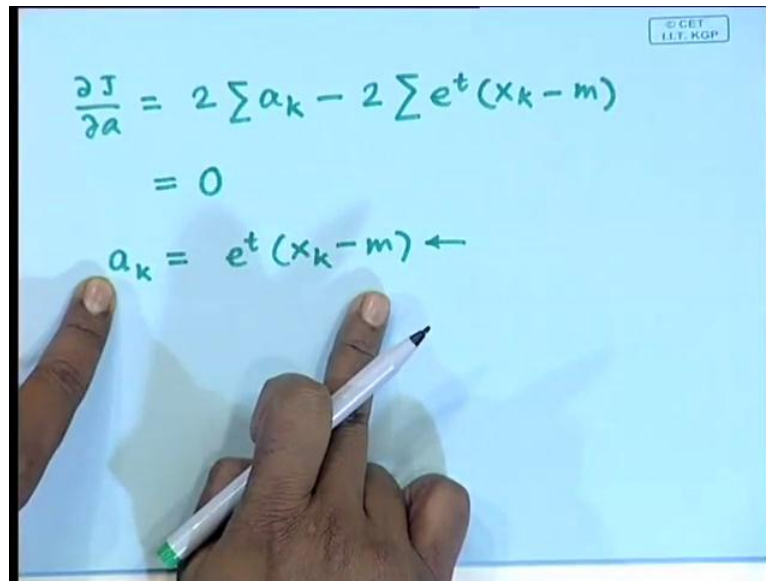
So, I want to find out that what are these ways or what are the positions of the straight line by which my d dimensional vector should be represented, assuming e to be fixed. I do not touch e for the time being, that is the direction of the line to be faced. So, given a line how these points have to be mapped on to the different points on the line. How the feature vectors have to be mapped on to different points on the straight line so that your

sum of squared error will be minimized. So, this expression now I can write by slight reorientation. This can be written as $\sum_k (x_k - m)^2$, take the summation over k varying from 1 to n . This x_k is same as the previous one, I just have simply rewritten this. It is same x_k , I am representing as $m + a_k$ into e . Otherwise, how do I get the error? This x_k is my original point and this is the projected point.

So, the difference between the original point and the projected point, that is my error. I am trying to estimate the error, right? So, this is the same as my original x_k . So, this if this square I expand, and it simply becomes sum of a_k square into mod of e squared minus 2 summation $a_k e^T (x_k - m)$ plus sum of $(x_k - m)^2$. I am not writing the limits of summation. The limit is n varying from 1 to n . So, this is my sum of squared error. What I want to do is I want to reduce, minimize the sum of squared error by varying the values of a .

So, what I have to do for that is take the differentiation of this with respect to a , equate that to zero, solve for values of a effectively. What I want to do is, I want to take gradient of J with respect to a and equate that to 0. And by solving this I get the values of different a that comes later. I said I will go stage by stage. Initially I am saying that assuming the direction of the line to be fixed, let us not try to handle all together. Dimensionality of e is same as e because it is a unit vector in the line in d dimensional space. It is a vector in d dimensional space. So, obviously the dimensionality of e will also be d . If I want to represent a vector in c dimensional space, the dimensionality of the vector is obviously is 3, ok?

(Refer Slide Time: 38:38)

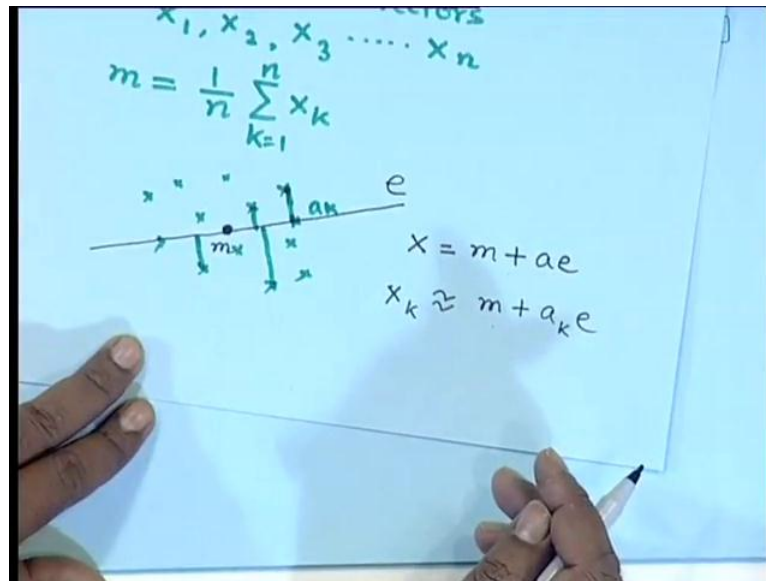

$$\begin{aligned}\frac{\partial J}{\partial a} &= 2 \sum a_k - 2 \sum e^t (x_k - m) \\ &= 0 \\ a_k &= e^t (x_k - m) \leftarrow\end{aligned}$$

So, when I take the differentiability of this with respect to a , what I get is effectively $\text{del } j$, $\text{del } a$ nothing but 2 into summation of a_k minus e transpose x_k minus m , ok? So, if I take the differentiation of this with respect to a , then what I get is if I take the differentiation of this. And what I forgot is the mod of e , that is equal to 1 because it is unit vector in the direction of the line.

So, mod of e is equal to 1 . So, effectively what I have is summation of a_k square. Take the differentiation that becomes two times a_k , take the differentiation of this, I get because it is a_k into e^t transpose x_k minus m . If I take to differentiate with respect to a , I am left with simply e^t transpose e^t into x_k minus m e^t transpose x_k minus m . This term is independent of a . So, differentiate this with respect to a , it becomes 0 , right? So, after differentiating this term, assuming by making use of the fact that modulus of e is equal to 1 , I get this output, right? I have to equate this to 0 .

So, when I equate this to 0 , I get simply a_k is equal to e^t transpose x_k minus m , right? Equating this to 0 , I get $a_k e^t$ transpose x_k minus m . Now, what is this? e is the unit vector in the direction from the line, right? So, this simply says that a_k is nothing but orthogonal projection of x_k on to a line, the direction of p passing through the mean m . It is just orthogonal projection of point x_k on to a line in the direction of p passing through the mean point m , right? So, simply what we get is when I have these sets of points, this a_k represents these projected point on to this line.

(Refer Slide Time: 41:20)



Now, how do I complete this? This is nothing but projections of different a_k 's. Projections of different a_k 's on to this line and these projection are nothing but orthogonal projection and that gives the position a_k . So, the best kind of representation in one dimension is to project the d dimensional feature vectors take the orthogonal projection of the d dimensional feature vectors on to a line passing through the centroid, ok?

Now, see that our assumption was we started with a given line, whose direction was known as e and for the given line the best type of projection is taken orthogonal projections on to that line. So, given a line and the best representation is orthogonal projection on to the line. Now, which given line is best, that is the question that he has raised, so far we have considered that our line is given. Now, which particular direction of the line is best, that is what direction is best? Let us try to answer that. So, for doing this I start with the same one that.

(Refer Slide Time: 42:48)

$$\begin{aligned}
 J(e) &= \sum a_k^2 - 2 \sum a_k e^t (x_k - m) \\
 &\quad + \sum \|x_k - m\|^2 \\
 &= - \sum [e^t (x_k - m) (x_k - m)^t e] \\
 &\quad + \sum \|x_k - m\|^2 \\
 &= - e^t \left[\sum (x_k - m) (x_k - m)^t \right] e \\
 &\quad + \sum \|x_k - m\|^2
 \end{aligned}$$

Now, I have to find the direction of the line, that is e . Now, I have determined that to find out s it should be orthogonal projection. Now, I have to determine that which direction of the line, which line will be best, that means what should be best e . So, I write this J , the criteria function as a function of e , you remember? This e is not error, I am not saying this as error. This is the unit vector in the direction of the line J e is equal to a_k squared minus. What did I have for that expression? y a minus y a, a is a vector having different a_k 's as components. When I say differentiation, it is gradient because in the vector space I cannot differentiate in the vector space. I have to take gradient which are nothing but components of that gradient comes as partial differentiation with respect to individual components.

So, I had J e e is equal to J k squared minus $a_k e$ transpose x_k minus m plus sum of x_k minus m square. I am not writing the limit of summation. Now, this a_k we have found that it is the same e transpose x_k minus m . So, this I can simply write as this one. So, this will be a_k squared. This will also a_k squared, this is minus a_k squared minus summation of a_k squared. This is plus summation of a_k squared, effectively I get minus summation of a_k squared and a_k is nothing but e transpose x_k minus m . So, I can rewrite this expression as minus summation of e transpose x_k minus m into x_k minus m transpose into e . See here what I have is this is the case first, this is minus. There should be a 2 over here, y a minus 2, 2 a_k squared because e transpose x_k minus m is nothing but a_k .

So, effectively I have minus a k square plus a k square minus a k square. So, I have minus a k square and a k is nothing but e transpose x k minus m. So, it is minus summation of e transpose x k minus m square. If I break that, I can write this as e transpose x k minus m into x k minus m transpose e, it is in the matrix expression. So, this plus sum of the other term remains as it is, x k minus m square. Now, this e is independent of k. So, this e terms I can take out of summation expression.

So, this I can write as minus e transpose summation x k minus m into x k minus m transpose e, because e is independent of k. So, I can take this e out of the summation expression, because this summation is over k, k varying from 1 to n, right? Plus summation of x k minus m square. So, it remains the same, right? Now, you find what is this term, sum of x k minus m into x k minus m transpose.

(Refer Slide Time: 47:37)

$$\sum = \frac{1}{n} \sum_{k=1}^n (x_k - m)(x_k - m)^t$$

If you remember from our earlier expression, what is our co variance matrix sigma upon n summation x k minus m into x k minus m transpose. Take the summation k equal to 1 to n and here you find that this term is nothing but a scaled version of this term. If I multiply this co variance matrix this is upon n, sorry if I multiply the co variance matrix by n, which is the number of samples over which I am trying to compute the co variance matrix. I get this term and this is what is called scatter matrix.

So, it simply represents that how that data is spread in the space because you remember that from your one dimensional Gaussian function variance is more. The curve is flat if

the variance is less, the curve becomes sharp. That means the point is distributed over a smaller space, if the variance is small. The points are distributed over a larger space if the variance is large, same is for co variance matrix. The co variance tells you shape as well as size, that means to what extent that data is distributed and whether the data distribution is oriented in some particular directions. Only that is what is captured by co variance matrix, same is the case of scatter matrix. Scatter matrix gives you the same information, but with a scale factor of n. So, if I multiply co variance matrix by the number of samples n, I get the scatter matrix, right?

(Refer Slide Time: 49:40)

© CET
I.I.T. KGP

$$J(e) = \underbrace{-e^T S e}_{\substack{\downarrow \\ \text{maximization of } e^T S e}} + \sum \|x_k - m\|^2$$

Lagrangian \rightarrow

$$u = e^T S e - \lambda (e^T e - 1)$$

$$\frac{\partial u}{\partial e} = 2 S e - 2 \lambda e = 0$$

$$S e = \lambda e \Rightarrow \text{Eigenvalue Expression}$$

$e \rightarrow$ an eigen vector of S
 $\lambda \rightarrow$ eigen value.

So, this expression now becomes this $J(e)$, see if I simply write this, rewrite this it becomes $J(e)$ is equal to minus $e^T S e$. That is the scatter matrix, e plus summation of x_k minus m square. Now, what is our aim? I want to minimize this $J(e)$ by varying e , this should be within bracket, as a function of e $J(e)$ by varying e , varying e and you find this term is independent of e . So, if I want to minimize $J(e)$, what I have to do is, I have to maximize this. It is a variability sign, minimization of $J(e)$ effectively means maximization of minus $e^T S e$, sorry maximization of $e^T S e$.

So, if I maximize this $e^T S e$, effectively I am minimizing this criteria function $J(e)$. So, how what I have to look for is the maximization of this $e^T S e$ and for this maximization, if I make use of Lagrangian. So, I can have an expression of the form u is

equal to $e^T (S - \lambda I) e$. So, I have to maximize this using Lagrangian, subject to the constraint $\|e\| = 1$.

So, to maximize what I have to do is I have to take the differentiation of this with respect to e . So, what I have to compute is $\frac{\partial}{\partial e} e^T (S - \lambda I) e$ and if compute $\frac{\partial}{\partial e} e^T (S - \lambda I) e$ then over here I simply get expression $(S - \lambda I) e$. Differentiate this with respect to e , I get $(S - \lambda I) e = 0$. I have to equate this to 0, right? If I equate this to 0, I get simple expression $S e = \lambda e$. What is this expression? This is an Eigen value expression.

So, this simply an Eigen value expression, where the vector e is the Eigen vector of S and what is λ ? It is the corresponding Eigen value. Now, you remember I have to maximize. Our aim is to maximize $e^T S e$, right? Our $S e = \lambda e$, that means I have to maximize $e^T \lambda e$. Our constant is $\|e\| = 1$. So, if I want to maximize $e^T \lambda e$, that means I have to take the maximum value of λ , that effectively says that when I am considering the direction of the line in the direction of the vector e . From here I find that this e is nothing but the Eigen vector and which Eigen vector I have to consider?

The Eigen vector corresponding to the maximum Eigen value and by using whatever value of λ , I get because my ultimate aim is I have to find out to get the value of λ , that is what represents the point. So, that value of λ is called the principal component. That is why it is principal component analyses, people also call it $k-l$ transformation.

Now, you remember how this $k-l$ transformation came into picture? You had the covariance matrix, take the Eigen values of the Eigen vectors of the covariance matrix. Corresponding to the largest Eigen values, from that you form a transformation matrix and the transformation you get is the $k-l$ transformation. That is not like this and there we said whenever we talk about the $k-l$ transformation, we talk about principal component analyses and why this principal component. This is the reason it is principal component.

So, here you find that if I take only one Eigen vector, corresponding to the largest Eigen value, then the corresponding principal component is a scalar one and effectively my d dimensional feature vector I am converting to one dimensional feature vector. Now,

instead of considering only single Eigen vector, if I consider multiple Eigens of vector corresponding to largest Eigen values.

So, instead of considering only one Eigen vector if I consider d' , d' prime number of Eigen vectors, d' is less than or equal to d . Corresponding to d' prime number of largest prime values, then take the orthogonal projection of your feature vector, d dimensional feature vector on to those Eigen vectors. The corresponding principal components of the projection values, projections that you get that gives you d' prime dimensional feature vector and because my scatter matrix is real and symmetric the Eigen vectors are orthogonal. That means, the principal components that I get they are also orthogonal, and because they are orthogonal one is uncorrelated from the other, ok?

So, if I go for this principal component analyses, I can reduce the dimensionality. Simultaneously, I also ensure that my d' prime dimensional feature vector that I generate those feature vectors will be or the components will be uncorrelated. That means every component of the feature vector gives you independent information that no other component will give. So, let us stop here today. We will continue with MDA approach in the next class.