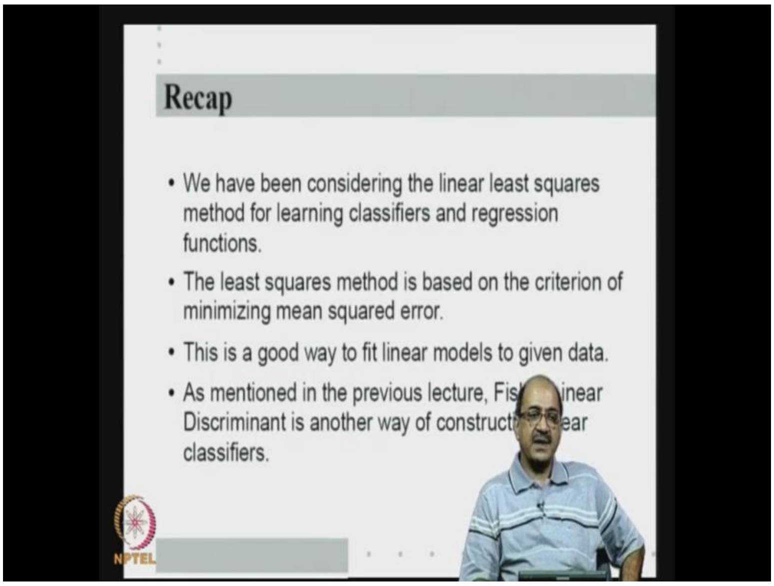
Pattern Recognition
Prof. P. S. Sastry
Department of Electronics and Communication Engineering
Indian Institute of Science, Bangalore

Lecture - 18
Fisher Linear Discriminant

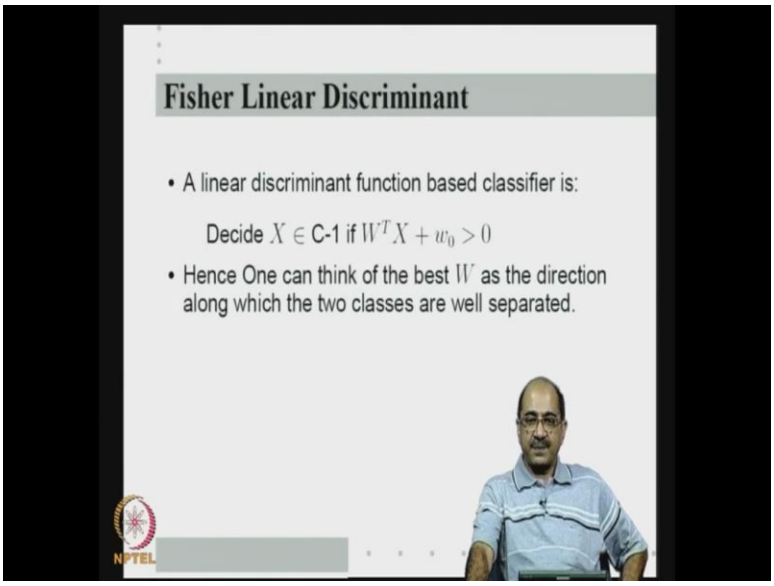
Hello and welcome to this next talk in the course and pattern technician. We have been considering linear classifiers for last so many lectures. Specifically, in the last few lectures we have been looking at the linear least squares approach both for classification and regression. Least square approach minimizes some of square of errors and we seen how we can use that to learn linear models and will continue with that now.

(Refer Slide Time: 00:51)



So, the least square method is based on the criterion of minimizing mean squared error. And we see how we can derive linear least squares algorithm for classifiers for learning regression functions. And we also looked at logistic regression of learning classifiers. There is one other approach to learn linear classifier which we very briefly mentioned at the end of the last class and that is the fisher linear discriminant while, least squares is also a good way of learning linear models and is a very very standard way. Fish linear discriminant actually somewhat predates the least squares approach, it is a very interesting approach to learning a linear classifiers. And today we will be looking at fisher linear discriminant as a method for constructing linear classifiers.

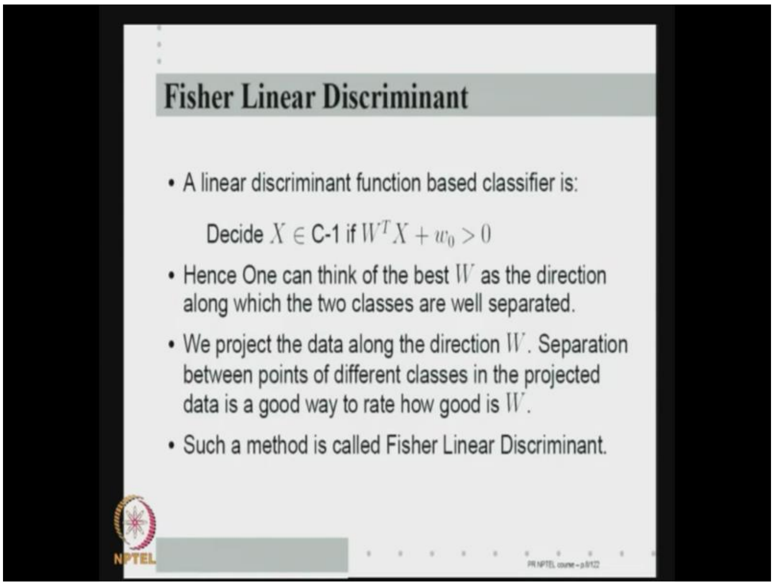
(Refer Slide Time: 01:54)



Any linear discriminant function base classifiers essentially, given a feature vector X; it decides that X belongs to class 1 if, W transpose X plus w naught greater than 0 for some given W and w naught. If W transpose X plus w naught is greater than 0 is one class less than 0 another class. that is a That is what a linear classifier linear discriminant function bayes classifiers is about. Now, because W is a vector, we can take it as a direction in the r d, the appropriate space. Then, W transpose X is nothing but the projected value of X onto the direction W. So, if I think of this inequality W transpose greater than minus w naught and if it is a good classifier. What it means is, if I take any X that belongs to class 1 and projected it onto W, the projected value, W transpose X is greater than minus w naught. And on the other hand, if I take any X in class C 0 and project it on to the direction W then, the projected value W transpose X less then minus w naught.

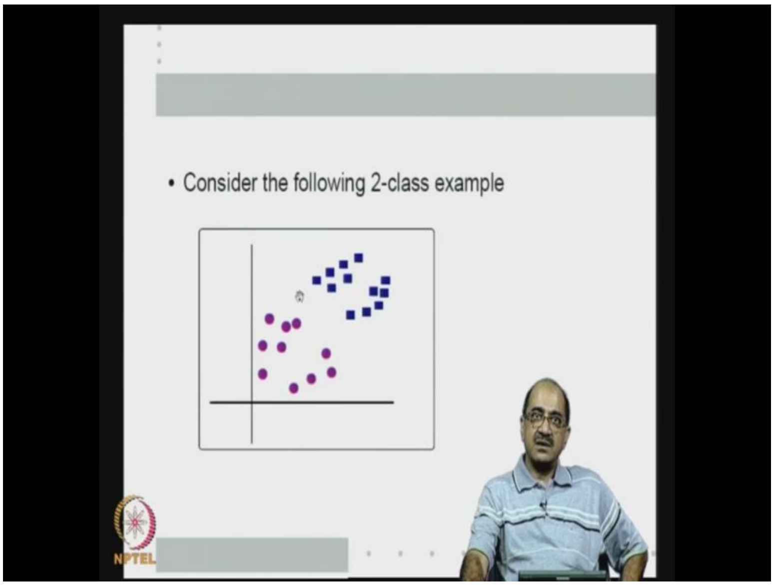
So, essentially what a linear discriminant function is doing is looking for a direction W. Such that, if I project the features vectors on to the direction then, along that direction the 2 class are well separated. On that direction there will be some threshold in this case minus w naught, some point along that direction. Such that, the projected versions of all features vectors are 1 class will be on one side of this threshold whereas, the projected versions are all the features vectors are the other class will be on the other side of this threshold along the direction W. so, we can think of the best W for a linear classifier to be a direction along which the 2 classes are separated. So, we are looking for a particular direction feature space. Such that, if you project all the vectors onto the directions, all the feature vectors onto direction, I get a very good separation between the feature vectors of the 2 classes.

(Refer Slide Time: 04:00)



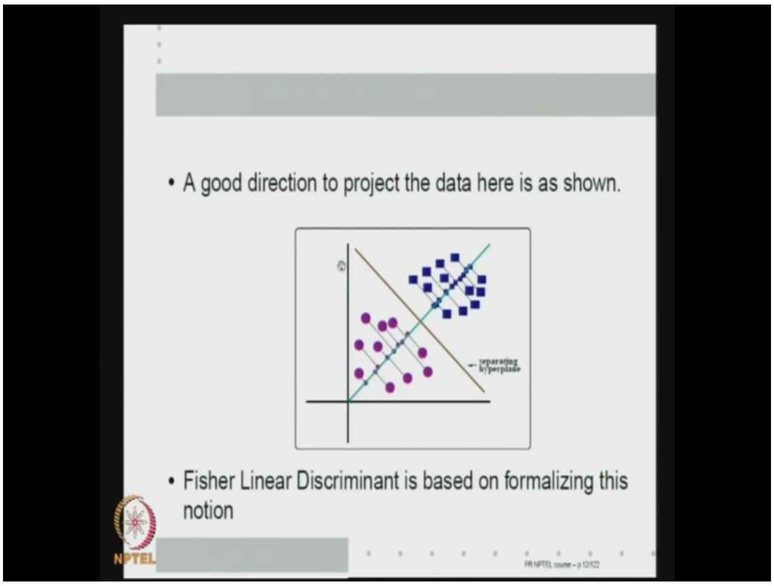
So, the idea is that we project the data along the direction W. And separation between points of different classes in the projected data is a good way to rate how good W is. So, now to learn a good W I can take different W's, project the data onto these W's, and I am asking which W is there which will maximize the separations between the classes. This is the basic idea fisher linear discriminant.

(Refer Slide Time: 04:28)



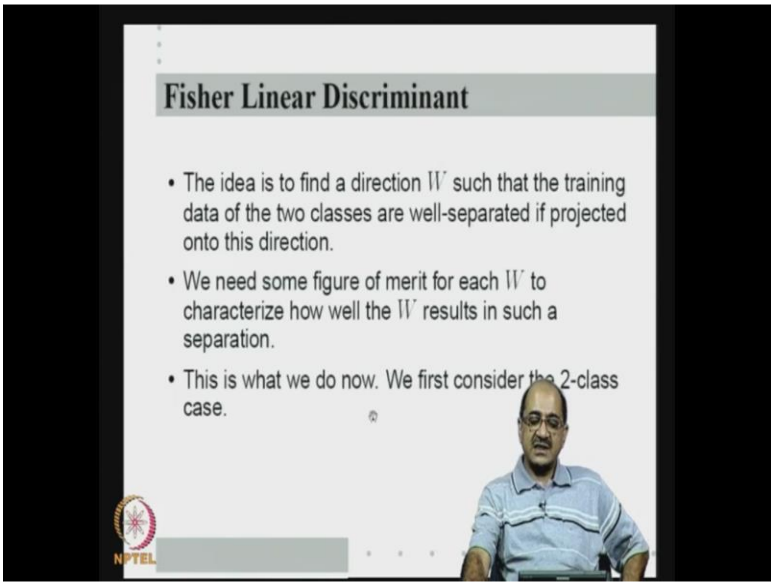
Before we go into the mathematically details let us look at a simple example. Let us say this is the 2-class problem: Reds are 1 class and blues are another class. If I project all data for example, onto the X axis, the reds and blues get inter mingled in X axis. Similarly, if I project it on to the y axis also they get inter mingled.

(Refer Slide Time: 04:47)



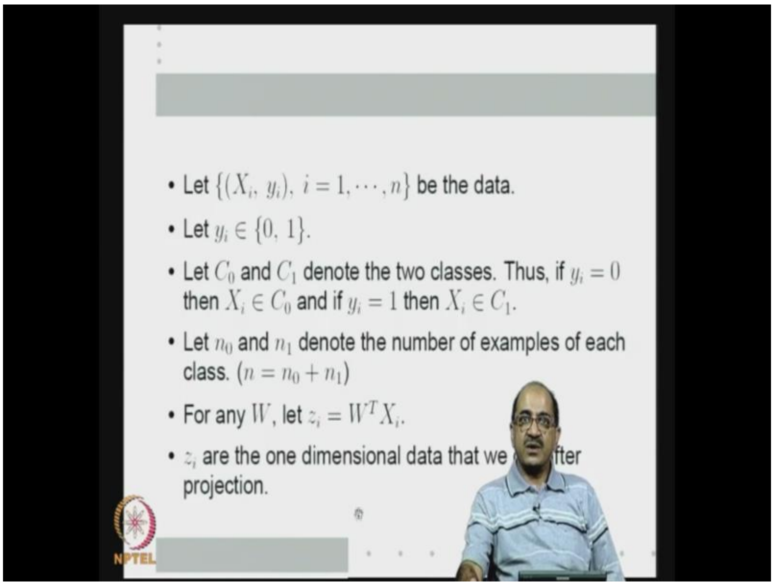
But, we can easily find a direction namely, this line. If I project the data along this line, as you can see; if I recode my 2 dimensional features vectors as 1 dimensional feature vectors by projecting them along this direction. Then, all 1-class patterns are one side and all other class patterns are other side. The actual separating hypo plane is perpendicular to this direction. So, If I think of W as this direction; obviously, W is perpendicular is the normal to the separating hypo plane. So, that will be separating hypo plane. So, I can think of finding the best W as finding a direction along which you have to project the data so that all the feature vectors have 1 class in the projected space or well separated from the feature vectors of the other class. So, fisher linear discriminant is a way of formalizing this notion of finding the best direction to project so that the 2 classes get well separated.

(Refer Slide Time: 05:49)



So, what does fisher linear discriminant do? We want to find a direction W, such that the training data the 2 classes are well separated if projected onto this direction. So, to do this off course we have to find a way of formalizing this notion of what you mean by well separated. So, essentially we need to have a figure of merit some some number that I can assign to W. So, I want some j of W which assigns a number to a W to figure of merit. Such that, the number tells me how well projection on that W will result in proper separation.

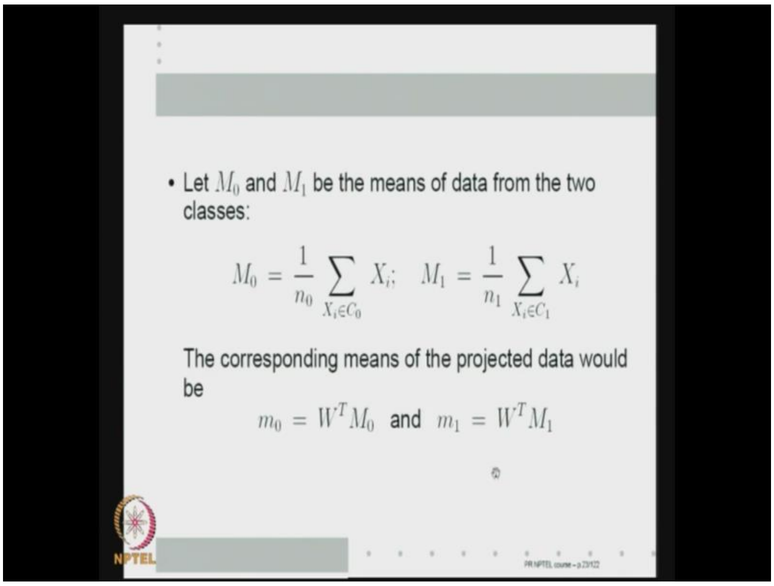
(Refer Slide Time: 06:29)



So, will do that, we will formalize this by first considering a 2-class case. So, let us get a notion right first as we been looking at our data is always X i, y I, this is our training data, we have n samples. X i are the feature vectors there in r d and y i is the class label. We are considering 2-class. So, let us assume y i 0 and 1. And this, 2 classes we will denote by C 0 and C 1.

So, what it means is when y i is 0 then, I can say X i belongs to C 0 and when y i is 1, I can say X i belongs to C 1.Let us say, out of the n samples, n 0 training samples are in class 0 that is C 0 and n 1 training samples from class C 1. Obviously, n is equals to n 0 plus n 1. For given any W, let z i denote W transpose X i. So, z i are the projected data along any direction W. So, z i are the one dimensional data that we get after projection. This is our notation.

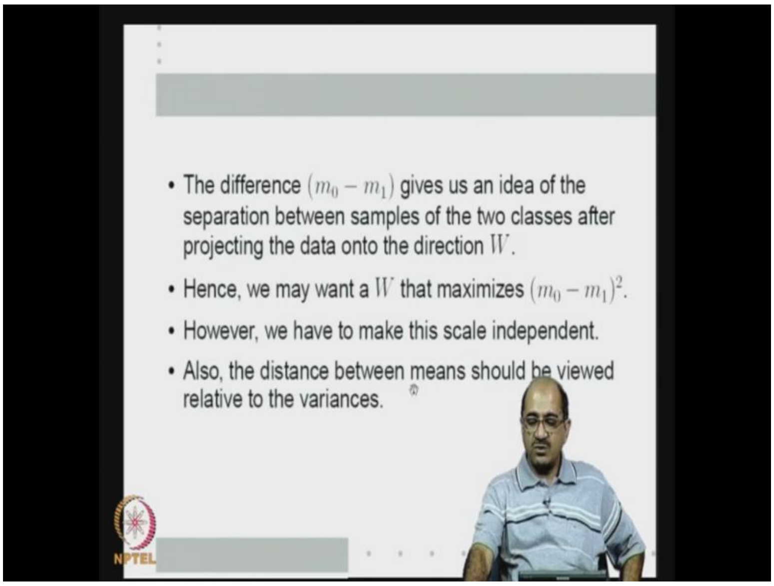
(Refer Slide Time: 07:35)



Let us say M 0 and M 1 denotes the means of the data from 2 classes. So, that essentially sample means so, M 0 is simply the mean of all X i that are in class 0 and there in M 0. So, 1 by M 0 summation X i, X i belonging to C 0 will give me the sample mean of the class 0 data vectors. Similarly, 1 by M 1 summation X i belongs to C 1 gives me the sample mean of class 1.

So, the corresponding means of the projected data, would be what? If I project X i onto W, I get W transpose X i. So, if I take mean of W transpose of X i will be same as W transpose M 0. So, on the projected data let us denote the mean by small m 0. So, the mean small m 0 will be W transpose capital M 0 and small M 1 will be W transpose capital M 1. So, these are the means of the projected data when data is projected onto direction W.

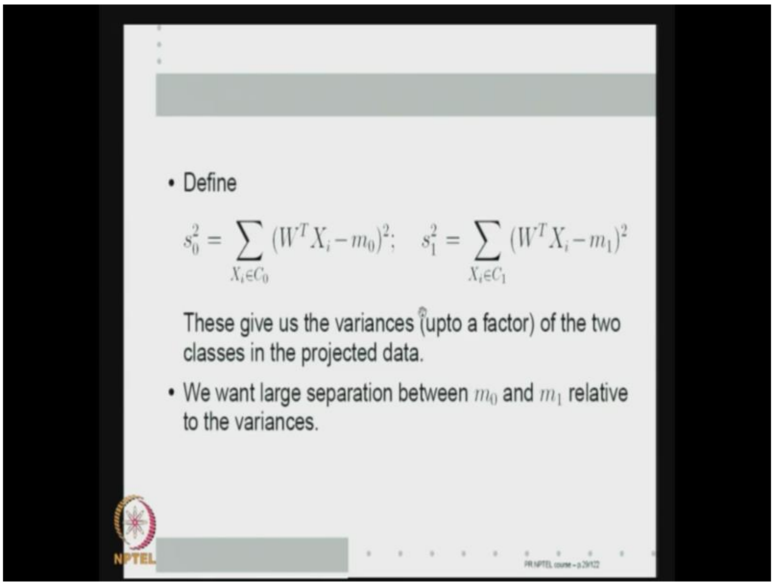
(Refer Slide Time: 08:42)



So, if I want good separation, what I mean is the difference between M 0 and M 1 should be large. So, we can say M 0 minus M 1 or M 1 minus M 0 is really does not matter gives us a good idea of the separation between the samples of 2 classes when the data is projected on to the direction W. So, we can say a good direction W is one which maximizes the separation. So, we may want W that maximizes M 0 minus M 1 whole square because that gives me large separation. But, there are 2 caveats here; just maximizing M 0 minus M 1 is not a good idea. Why? What is M 0 minus M 1? W transpose capital M 0 minus W transposes capital M 1 whole square. So, is W transpose capital M 0 minus capital M 1 whole square.

So, just by scaling W, I can increase this difference but, that is meaningless. A scale factor, a positive scale factor makes no difference to the linear classifier. So, first we have to find some function that is scale independent. But, more importantly what is good separation; depends on how much data spread is there. We are only looking at the distance between the means as we already seen we can consent bayes classifiers normal class conditional densities. Essentially, the difference between the mean relative to the variances is what tells us whether the classes are well separated or not. So, to look at the distance between the means M 0 minus M 1 in relative to the variances of the 2 classes. So, that is how we have to formulate our objective function W.

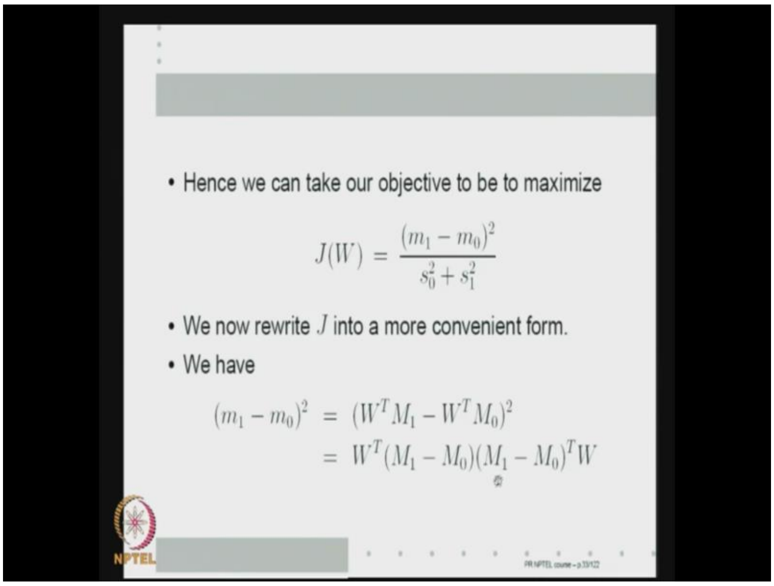
(Refer Slide Time: 10:34)



So, let us first calculate that is something like variance. As define as 0 square to be W transpose X i minus m 0 whole square. W transpose X i is the projected data that is z i. This is summation or X i belongs to C 0, for all X i in C 0 the mean of the projected data is m 0. So, W transpose X i minus m 0 whole square, sum dot X i belongs to C 0, is like the variance of the projected data of class C 0. Say like because I dint put 1 by m 0, I could have put 1 by m 0 but, we did not. So, this essentially just accept for a multiplicate, for a constant multiplicate factor; these essentially variance of the projected data from class C 0.

Similarly, if I define s 1 square to be summation or X i belongs to C 1 W transpose X i minus m 1 whole square because m 1 is the mean of the projected data of class 1. This is accept for multiplicative factor if the variances is the projected data in the for class 1. So, what we want? We want large separation between m 0 and m 1 relatively to the variances. That is our objective. So, I can formulate this as a objective function for W as follows.

(Refer Slide Time: 11:50)

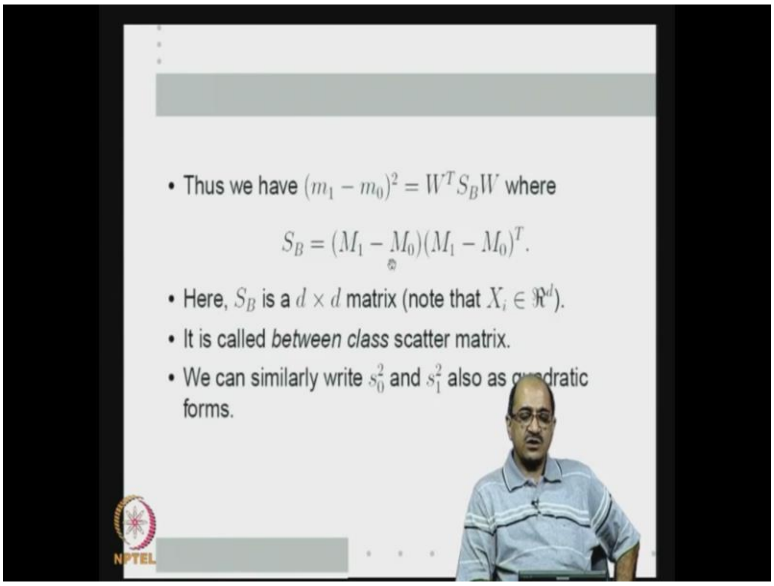


So, we say we want to maximize J of W, with J of W defined as m 1 minus m 0 whole square by s 0 square plus s 1 square. So, we want large separation m 1 minus m 0 whole square, relative to the variances s 0 square plus s 1 square. By the way even though this equation looks as if there is no W at the right hand side, you know all m 0, m 1, s 0, s 1 everything depends on W; s 0, s 1 square dependent on W.

Right; s 0, s 1 square dependent on my m 0 depends on W. So, this is a function of W. So, we want to maximize the difference between the means in the projected space relative to the variances. This off course is also scale independent that is not really evident right now but, will see it by rewriting this in a more convenient form. To rewrite this in a more convenient form; let us start with a numerator.

What is m 1 minus m 0 whole square? m 1 is W transpose capital M 1 minus W transpose capital M 0 whole square. This I can write as W transpose into M 1 minus M 0 the whole square. So, because W transpose M 1 minus M 0 is a vector, I can always write it as W transpose M 1 minus M 0 into M 1 minus M 0 transpose W. So, m 1 minus m 0 whole square can be written as W transpose M 1 minus M 0, M 1 minus M 0 transpose W. M 1 is M 1 and M 0 are the means of the data, if data is r d these are d vectors. Recall that our notation is all vectors are column vectors. So, m 1 minus m 0 is a d by 1 vector, M 1 minus M 0 transpose is a 1 by d vector. So, this is a matrix. So, this often called the outer product matrix for any 2 vectors.

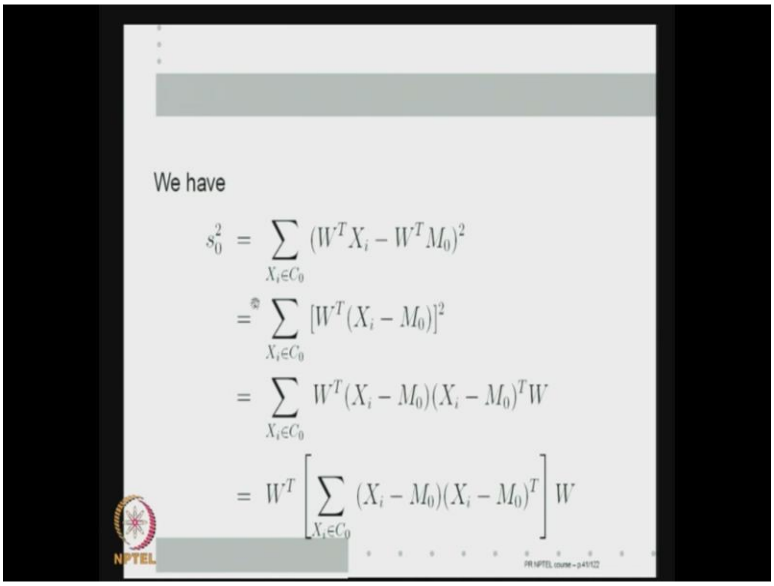
(Refer Slide Time: 13:51)



So, I can write; m 1 minus m 0 whole square as W transpose some matrix into W. So, I can write m 1 minus m 0 whole square as W transpose, thus called the matrix S subscript B. So, W transpose S B W. Whereas, B is M 1 minus M 0, M 1 minus M 0 transpose. As I said this is a d by 1 vector, this is a 1 by d vector so, this is a d by d matrix. Given any vector x x x transpose is called a outer product. An outer product is always a symmetric matrix. Also, I hope all of you remember that the outer product is a rank 1 matrix because all columns are multiple by the columns. So, is a rank 1 matrix.

So, such a matrix is d, this is a d by d matrix because our features vectors are on d. This matrix S B is often called between classes scatter matrix. So, I can write m 1 minus m 0 whole square as a quadratic form involving the matrix S B which is the between class scatter matrix which is defined by this. In the similar way, I can write s 0 square and s 1 square also as quadratic forms.

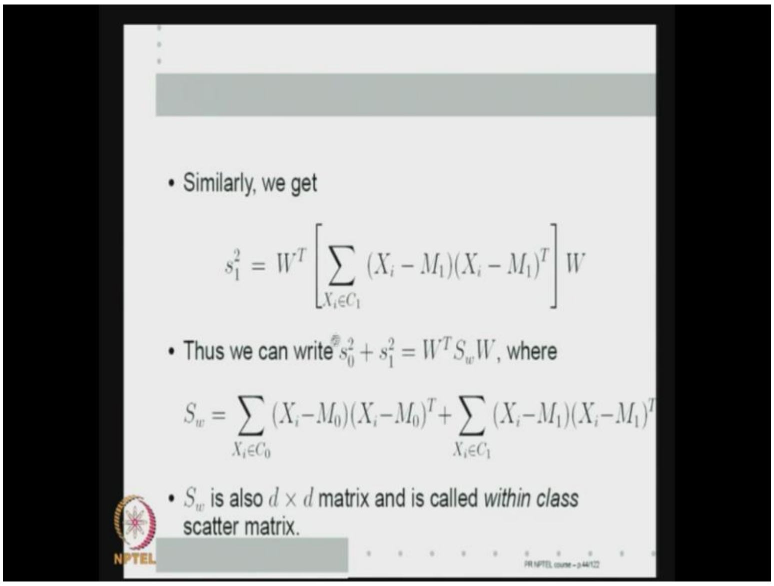
(Refer Slide Time: 15:05)



So, let us try to do that now. s 0 square by definition is W transpose X i minus small m 0 whole square. We know small m 0 is W transpose capital M 0. So, s 0 square is W transpose X i minus M 0 whole square summed over X i will learn to C 0. So, I can take or I can rewrite this as: W transpose into X i minus M 0 whole square. Now, once again just like what we did for earlier, I can write what is inside this summation in terms of outer products. So, I can write this as W transpose X i minus M 0 into X i minus M 0 transpose W. Now, the summation is over X I. So, W can come out of this summation. So, if I pull W out of this summation, I get W transpose summation X i belongs to C 0, X i minus M 0, X i minus M 0 transpose W.

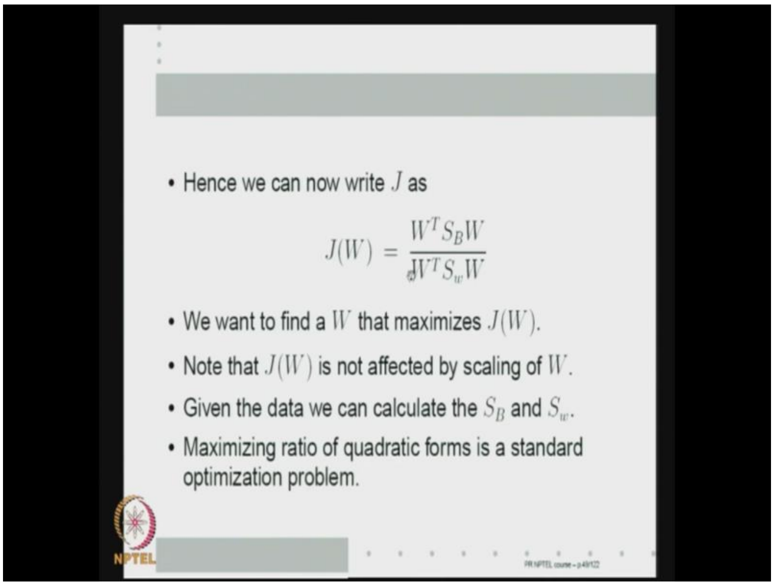
See, M 0 is the mean of all X i in class 0. So, if I did do this and divided by 1 by M 0; if you remember this is the maximum likely hood estimate for the co variance matrix. This is the m l estimate for the coefficient matrix of class 0. So, I can write s 0 square as W transpose X i minus into C 0, X i minus M 0, X i minus M 0 transpose W. This matrix off course is not at the coefficient matrix, if the coefficient matrix X i accepts for a multiplicative factor because I need 1 by M 0 to make it an estimate of the coefficient matrix. If I do the same thing for s 1, the only difference will be the summation will be C 1 and M 0 will become M 1.

(Refer Slide Time: 16:44)



So, similarly s 1 square will become W transpose the same matrix but, summation C 1 X i minus M 1 X i minus M 1 transpose W. Which now means, I can write s 0 square plus s 1 square as W transpose S w W. Where, S w is another symmetric matrix given by this sum of this 2 matrixes X belongs to C 0, X i minus M 0, X i minus M 0 transpose plus X belongs to C 1, X i minus M 1, X i minus M 1 transpose. So, this matrix, this S w is also the d by d matrix is called within class scatter matrix. Essentially, the first term here is propositional to the coefficient matrix of the first class of class C 0 and second term is propositional to the coefficient matrix of the class C 1 by matrix we mean the sample mean estimate of the coefficient matrix. So, S w is called the within class scatter matrix. So, we can write s 0 square plus s 1 square as a quadratic form on S w, M 1 minus M 0 whole square at the quadratic form on S B.

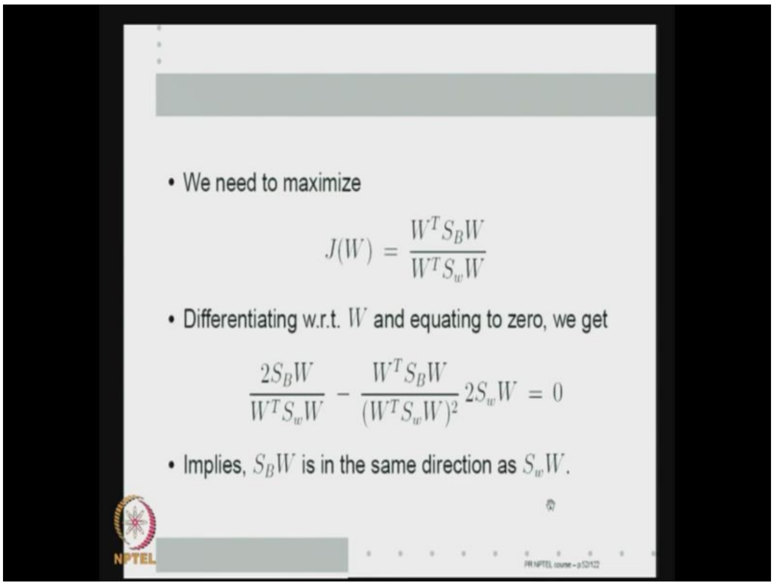
(Refer Slide Time: 18:03)



So, now J of W becomes W transpose S B W by W transpose S w W is a ratio of 2 quadratic forms. So, first thing to note so, basically this is the same J as earlier. So, we just read it on a originally J W is M 1 minus M 0 whole square by s 0 square by s 1 square because both the numerator and denominator can be written as quadratic forms on some matrixes we written them like that. So, this is the J that we want to maximize. Now, is very clear that is not effect by scaling. If W replaced by k W, the k will be cancelled from both numerator and denominator.

So, this W is not affected by, this J W is not effect by scaling of W as it should be. And it is easy enough to calculate, given the data I know how to calculate S B, I know how to calculate S w. So, I can calculate both the matrixes and hence given any W, I can calculate J W. And narration, this is very nice optimization problem. Essentially, you want given 2 matrixes as S w, you want to find a vector capital W which maximizes the ratio of quadratic forms. Maximizes ratio of quadratic forms say every standard optimization problem and we can solve it generally easily enough.

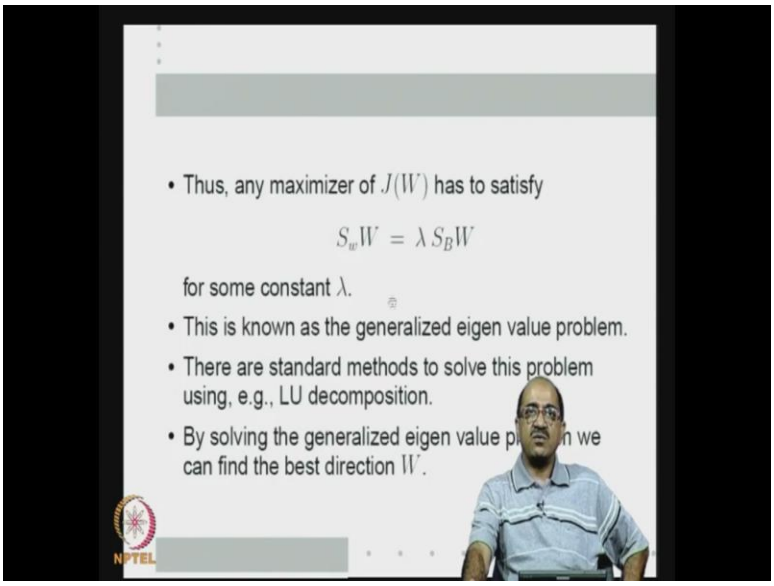
(Refer Slide Time: 19:24)



So, let us see how to do that. So, this is what we want to maximize. So, if you want to maximize, we differentiate or find the gradient and equate to 0. So, if I differentiate with respect to W and equate to 0, what will I get? I can first do, 1 by W transpose S w W into derivate of the numerator that will give me 2 times S B W plus the numerator W transpose S B W into derivative of 1 by W transpose S w W, I can write it as minus 1 by W transpose S w W whole square into the derivate of the S w W which is 2 S w W.

So, just by differentiating this and equating 0, I got this. In this, this is a quadratic form, this is denominator so, it is a scalar; this a quadratic form, that is a as a scalar; this is a quadratic form; that is a scalar. So, I have 1 vector here S B W, another vector here S w W. So, it is some constant to the vector S B W minus some constant vector S w W is equals to 0 that means, this vector and this vector are in the same direction because constant times 1 is the other. So, what this implies is that, the vector S B W is in the same direction as respective. So, the W that maximizes J is such that the matrix S B times W is a vector in same direction of the matrix S w times W.

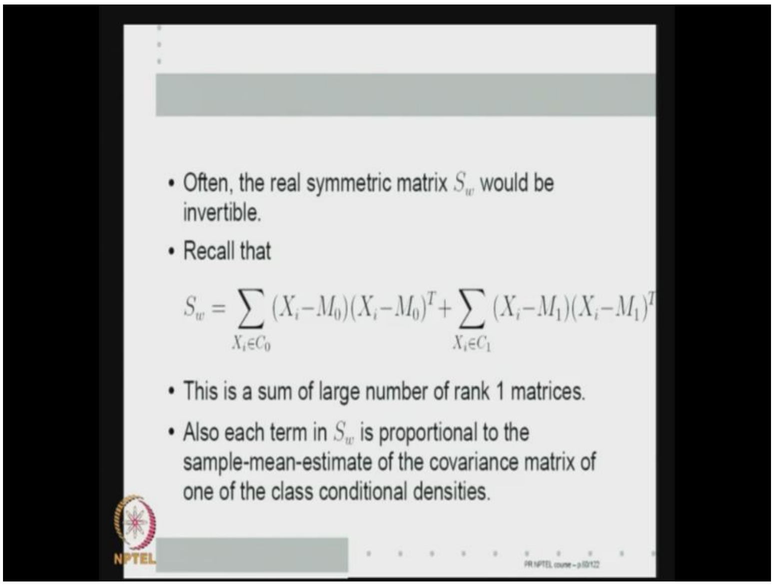
(Refer Slide Time: 20:56)



Thus, any maximize of J has to satisfy S w W is some constant times S B W. Where, lambda is the constant. This is reminiscent of the Eigen value problem. In Eigen value problem you get a X is equals to lambda X. So, some matrix into a vector, is lambda times the same vector. Here, I am not gaining the same vector but, some other matrix multiplies W. But, this is also very similar to the Eigen value problem. So, this is known as the generalized Eigen value problems.

The standard ways of solving the analyzed Eigen value problem, there are methods based on LU decomposition so on. So, once we know is a analyzed Eigen value problem; for example, your mat lab will have a very standard protein for solving this. But, anyway we will not get into the details of special methods to solve general Eigen value problems because often we can solve it easily but, right now we know that because the generalized Eigen value problem. By solving generalized Eigen value problem I can always find the best direction W.

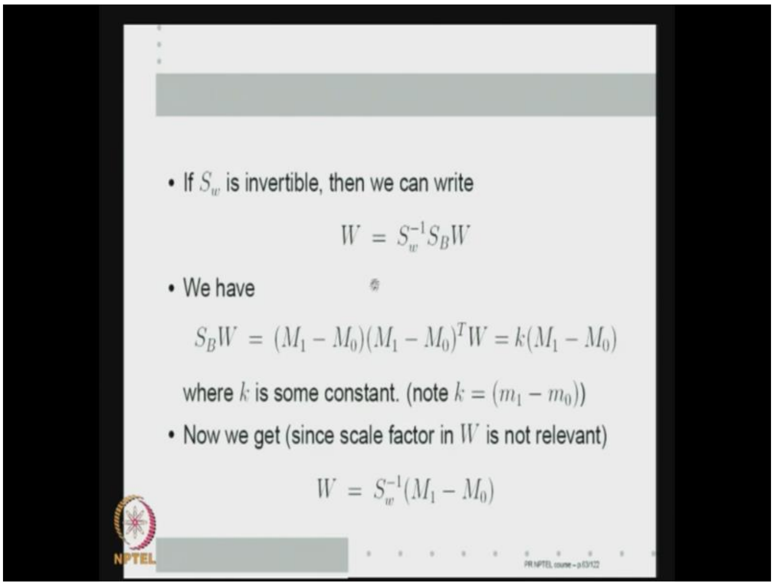
(Refer Slide Time: 22:04)



But as I say often, I may not have to solve the generalized Eigen value problem. This is because the real symmetric matrix S w is often invertible. And, why is that so? S w is this matrix, X i minus M 0, X i minus M 0 transpose, sum door X i belongs to C 0; X i minus M 1, X i minus M 1 transpose some door X i belongs C 1. There are atleast 2 reasons why we can suspect that this matrix is invertible. One is, we know for each i, X i minus M 0 into X i minus M 0 transpose in outer product is a rank one matrix. So, this adds lot of rank one matrix. When you add rank one matrixes suppose, I add x 2 minus M 0 into x 2 minus M 0 transpose to x 1 minus M 0 into x 1 minus M 0 transpose. If x 1 and x 2 are linearly independent then, this becomes a rank 2 matrix.

So, if I keep adding many rank one matrixes because the rank one at most go by 1 but, if we should the sufficiently many. So, the first reason is that this is a sum of a large number of rank one matrixes. If the number is large and X i are in general position, it is unlikely that all the X i will be in some lower dimensional subspace. Since, if all the X i are not in lower dimensional subspace then, adding all these rank one matrixes should give me a full rank matrix. Normally, the number of samples is much larger than the dimension d and hence, adding that many rank one matrix should give me a full rank matrix. Another reason, why we think why often this as a invertible is that as you seen, each term in this is a good estimate especially the data is large or prepositional to a good estimate of the covariance matrix X i and the covariance matrix will be invertible. So, by in both ways we accept S w to be invertible. Suppose, S w is invertible then, it is very much easier to solve for d.

(Refer Slide Time: 24:18)



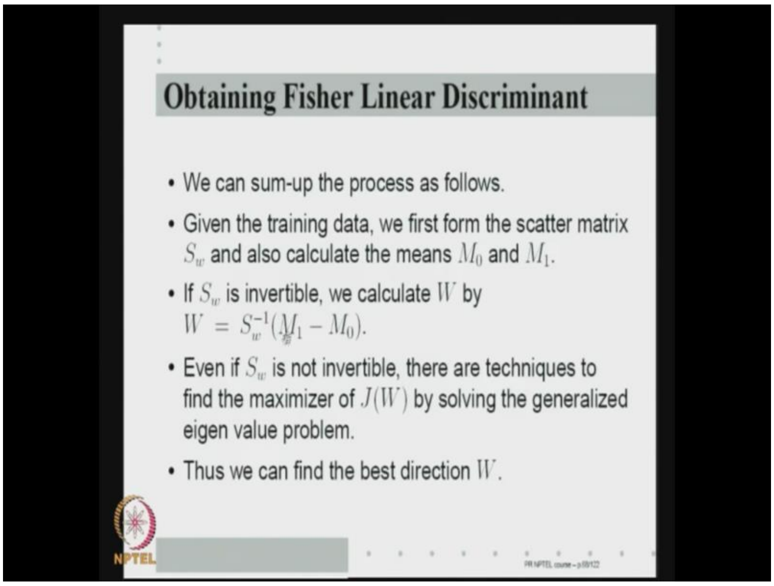
Let us say, S w is invertible. So, we have seen in the W has to satisfy S w into W is equals to lambda S B W. So, if this matrix is invertible, I can multiply by S w inverse. So, I get an equation for W or atleast relation W is equals to something. So, let us look at that relation.

Suppose, S w is invertible then, W can be written as S w inverse S B times W. There is There has to be lambda here, I omitted the lambda because ultimately constant factor is do not make much difference but, any way there is a lambda here. Let us not worry about that right now. This looks like off course an Eigen value problem. Putting a lambda here, it is the Eigen value of S w inverse S B. But, we do not even have to look at the Eigen values because S B W, what is S B? M 1 minus M 0, M 1 minus M 0 transpose into W. We know M 1 minus M 0 transpose W is nothing but little M 1 minus little M 0 which is scalar. So, this is some scalar k times M 1 minus M 0 as I wrote here, k is equals to M 1 minus little M 1 minus M 0.

So, S B W is a vector is propositional to M 1 minus M 0. So, I can replace S B W by some constant times M 1 minus M 0. So, which gives me W is S w inverse M 1 minus M 0. There are so many constants; there is a lambda here, there is a k here. So, ultimately there will be some constant times S w inverse M 1 minus M 0. But, as I said constant do not make difference in a linear classifier. Right. We can accordingly k W on b to anything that we want.

So, essentially we can find the W x surface scale factor to S w inverse M 1 minus M 0 and I do not want to worry about the scale factor, I can take this to be W because I am in a linear classifier context. So, this is the W that I can use as the fisher linear discriminant, if the matrix S w transpose to be invertible which as we just now said very often that would be the case.

(Refer Slide Time: 26:28)



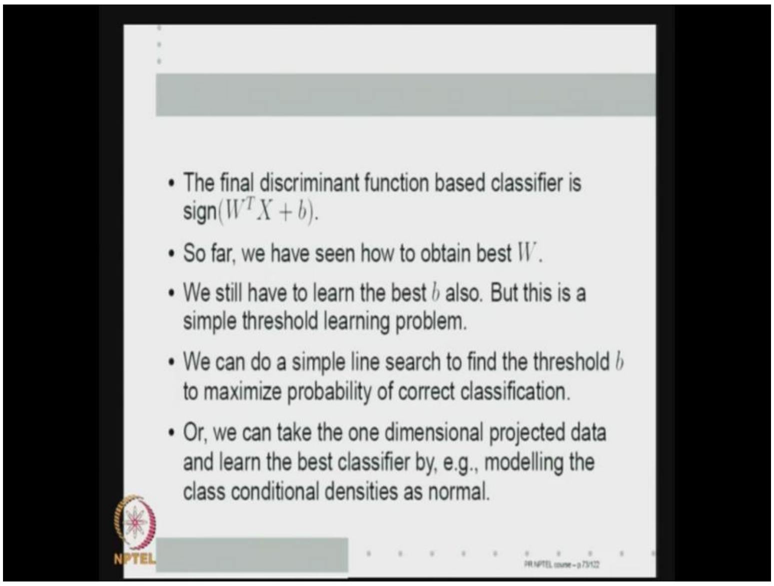
So, let us sum-up how we obtain fisher linear discriminant. With given the data, given the data, we first form the scatter matrix S w and also calculate the mean M 0 and M 1. Once the within class can I matrix S w is invertible, I can directly find W. As S w is invertible, I can calculate W by S w inverse M 1 minus M 0, that is the fisher linear discriminant direction. Or it is normal to the hypo plane classifier given by the fisher linear discriminant.

Just for completeness let us remember that, even if S w is not invertible there are techniques to find maximizes as a W by solving some generalized Eigen value problem. There are some techniques, even the generalized Eigen value problem. If S w is not invertible, there are some issues. I will just show that, so that you will appreciate that there are some issues that means just go back to.

If S w is rank deficient, what it would not mean is there can be some W at which W transpose W, W you can go to 0. Right. Thus, there would be some W for which this will be infinite but, thus not the W we want. So, we have to define this little more carefully. But, it does not matter there are techniques to take care of it. So, we will just we would not go there but, will just remember that.

If S w is invertible, W is given by this. Even, otherwise one can find a maximize by solving some generalized Eigen value problem. So, that is how we obtain the best direction W; very often this is what is no given as the fisher linear discriminant, as said in most pattern recognition problems S w would be invertible. So, this is the fisher linear discriminate. But, obviously we have found only the best direction W. We have not yet found a classifier. Right.

(Refer Slide Time: 28:36)

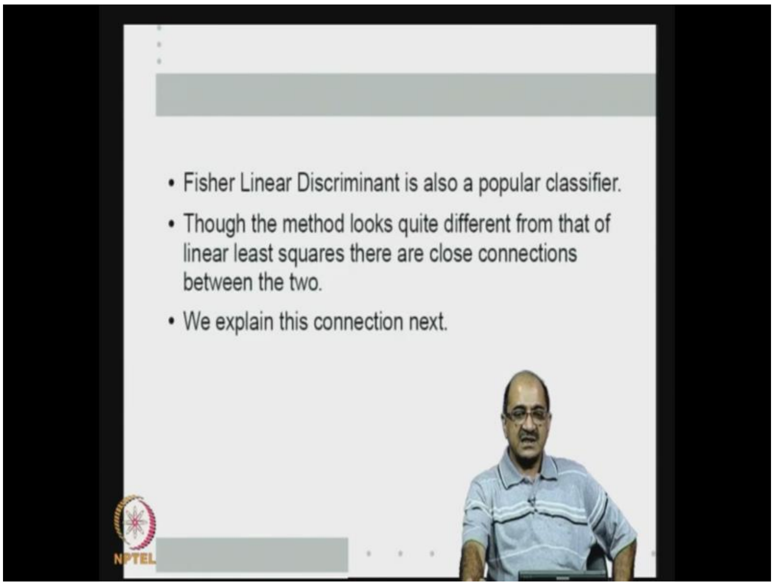


Let us remember that a discriminant based classifier is sign of W transpose X plus b. So, I have to find the b also. Right. All that we have done so far is how to find the best W. And have been found best W, to convert it to a classifier I still have to find W. But, this is a must simple of problem right. We have to tell on the best b but, this simple problem just learning a threshold. We are saying this is the best direction now, we have one dimensional data and I want the best threshold. So, now given one dimensional data and class labels, I just want a threshold bayes classifiers. I am asking which threshold is good, there are many methods.

For example, we considered ROC, how we can experimentally calculate ROC and come to a threshold; that is one thing. We can do we can do any that kind of such for a threshold. We essentially have to along W, we keep putting the threshold at various points and keep asking is my accuracy and the training data improving. I need only one threshold. So, I will start from some end of some point and keep moving, just like a line search in the optimization problem. So, we can do a simple line search to find the best threshold, to maximize the probability of a correct classification and the training data.

Or, what we do is, the projected one dimensional data z i that is, what we have with class labels. We can simply take it as a one dimensional pattern recognition problem and is very easy to solve one dimensional pattern recognition problem. For example, we can take the class conditional density to be normal, estimate the associated normal densities very easy to do for one dimensional as you have seen. And then, take the classifiers based on the estimated densities. This is also often done. So, in whichever way we do that gives us the final fisher linear discriminant. So, this is how I would obtain the fisher linear discriminant.

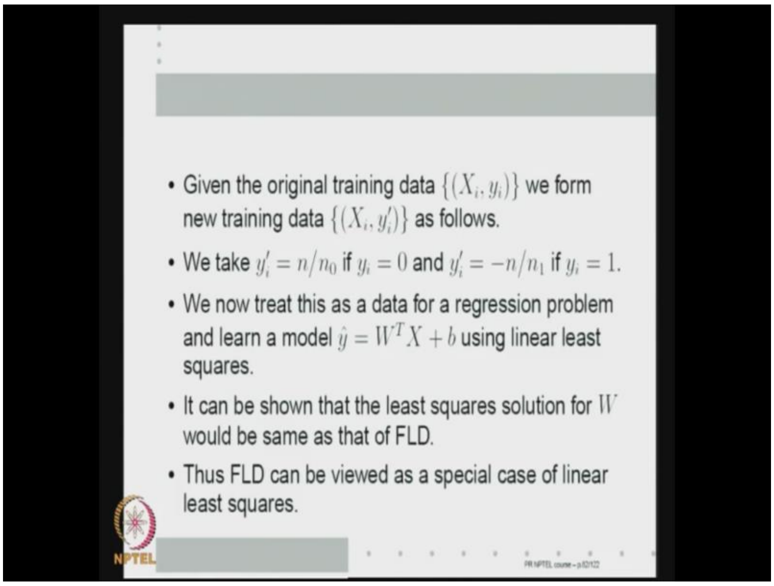
(Refer Slide Time: 30:37)



Officially, a discriminant is also a popular classifier, just like the least squares. Now, the way we derived fisher linear discriminant it looks quite different from the least squares method. As I had told you, a fisher linear discriminant came from also, came from statistic but, not it came from different direction.

So, essentially linear least squares and fisher linear discriminant evolved historically as different algorithms. However, how different it may look, there are close connections. As the matter of fact we can think of fisher linear discriminant as a special case of linear least squares. The reason we did not start there is that because historical is important, it is very often used in a applications and is nice to know another way of formulating what is a good linear classifier. But, having said all that, one can actually mathematically show the equalence of linear least squares and fisher discriminant in a in a very specialized sense. What is specialized sense; is the following.

(Refer Slide Time: 31:48)



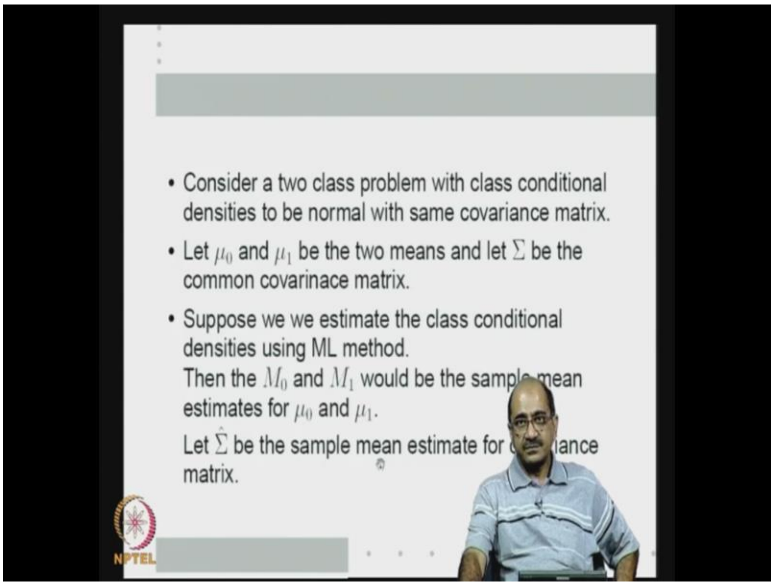
We start with some data X i, y i. This is our given data X i is in r d, y i is 0 1; this is our classification data. Given this data suppose, we form some new training data where, X i are same but, y i's change it to y i prime.

How do I form the new y i primes? Originally, y i's are 0 and 1. So, if y i is 0 that is, X i is coming from class 0, I take y i prime to be n by n 0. So, I am just changing the targets for all class 0 patterns, the correct value to predict is n by n 0. Where, n is the total number of samples and M 0 is the samples of class 0. On the other hand if y i happened to be 1, I take y i prime to be minus n by n 1. So, now I got new data X i, y i prime where, y i prime take some real numbers not 0 1 because n by n 0 and minus n by n 1 are some real numbers. Even though we take some real numbers we will think of y i prime as some generic real numbers. Then, we can treat this X i, y i prime data as a regression problem. X i is an r d, y i prime are a naught; I want a predictor, I want a linear predictor where, I has is W transpose x plus b.

Simply, viewing X i, y i as some data for a regression problem, I can certainly use least squares to find a linear predictor. What I have is W transpose X plus b. Using the outstand linear least square method that you have done in the last couple of lectures. One can show algebraically the algebra is a little tedious but it is there in one of the prescribed text book namely, Bishops book. But, it can be shown through somewhat long variant algebra that, if I done that, if I take this data recoded X i y i prime and then, think of this as a data for regression problem and then, find a linear regression. Then, one can show that the least square solution W that I obtain for fitting this model to this changed data will be exactly same as the W that we get out of fisher linear discriminant.

I am not prove this I am not proving this, as I said the algebra is tedious and it possibly does not add much score through the algebra, it is also there in bishop book. But, it should know that, it can be shown that the least square solution have obtained here would be this, would be same as the fisher linear discriminant. Thus, one can show or one can see that fisher linear discriminant can be viewed as he special case linear least squares. So, it is not really different from linear least square but, at the same it gives you another very important useful idea of how one looks at what is a good linear classifier.

(Refer Slide Time: 34:32)

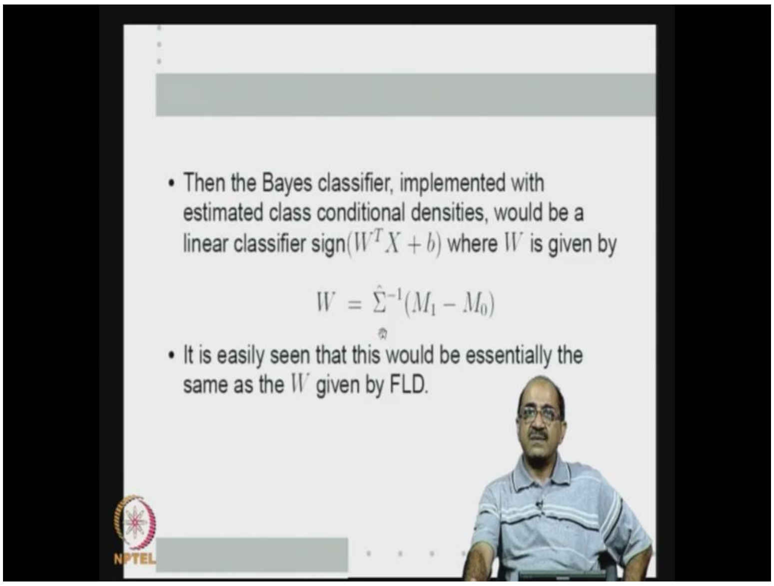


There is other ways of looking at how good fisher linear discriminant is. For example, we take simplest 2 class problem; let us say we have a 2 class problem with both the class conditional densities to be normal unless they say they have the same covariance matrix. Why same covariance matrix? If the 2 class of the same covariance matrix we know that the based optimal classifier is a linear classifier. Right.

So, now one can ask, will fisher linear discriminant give me the optimal linear classifier because fisher linear discriminant can give me only linear classifiers. We can only ask whether will give the optimal classifiers when the optimal classifier itself is a linear. So, let us take a case class conditional density is normal with same covariance matrix and we know that the based optimal is a linear classifier and will show that fisher linear discriminant will give me the same classifier as the based classifier. So, let us say mu 0 and mu 1 are the 2 means of the classifiers classes and because the covariance matrix is same, there is only 1 sigma for both classes. Thus, sigma is the common covariance matrix. Suppose, given this data if we want to actually implement bayes classifiers we would estimate class conditional densities; let us say using maximum likely hood method. If we give maximum likely hood method, we know mu 0 and mu 1 will be estimated in sample means. Similarly, a similar sample mean estimate exists for the covariance matrix.

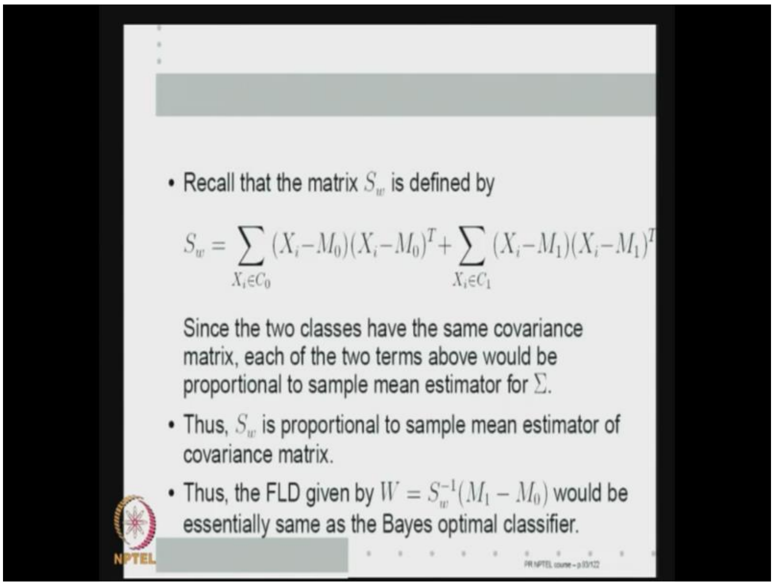
So, then we know that M 0 and M 1 would be the sample mean estimate is for mu0 that is how we obtain. M 0 is 1 by M 0, 1 by M 0 summation X i so, the sample mean estimated for mu 0 similarly, M 1. So, I would have obtained the same M 0, M 1 as the sample mean estimate for mu 0 and mu 1 had a try to do this estimation of class conditional densities under maximum likely hood. Similarly, I would I would have got some sigma hat as the estimate with the covariance matrix.

(Refer Slide Time: 36:28)



Then, what will be the bayes classifiers, we implement with this estimated quantities. If you still remember the bayes classifiers that we derived for common covariance matrix k. It is a linear classifier W transpose X plus b where, the W is given by sigma inverse mu 1 minus mu 0. So, what I would have implemented is sigma hat inverse M 1 minus M 0. Now, we know that this has to be same as the FLD because we know that the S w inverse that comes there is like covariance matrix. So, very very quickly will show that this W would be same as the W given by the fisher linear discriminant.

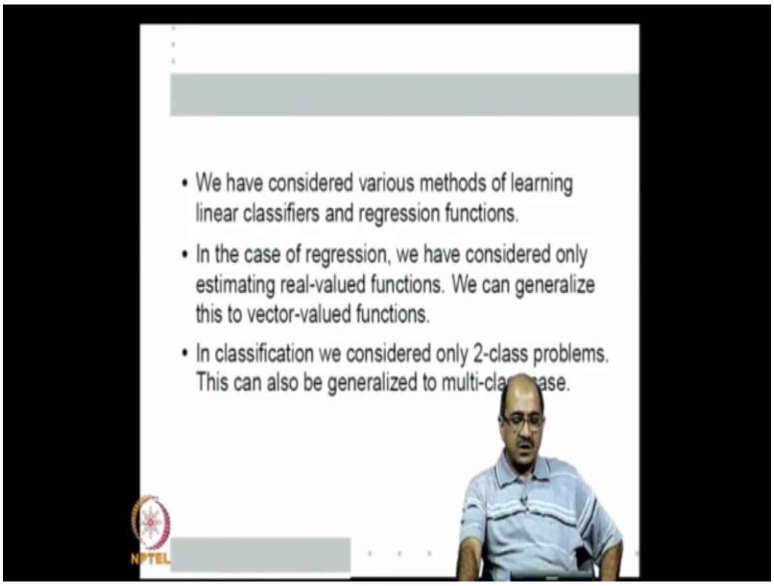
(Refer Slide Time: 37:12)



Coming back; this is the S w matrix. Because both classes are having the same covariance matrix, I could estimate the covariance matrix either from class 1 or from class 0. So, if I estimate this from class 0, that estimate would be propositional to this accept for a propositional constant. Simply, if I estimate it from class 1 it will be propositional to this accept propositional constant. And the classes are the same covariance matrix; each of the 2 terms above would be proportional with the same sample mean estimated. Right. So, the first term will be some a times sigma hat and the second term will be some b times sigma hat. So, essentially S w will be propositional to sigma hat.

So, since S w propositional to the sample mean estimate for sigma, we know that the fisher linear discriminant is given by S w inverse M 1 minus M 0 is same as the bayes optimal classifiers. So, fisher linear a fisher linear discriminant is an interesting classifiers because atleast in this kind of cases it gives you the bayes bayes optimal classifiers. So, it has a variant testing geometric interpretation namely, finding a direction along which classes are well separated. That is that is in itself is a nice interesting geometric interpretation. In addition, as you seen it can be viewed as a special case of least squares. And also, you know in this in the cases that bayes optimal is linear, I am iam getting atleast in the normal class conditional densities case. Fisher linear discriminant gives me the same classifier as the bayes optimal. So, that is the story of the fisher linear discriminant.

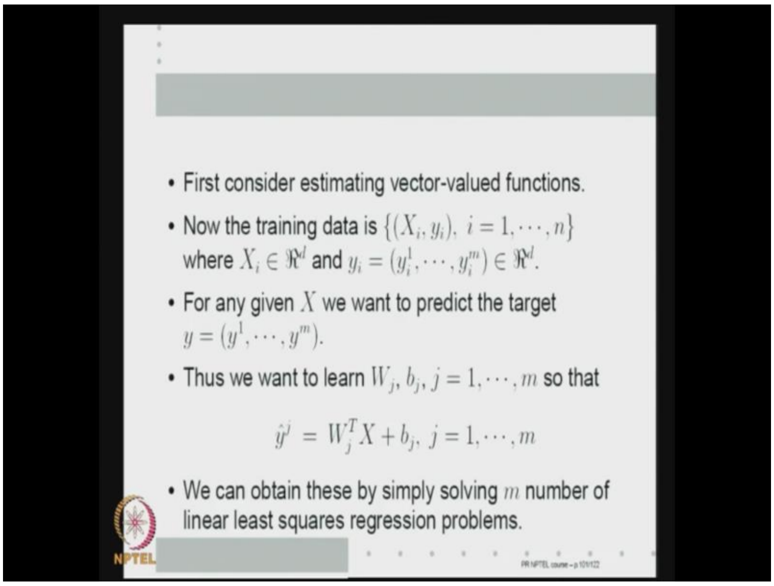
(Refer Slide Time: 39:06)



So, to move on; we consider various methods of learning linear classifiers and linear regression models starting with perceptron and then, we spent a lot of time on linear least squares methods and now, fisher linear discriminant. But, most of the time we are restricting ourselves when we doping regression we are talking of targets being in r in the regression case where we assume the training data as X i, y i where, y i belongs to r. What it means is, we are only estimating the real valued functions.

So, one generalization that we may want to do is to generalize this to vector valued functions. Right. Similarly, whenever we consider classifiers the classification problem we always restrict our self to do class problems so we have to generalize this to multi class problem. These are the 2 issues still left. So, in this class will atleast look at the issues involved and will tell you what the things are and pretty much that is all the least for multi class classifier. We may just briefly touch upon it next class and leave it at that. So, let us do it one by one.

(Refer Slide Time: 40:13)



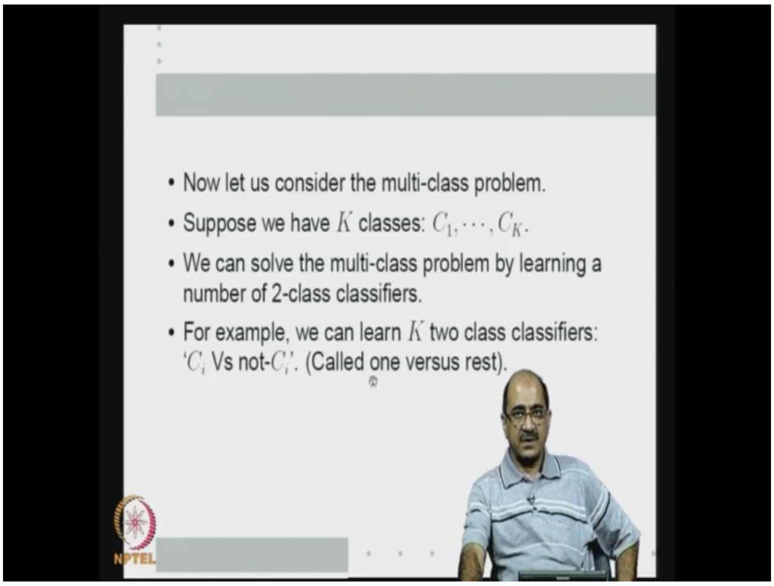
So, let us first look at the regression problem. First consider estimating vector-valued functions. What does vector-valued function mean? Now, my training data is given by X i, y i. As earlier, X i is a feature vector it belongs is an r d, y i is the target. Now, bacause I am learning vector value function let us say the functiom map from R d to R m. So, y i will be m vector. I am sorry this should not be r d, this should be r m. I am sorry about that. But, anyway y is a m vector. So, because we already put i for denoting the i'th sample.

Let us put, let us denote the components of y i by y i 1, y i 2 upto y i m. So, the main difference in the vector value case is that y i is now a vector of some m components for some orbitary value of m. So, given any X, we want a model that can prdict the target which is a m vector. So, given any X we want to predict y which is actually an m vector given whose components are y 1 to y n. And we want to do it using our basic linear methods. So, what it means is that, we want to learn m W's and b's such that, the j'th component can be predicted as some a fine function that is, W to j transpose X plus b j. So, essentially the problem falls down to find m vectors W j and m scalars b j.

So, that is a good model, my good model my prediction model would be the j the estimate of the j'th component in the target y hat j is W a traspose X plus b j. What is mean? If i just take this data and split it into many data's X I, y i 1 is 1 data set, X i y i is another data set, X i y i is another data set and so on. For each data set I find a standard least squares methods. Right. So, this learning a vector valued function is no different from simply solving m number of linear least squares regression problems.

So, simply by learning running m linear least squares regression problems, we can solve the vector vector case. Of course, we can all put it into a better formulism. We can actually put all these W j's as columns of a W matrix in d right a the entire least squares solution in a vector matrix notation. But, except for some complecation notation, there is nothing really conceptually linear over here because we are simply effectively running m number of linear least square regression problems. So, which means in principle, we know how to solve the ah linear model problem even if we have to lane a target function which is vector value.

(Refer Slide Time: 43:22)

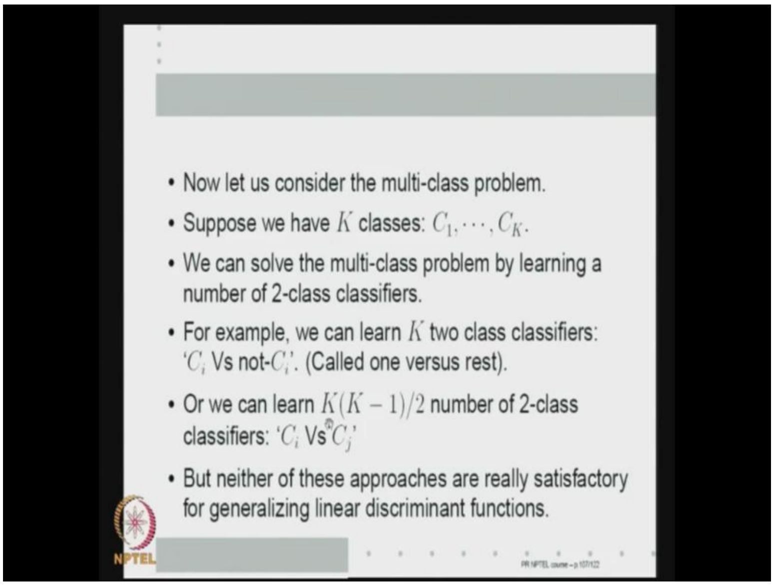


Now, let us come to the multi clas problem. If i know so far we know many methods for learning 2 class problems. So, If i know how to learn 2 class problems, can i find? Can i use the 2 lane multi class problem? The issue is just a little more complicated here, we have earlier considered this. So, let us say we have k classes C 1, C 2, C K. So, will come to decide how the data will be. For knowledge let us assume that we get X i y i then, y i takes any any any value between 1 and k.

Note that earlier we always been looking at our classes some C 0, C 1 so on. At this time we look at a C 1, C 2, C K. Now, as i said some time ago essentially 2 class problem is the more fundamental problem because we can solve multi class problems simply by if you know how to solve 2 class problem. So, we can in principle solve number of 2 class problems to solve a multi class problem. There are atlest 2 very generic techniques of a learnig in a multi class problem using 2 class methods. One is that i can learn K 2 class classifiers where, K is the number of classes as follows. I learn C i verses not-C i.

So, the i'th classifier I learn is learning to classifier C i verses not-C i. What you mean by C i verses not-C i? I have the data, a training data of the K class K's so, I take all the data, all the X i that belongs to a particular classes say C i or say all data that belongs to C 1 as 1 class all the rest of the data as another class. Then, I solve a 2 class problem whose objective is to say given an X is it is in C 1 or is not in C 1. Similarly, is it in C 2 or not in C 2. So, I learn key a number of 2 class classifiers, each classiffier is learnig to distinguish between C i or not-C i. This apporach is often called one verses rest and is a very standard approach in learning multi class case. Especially, in when you one one does all in a clasifier very often one goes for on verses rest apporach.

(Refer Slide Time: 46:02)



Another way of doing it is inster learning K, I learn K C 2 number of 2 class clasifiers. Now, for every distincit pair i and j I am learning 1 classifier to distinguish here C i from C j. The main of course, I am learning more number of clasifiers here. Why should I learn more number of clasifiers, why cant i use the previous method where I have to learn only K classes? One problem with this C i verses not-C i learning could be that for this problem, the effective training set could be very (( )) . Let us say I have 100 classes and I have let us say 50 examples of each class making a total of 5000 examples.

Now, when I want to do C 1 verses not-C 1, I have only 50 examples of C 1 whereas, 4950 examples of not-C 1. fine. Then, if one of the classes has much less much less represent the training set then the another 2 class problem. Predominantely, all traning data belongs to to 1 class, that is very diffucult to learn the classifier. For example, in this case if i have 50 of 1 class and 5000 minus 50 of the other class. Then, a classifier always says not- C 1 has only a error rate of 50 by 5000 on the traning data which is very small right. So, stupid classifiers can have very low error on the training set and that always makes it difficult. Often, what one does is when 1 class as too many samples then, you resample from there. So, out of this 4050 I may be I take some 50 or 100 randomly selected samples and use that to do C i verses not-C i. But, when I do that may be that does not represent all the other not C 1 not-C 1 clases.

So, if the number of classes is very large then, one versese rest can be difficult. Because the training data squte will be very squte for the resulting 2 class problems. If I have only 3 or 4 classes then it may not be so bad. But, if i have hundred or 200 class 100 classes or 50 classes then learning C i verses not-C i could be a squte data set problem and I have to properly balance the data set. It is in those casses that learning C i verses verses C j classifiers could be because if essentially when I have got K class problem and I have training data, one accept that all class all clases are represented about equal measure. So, for a 2 class problem C i verses C j, I will have roughly equal number of example for both clases there is no squte traning training say distrubution.

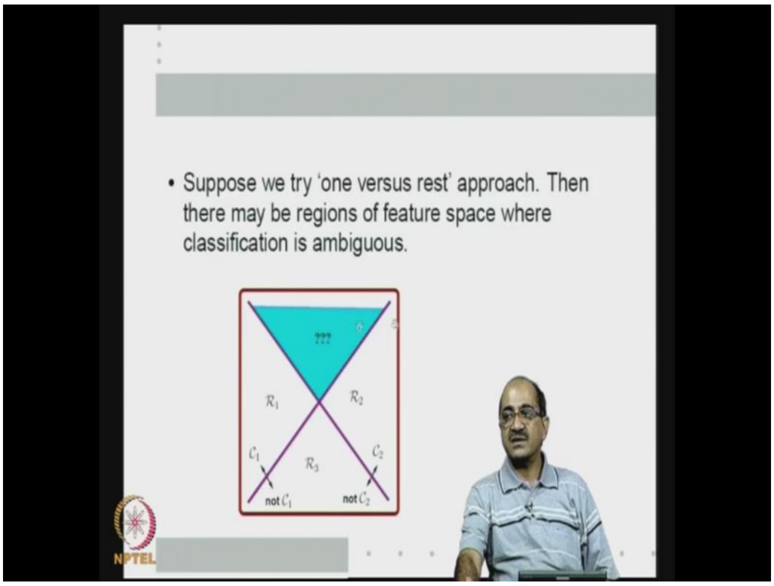
So, this is a little more easier, is more stable 2 class clasifier learning and that the reason even though I am learning more number of classifiers, I may want to do this. So, these are essentially 2 ways of using, there are others but, these are the 2 main ways of using an algorithm that can learn a 2 class classifier to classify K classes. But, however there are issues with both of these. Right. I will we will We will see look at some problem, there is not to say that there are not followed.

But, there are pit falls with both the apporaches in as much as any set of such classifiers, this set of K classifiers or this set K into K minus 1 by 2 clasifiers do not make a perfect, a well defined classifiers. Perfect is wrong word to use because we are not saying it has to have 0 error or anything like that. But, atleast it should be a well defined classifier. What do you mean by this? A classifier is a black box to which if I give an X. There is no doubt in the classifiers mind as to what class X should be goin because this is an algorithm, it has to be an algorithm. Whether, it is right or wrong is a different issue but, something to be call a classifier atleast structurelly, given an X it should be assigned to one unique class.

But, if I learn this many 2 class clasifiers or this many 2 clas clasifier, together they do not really define a very a proper classifier in this sense. Thus, what will see next that neither this apporaches are realy satisfactory generalizing the linear discriminant function to multiple classes. So, before I will show you some examples of why this happens and tell you how we can solve this.

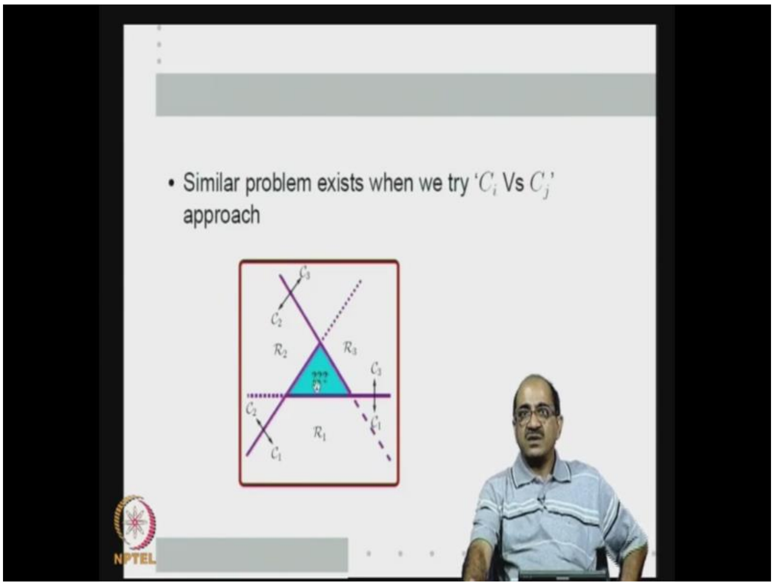
But, before I go there I spent a lot of time telling you about this 2 ways of dealing with multi class problems because inspite of the problems I am going to tell you, sometimes people use this apporach for solving multi class problem. But, we should still know what the pit falls in the apporach are.

(Refer Slide Time: 51:18)



So, let us first consider one verses rest apporach. The problem is that there are regions of feature space where, the classifiaction is ambiguous. Right. Let us say, this is the linear classifier that separates class C 1 on one side and not C 1 on the other side. Let us say this is a linear classifier that separates class C 2 one side and not C 2 on other side. This is the region 3 where, it is class 3 as it should be not C 1, not C 2. But, there will be region like this where, this classifier will say C 1 and that classifier will say C 2, may be my third classifier will may say not C 3. That is all right. But, what should I do here; because if I take any X here one classifier say C 1, other classifier say C 2, my third classifier say not C 3. So, that is okay, not C 3 is fine. But, is it C 1 or C 2 or no way of knowing. So, this does not even define a proper function that maps every feature vector to a class. Right.

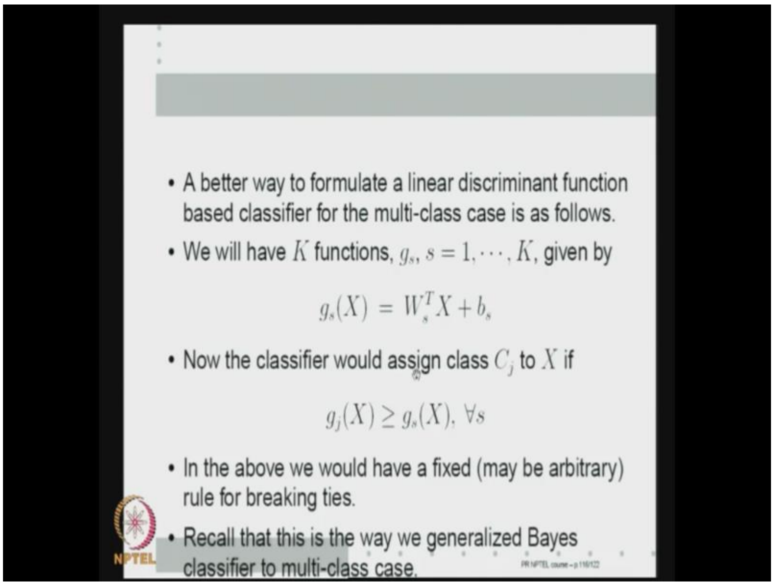
(Refer Slide Time: 52:21)



The same this is to even, if you use C i verses C j apporach. Once again here is an example. So, in 3 class case. So, this is a C 1 verses C 2 line, Right. That is, a C 1 verses C 3 line, this is C 2 versesc C 3 line. In the centre triangle where, I put the question marks if I take any x; one classifier will say C 1, another clasifier will say C 2, the third classifier will say C 3. A perfect disagrement between the 3 classifiers. What should i assign it as the class label?

So, there is always this issue when I have that many 2 class classifiers. If the responses given them are consisted then, given any x, I can decide on the class label for that x. But, there will always be the regions in the feature space like this. Where, the responses are in consistence and one does not have any simple solution of how to put together all those responses into a perfect or into a proper class labels. This is the problem of generalizing 2 multiple classes.

(Refer Slide Time: 53:27)

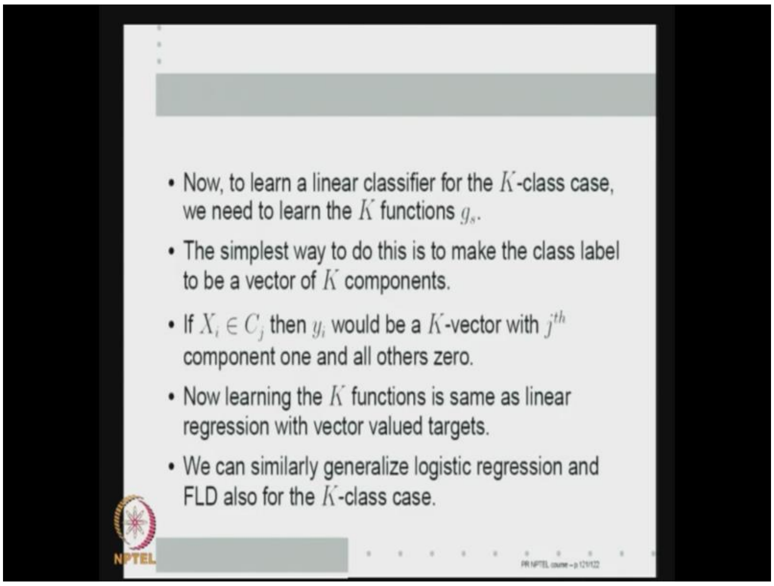


Of course, there is a better way of formulating this. We have seen this when we considered multi class bayes clasification for minimizing this. The same thing is to do in the linear discriminant functions. Essentially, the way to generalize 2 multiple classes is will have K functions, I represent them as g subscript s, s going from 1 to k and each function is a standard linear function. That is a fine function rather, j s of X is W s transpose X plus b s. Right.

Now, I define a way of making this into a classifier for all such functions. Now, once I have all these functions, if i if you give me an X, I will assign X to class C j, if g j of X is greater than g s of X for all s. So, I calculate the value of X for each of this functions and which every function has the maximum value of X at that point X, I will assign X to that class. of course, this is not at complete because there will be ties. Right. So, there might be say g 1 X and g 2 X might have the same value and that is bigger than the value of g j X for all other j. Now, should I put X in class 1 or class 2. But, once we come this for we can have a very simple and arbitrary rule for breaking ties. For example, I can say that if the maximum is attained at more than 1 j then, I will put it to the j to the least j.

So, if j 1 X and j 2 X are the 2 maximum of function values for x then, I will put it in one. Right. So, I can have any arbitrary but fixed rule for breaking ties so that it becomes a proper function. Given any X it unquely defines a class label. So, essentially this is how one would like to generalize linear discrminant functions to multiple classes. Recall that, this is the we have generalize the bayes classifiers.

(Refer Slide Time: 55:28)



Now, to learn a linear classifier for K-class case, we need to learn all the K functions g s. The best way to do this is to use or vector valued idea of regression functions. So, what we do is, we make the class label to be a vector of K components. What is that mean? If X i belongs to C j then, in the training data the class label y i instead, of simply taking value j, it actually will be a K vector with the j'th component 1 and all others 0.

Now, y i will be K vector now. And to denote j'th class j, I put 1 in the j'th component and 0 will be. So, each y i will be a unique one of the co-ordinate vectors. So, we seen similar coding when we consider this K random variable as M l estimation. So, the same codind we can use to make the class labels K vectors. Now, learning K functions is same as a linear regresion with vector valued targets. I am given X i, y i; y i are vectors. I want to learn a linear function. So, I can learn K functions using linear regression with vector value targets. So, this is one standard way in which linear (( )) analysis can be extended to multiple classes. There are of course some slighty different ways and this by itself does not tell you how y can generalize logistic regresion or fisher linear discriminant because they there the structure of landing is slightly different there.

But, similar ideas can be used for generalizing logistic regresion fisher linear discriminant also. So, with the next class we will just briefly look at how to generalize logistic regression to K classes. And then, we will step back we since some algorithms now for learning a classifiers. We will just take a look at all the algorithms and then, take a step back and ask, is there some therotical ways in which we can say whether one classifier is good or not good; how do we decide whether a classifier is optimal or good; is there a a statistical way of defining the goal of learning in learning a clasifier. So, we will do a better statistical learning theory and then, we returned to learning non-linear classifiers. Now, that we finished all linear clasifiers, we will step back to do some statical learning theory and then, come back and do the non-linear classifiers.

Thank you.