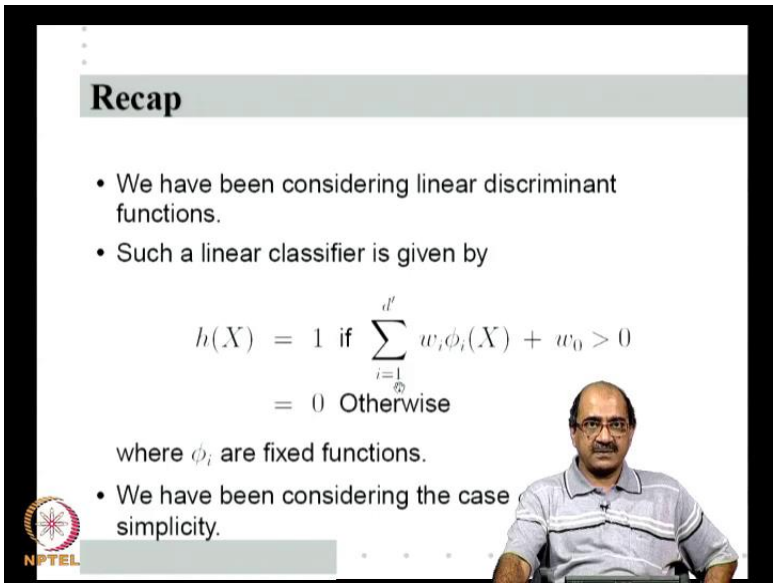
Pattern Recognition
Prof. P. S. Sastry
Department of Electronics and Communication Engineering
Indian Institute of Science, Bangalore

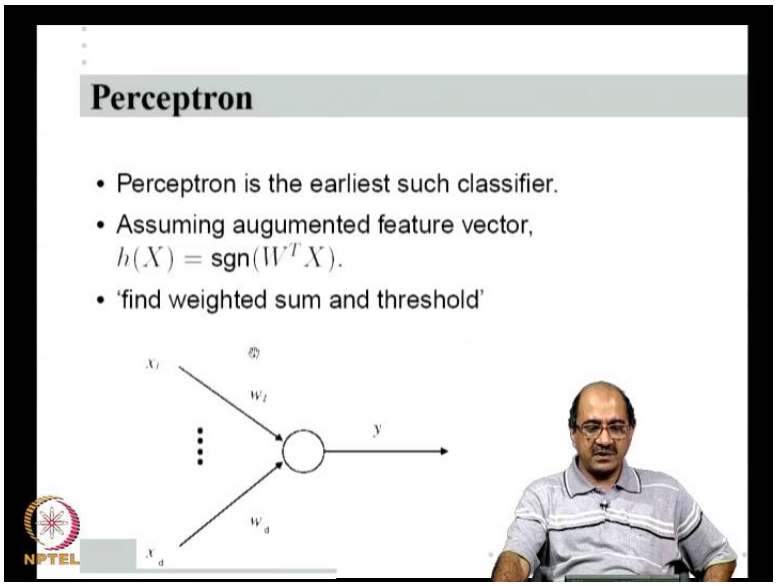
Lecture - 15
Linear Least Squares Regression; LMS Algorithm

(Refer Slide Time: 00:40)



Hello and welcome to the next lecture on pattern recognition.We will, we have been discussinglinear discriminant functions last class. We havedone withlooking at Bayes classifier. We are looking at other ways of learning classifiers and regressors. So, we started with thelinear discriminant functions.Basically, a classifier using linear discriminant functions hasthe following structure.The classifier will say one, if sum of w i phi iX plus w naught is greater than 0 for some prefixed functions phi i, right? This is what we have seen last time, otherwise 0.As I said in the beginning of last lecture, for now we are only looking attwo classes. At the end ofmay be next lecture or the lecture after that,we will consider how to generalize thisto multi class problems?

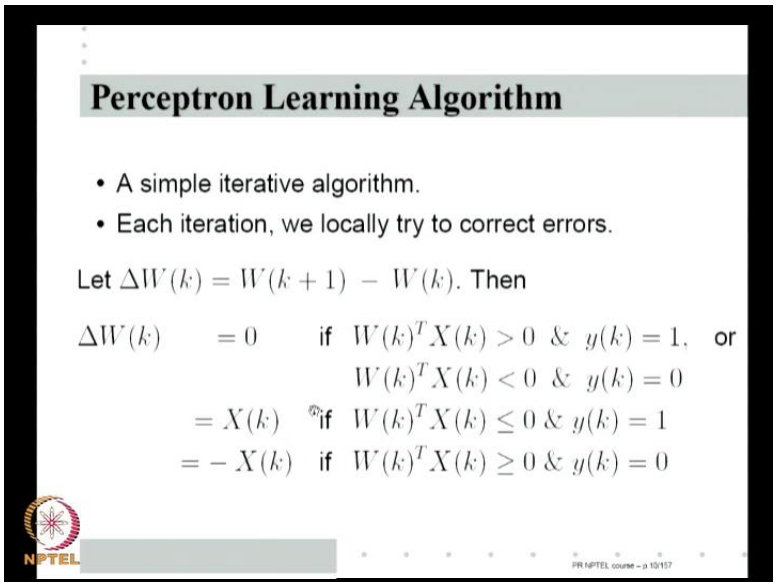
(Refer Slide Time: 01:38)



So, this is the structure of the classifier thatwe are considering.We have considered last class.Of course, we have takenphi iX to be X i for simplicity. So, this simply become wixi and d prime becomes t which thedimension of x. As we said whether we use fixed functions phi i or whether we use w i x i, the thethe techniques involved in learning them are the same. So, all of them we will look at as linear classifiers. Now, the linear classifiers algorithm that we considered last timein the last lecture is perceptron, which are the earliest such classifiers.

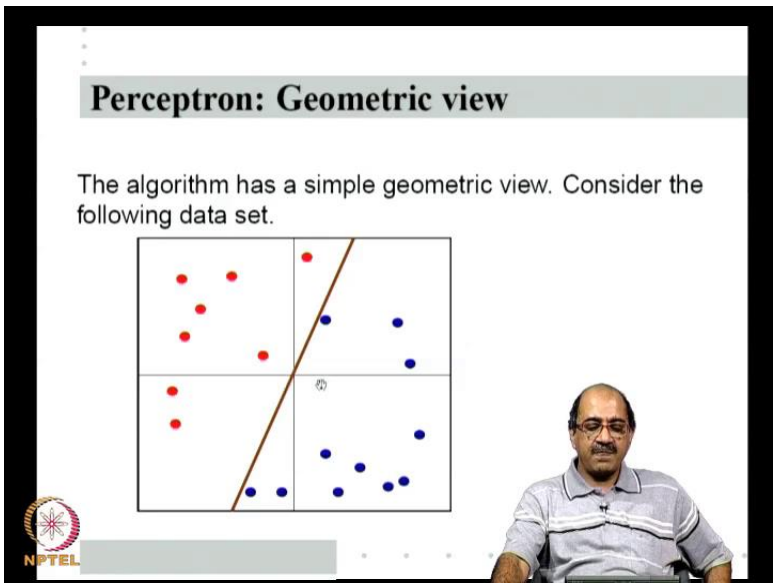
Essentiallyas we have seen by augmenting thefeature vector, we can write any linear classifier as W transpose X.Now, both w and x will bed plus one dimensional vector because of augmentation. So, this is nothing but you find a weighted sum.w transpose x is simply weighted sum of the feature values.You sum w i x i. Then,this this sign function simply threshold set.So, we seen that is that kind of a structure wherethere are n input signals which are the features.Each one has it is associated weight.You find this unit finds the weighted sum and then thresholds and that is the output. Okay?

(Refer Slide Time: 02:32)



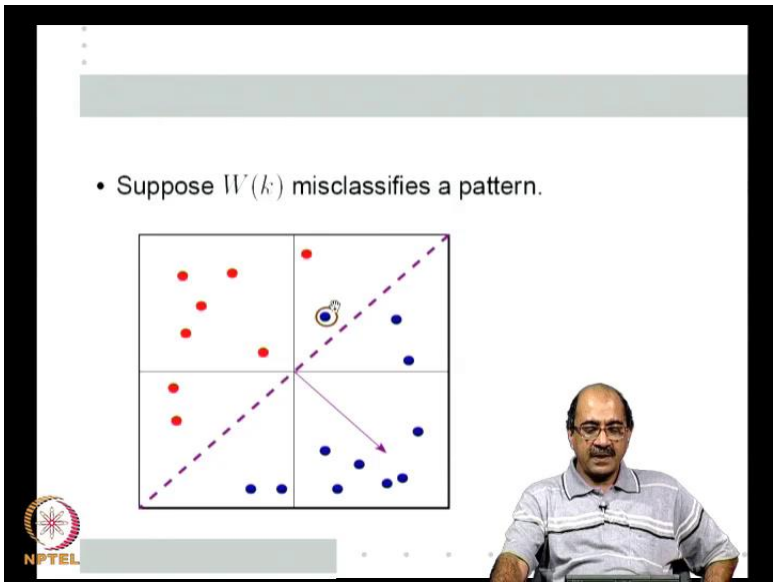
Such a device is called perceptron.Perceptron learning algorithm, as we see in last class is a very simple and iterative algorithm.It tries to learn the weights iteratively.At each iteration, we take onefeature vector trained. Ask whether the current weight vector correctly classifies it.If it does not, then we locally correct the errors, right?To recall, this was the algorithm.So,if I take delta W to be W kplus 1 minus W k, this is the correction that is too made true. W, it is 0. If the current weight vector W k correctly classifies the next training part and that is X k; X k is the training pattern picked up at iteration k.

(Refer Slide Time: 03:32)



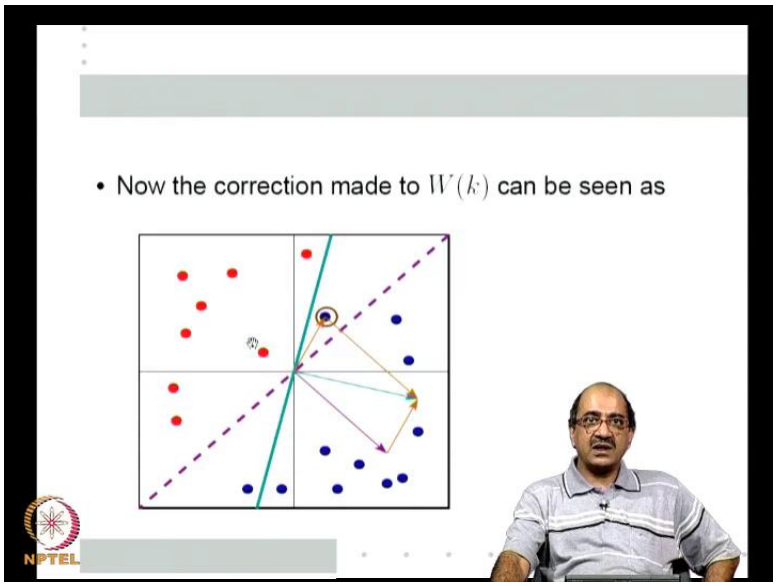
So, if Wk transpose,X k is greater than 0 and y k is 1, that is correct or and y k is equals to 0;W transpose X k is less than 0.Then, I do no correction.In the other two cases of errors,I either add or subtractX k.As we have seen, this is a very nicemethod as to recall. It has a very simplegeometric interpretation.

(Refer Slide Time: 03:44)



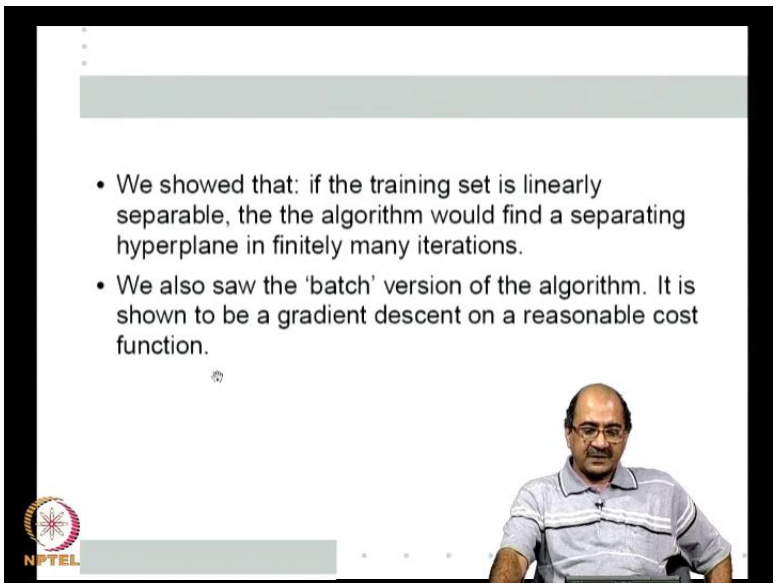
So, here isa simple linear least separable class that isthat hyperplane could separate them.But, let us suppose we started with some hyperplane, which makes another one point is on the wrong side.This is the W vector.So, the all the blues are the, so to say positive class. Reds are negative class in the sense for all the blues W transpose X is positive. For all the reds, W transpose X is negative. W transpose X is simple, thedot product. As you can see in the augumentedfeature space, all the hyperplanes pass through the origin; not the hyperplane equation is W transpose X is equal to 0.

(Refer Slide Time: 04:19)



So, now we have this wrongly classified thing, which is on the wrong side of the hyper plane. As we seenwhat it means is that,I take W k,add X k to it. So, that becomes a new normal vector, which is essentially same as moving, therotating the hyperplane in the right directionwhether we adding or subtracting; essentially a matter of rotating the hyperplane in one direction or the other.Of course, when we do a local correction like thatin the previous hyper plane, this point is correctly classified.Butnow, after I took care of correction for this point, now this point becomes wrongly classified.

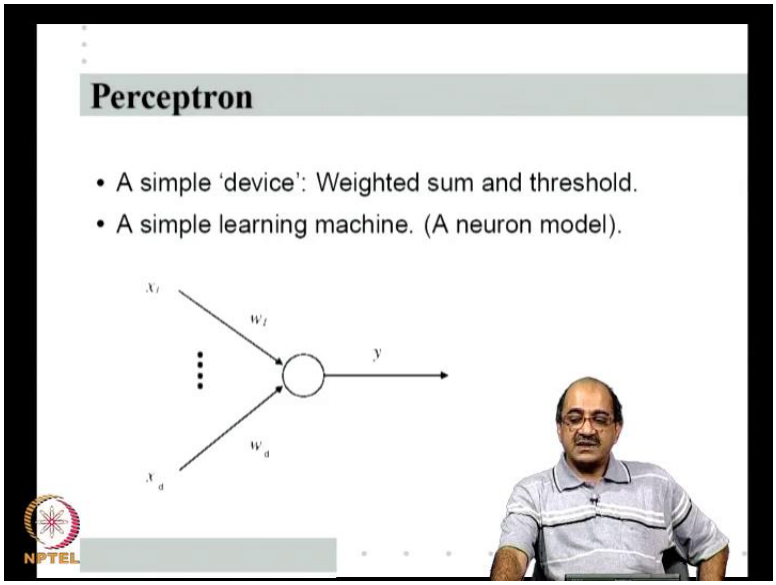
(Refer Slide Time: 05:02)



So, correction now could have made something else that was correct earlier to becomenon correct.But, not withstanding thatweknow thatthis algorithm converges.We showed thatif the training set is linearly separable, these series of corrections will stop at some finitetime.We will always find a separating hyperplane after finitely many iterations.So, this algorithm is of course incremental in the sense I look at only one feature vector at a time.

So, in principle if I am just getting a stream of feature vectors,I can run the algorithm with that withoutneeding to store all the feature vectors in the main memory. We also looked at aa batch version where using the current W k, we classify each one of the training vectors. Then, looking at the correctness otherwise of all of them together we make all the corrections together to the W k.We shown that the batch version is essentially a gradient descent and a reasonable cost function and hence that also converges.

(Refer Slide Time: 05:58)



So, to sum up, perceptronis a simple device, essentially weighted sum and threshold. Itis also a simple learning machine, one of the first learning machines, one of the first learning algorithms of pattern recognition. It is also can be thought of as a neuron model. As I told you,aa simple model of a neuron is that it has many input portswhich are calleddendrites, on which inputs come at places called synopsis. Each synop can have a weight in the sense how much that synop have fixed the on the surface of the neuron.

So,I can think of a neuron,a simple neuronfor a mathematical model to be essentially doing a weighted sum of all the inputs and then thresholding it. Right? In that sense, is a simple neuron model and the weights can be adopted.As a point, we have to last classbecause the the adaptation is simply based on the kind of error that is known at this point, you add W to X. That means for each component, you add the corresponding component.

So, to update this weight,all theinformation needed is available on this link it iself because it just needs to knowthe value of Xi there. So, it is a very nice local algorithm.Idid not really explain it in detail last class. But,perceptron which waproposed by Rosenblattwaswaspropose not just as a abstract pattern classification algorithm. But, this is aproposed as one possible way in which our brain can learn.The idea is that, by then people knew that there are neurons like this.

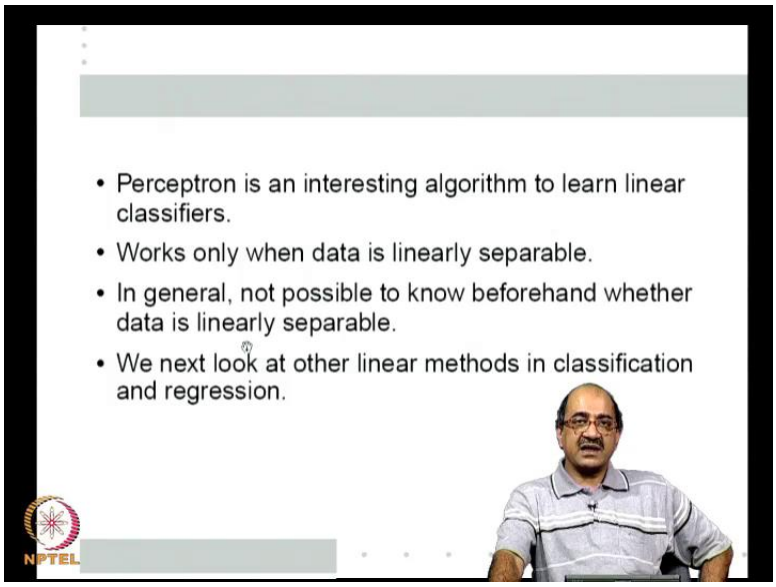
The outputs of other neurons will come and synapse at the inputs of some other neuron.That is how they interact. Essentially, all the currents passing through all the synapses will increase the potential on the surface of the neuron. Oncethe potential goes below someabove some value, it fights an action potential which will be then input to other neurons and so on.So, given this,any change in the in the information knowledge problem solving ability of the brain can come about only by changing synaptic weights.

So, everybody is looking at algorithms of how synaptic weights can be changed, so that non-trivialprocessing can be done. So,Rosenblatt proposed this as a model of how we may learn visual categorization.So,I am shown some images and say these are dogs.I am shown some other objects, say these are cats.We can think of these features as measuring some quantities on the image.As I as we seen already, if Xi is replaced by phi i of X,nothing changes in the algorithm. So,I can think of each of these as phi i of the pattern that is put in front of the neuron which could be in front of your retina, for example.

So, given some pattern,there could be some measurements that we are bound with. So, these phi’s are fix.So meaning, we have born with some kind offeature detection we can do on the image that is in front of us. Given thatthese things are fixed and somebody wants to teach us,let us say teach us to distinguish between maruti cars and let us sayambassador cars. Obviously,it is too much to expect that we have born with an ability to distinguish them.

So, using some generic measurements that we have, we somehow learn.Given enough supervise examplesbeing pointed out that this is a maruti car and that isaambassador car. Even though we may not be able tovery preciselydescribe the difference in shapes between maruti cars and ambassador cars,we we learn to distinguish between the shapes and can do theclassification correctly. This could be one example onemethod by which such visual categorizations could have been learnt.

(Refer Slide Time: 10:44)



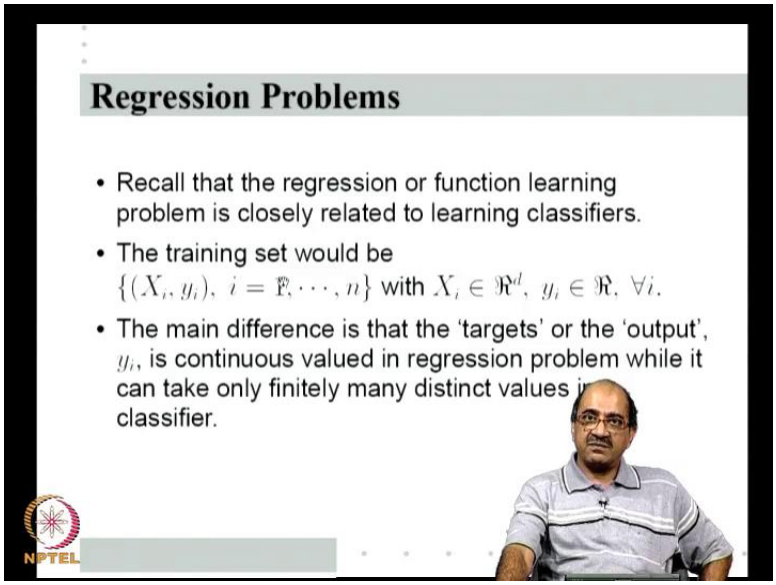
It is in that sense that perceptron was originally proposed. For it is time,it is a,it is a very very interesting algorithm witha very interesting convergence proof. As a matter of fact, it was actually built using electronic hardware those days. These are pre ic days using mostly op-amps.A matter of fact, a model of the perceptron iseven kept in the Smithsonian museum.So, it was in the in the development of a in pattern recognition.Perceptron is a is historically very important step.

Anyway as you seen, it is an interesting and simple learning machine.It it can learn linear classifiers. It is main drawback is that it works only when the data is linearly separable.Now, for a particular given task say,maruti cars verses ambassador cars. I do not even know what features, detectors I have.How can I at all say whether the data is linearly separable or not?In general, even if you are given some vectors in a d-dimensional space,it is not possible to easily deduce whether or not the data is nearly

separable. That brings andas we as as we have seen in last class, if the training data is not linearly separable, then the perceptron algorithm can potentially go into an infinite loop.

Even though we have a convergence proof of finite iterations, we do not know how many iterations it actually takes.We do not know whetheryou know we are going towards convergence or we are going to a infinite loop.So, that is one real problem with the perceptron algorithm.So, we next look at other methods for learning linear classification.When you are looking at this, we look at two class classificationalong with the regressionproblem. If you remember right in the beginning, we looked at both classification as well as function learning problems and said both of them are quite similar the function leaning problem is what the regression problem is about.

(Refer Slide Time: 12:31)



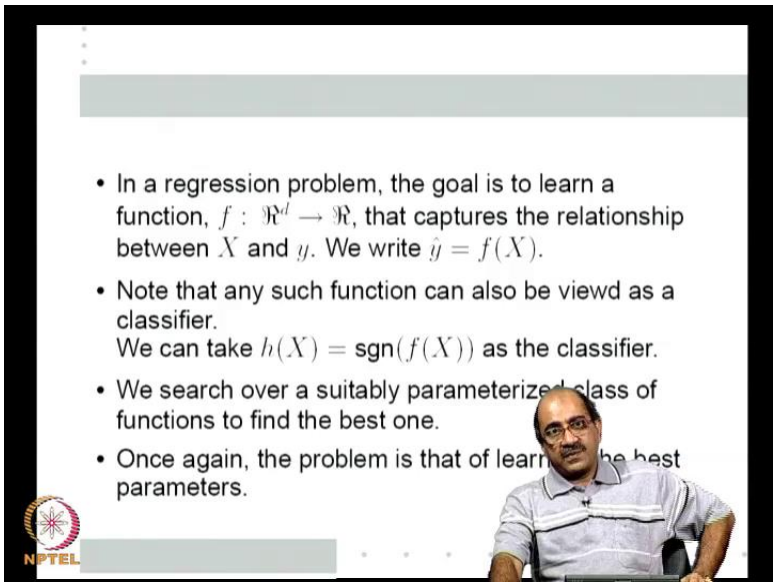
So,the the next techniques we look at learning linear classifies are also techniques for learninglinear modelsof a functions are linear regression models.So, we look atlook at them together.So, to briefly recall, the regression or function learning problem is a closely related problem; a problem closely related to learning classifiers.Essentially, when I am learning a function, when I am ask to learn a function; from examples, the training data is X i y i. Again,i going to 1 to n. Xi is still a d vector, but y i can now be any real number,right?

So, the main difference between a classification and a functional learning problem is that, the targetswe can call, or the outputs;namely y ican be continuous value in a

function learning problem whereas they take only finitely many values in a classification problem.In a two class classification problem, they take only two values. In a multiclass classification problem, may be they will take k distinguishable values.But in general, in a classification problem, because it is a categorization problem, the targets or the outputs; they take onlyfinitely many values whereas, in regression problem the output is continuous.

That is the only difference between aa learning a function or instead of learning a classifier otherwise both of them are functions on Rd.Only thing is one could be a general real valued function. The other is a boundary valued function; a function that takes only finitely many values. This being so, it is no surprise that similar technique should work for both.

(Refer Slide Time: 13:58)

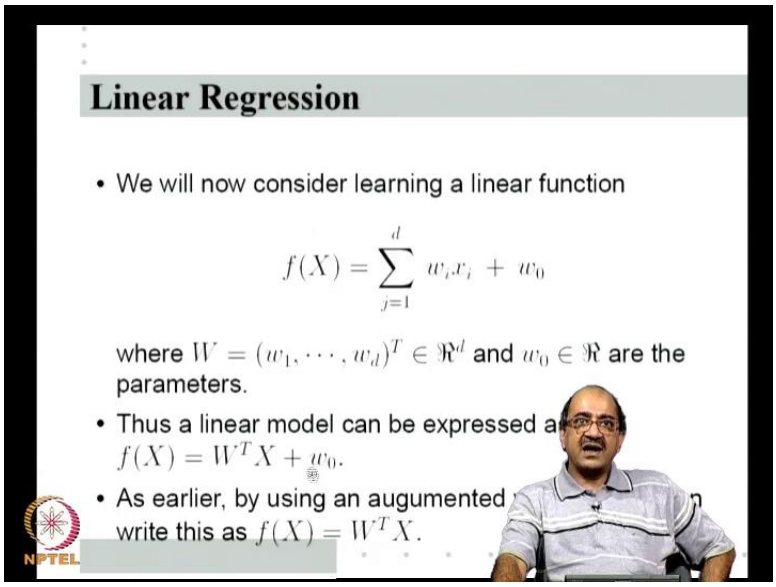


So, the next technique, we are going to look at,can we can learn either classifiers or functionswith the same algorithm.In a regression problem what is the goal? The goal is to learn a function from Rd to R because of my target values are in R. So, the idea is that, we arecapturing the relationship between x and y using a function.So, we can write y hat some function f of X.The function f is what you want to learn. So, given X i y I, wewe abstract the functional relationship between X and y, so that now you give me a new X. I can tell you what is y, namely, y hat is equal to f of X.

That is how I am going to make my prediction of the targets for any newX.In the classifier, this will be a binding function, but otherwise it can be a general;a function that takesany arbitrary real value.So, any such function can also be viewed as a classifier. For example, I can take sgnof fX as a classifier, ifI am any two class problem. So, learning a functionRd to Rcan also be viewed as a classification problem,two class classification problems simply bytaking the classifier to be sgnof fX.

So, essentially the idea now we use to learn this function f given the data.So, if you have to learn a function,what we do is we we start with a family of parameter as a functions. Then, wefind the best function among them which boils down to once againlearning the best set of parameters from a parameter space base.Given, a parameter class of functions and the training data using which we want to learn to get through the best function.

(Refer Slide Time: 15:41)



Now, in in this class,in this lectureand the next few lectures,we will be mostly be consideringlinear classifiers and linear regressors.Hence,we are looking at functions of the following general form. They are essentially called affine functions because of the constant. As you know, if there is a constant the function is not linear, butwe already seen it earlier in the perceptron case that we can easily make it linear that is W transpose Xby simply augmenting.

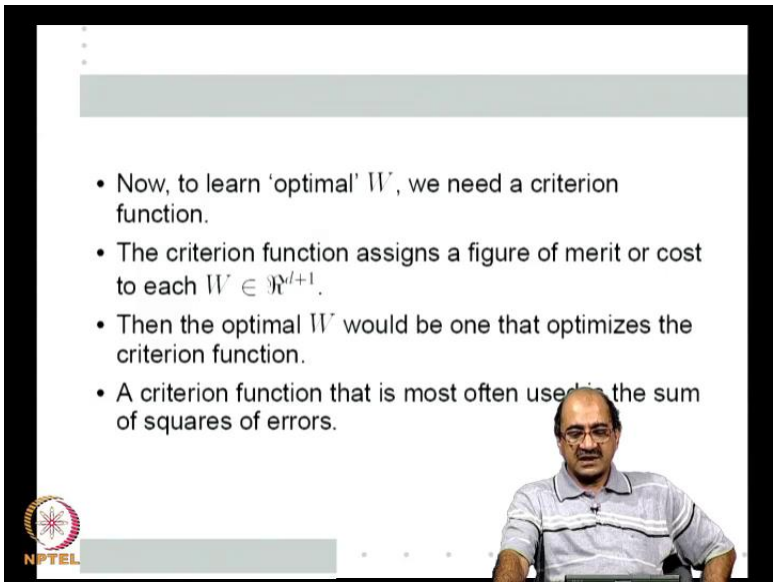
So, linear and affine functions are same for us, so looking atlearning a general linear function of this formfX is J. It is equal to 1 to d WiXi plus W naught where W1 to W d.

We call that vector W, is an R d n. w naught is an r. These are the parameters of the function. That is the linear model, that is, we are representing thethe function relationship between y and x through a linear model.Namely, W transpose X plus w naught.

Just like in the classifier case, I can simply augment x with an extra feature which is always 1, extra component which is always 1. Assume that w 0 is substituent to w. So, the w now becomes d plus n dimensional vector. Then, I can write that asf X equals to w transpose X.Soif you remember, last time when we did this, we for a while we used x tilde for the augumentedX and W tilde for the augumentedW.Butnow that we know this trick, we either write it as W transpose X plus w naught or W transpose Xwithout changing the notation.

Here also we use the same w. When I write like this, we mean X is d plus one dimensional w is d plus one dimensional.When we write like this, x is d dimensional and w is d dimensional. Right? As long as the thethe difference is clear from context,we canwe can easilytake this. Hence,you know, without without changing our notation; when we want we use augumentedvector notation, when we want we use the full expanded notation.

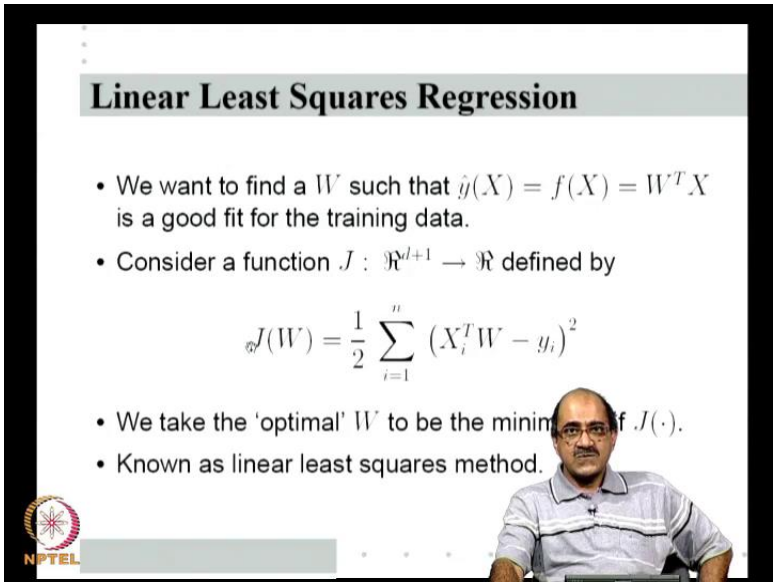
(Refer Slide Time: 18:04)



So, now our our task is tolearn this W vector given all the training data.So, to learn the optimal W we need a criterion function.What does a criterion function do? We want

some function that assignseither a figure of merit or a costto each W belonging to R d plus 1 d plus one dimensional space. So, essentially the function is telling how good W is, we want to maximize the function.Is the function telling me how bad W is, we want to minimize the function.Any case is called a criterion function.We use the criterion function to define what is our optimal W. So, the optimal W would be that value of W which will optimize the criterion function.A criterion function that is most often used is the so called sum of squares of errors.

(Refer Slide Time: 18:57)



So, we will look at this criterion function next.So, basically before we write the criterion function let us ask what is ourour objective. See, the criterion function should be such that our objective is well captured.We want to approximate the relationship between X and y through a function f.So, we are writing y hat of X is f of X is W transpose X.So, we want W transpose X to be a good guess at y corresponding to X. So, in particular, we want this to be a good fit for the training data that is, if I put X to be any training data,XiW transpose Xi should be close to the correspondingy, which is y i.

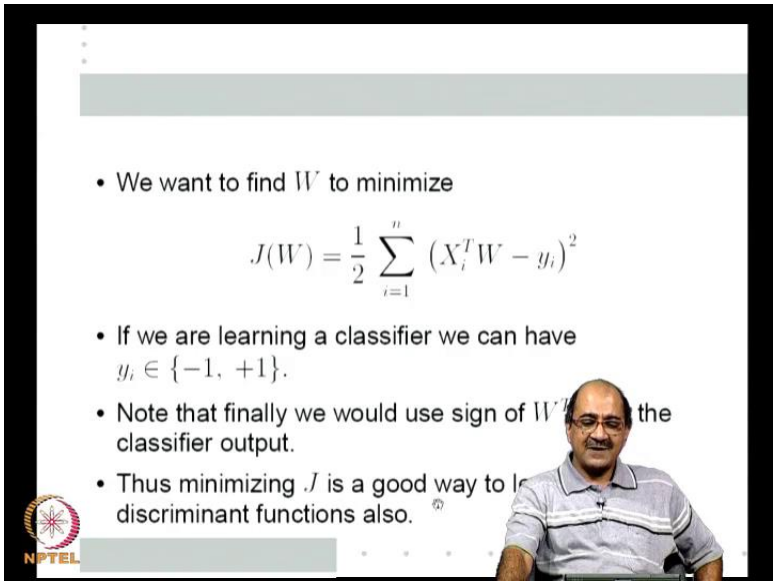
So, we can look at a criterion function like this.This is the function defined from R d 1 plus to R so it assigns a cost to every vector W. The cost is if W is my finalsolution for the function with with that function w, with that w if you give me X i, I would have predicted X i transpose Wwhere as the true value is y i. So,Xi transpose W minus y i is the error.I square the errors because some errors may be positive some amount of errors

may be negative.So, if I simply add even though I make lot of errors, I may artificially think that I have no error.

So, we square out the errors and add them.So, this JW is always positive. It will be 0 if i reach i y i exactly matches Xi transpose W, otherwise it will not be 0.If JW is low then it means that inyou know,most of the time using W,I am able to predict the corresponding y i for a given Xi. If you give me Xi, my prediction is Xi transpose W.The true value is y i.I am taking the sum of squares of errors. If sum of squares of errors is small, that means my prediction is not true for half.

So, it is good to look atoptimal W as the minimizer of J.If Iif I can find a W thatfinds a global minimum of J,then that is the best error I can get.This is often called the linear least squares method or sometimes mean squaremethod. So, we are taking thesum of squaresof errorsand asking which W will minimize. This is called the linear least squares method. So, we want aa least square solution is is that value W which minimizesthis function. This function isvery reasonable because essentially each W is being rated on how well it predicts the correct target value for each of the training samples.

(Refer Slide Time: 21:39)



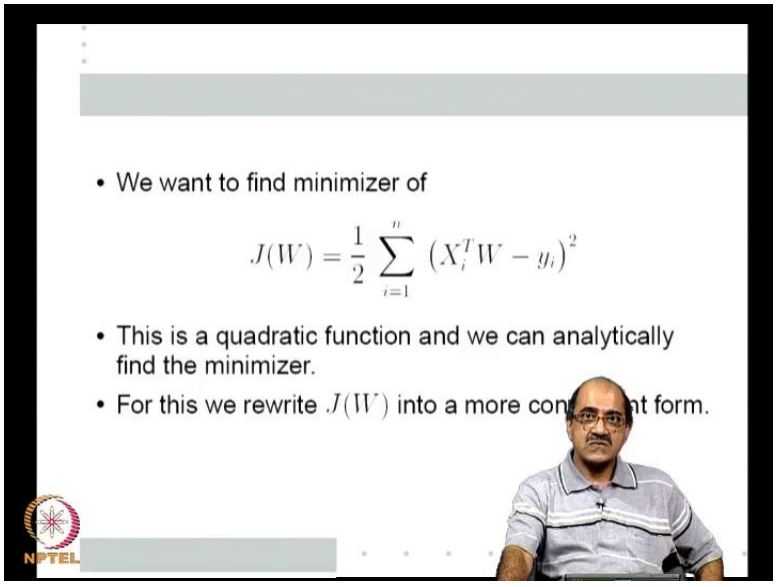
So, our objective is to find a W to minimize this.As we seen for a function learning problem, this is a this is a this is a goodcriterion to have because essentially I am learning the function f X is equal to W transpose X. So, if you give me X i as the input, I would have predicted Xitranspose W and the true value is y i.So, if thatW minimizes this then,

most of the time I am I am getting my predictions right.This can also be used for classifier.

Suppose, we want to learn classifiers. Then,let us say y i are are in the range minus 1 plus 1 instead of 01.Let us say, the two class error represent as minus 1 plus 1,that meansif I once again put it in this, I will get low error iffor all XI, which are in class plus 1; W transpose X i is closer to plus 1.For all X i that are in class minus 1, W transpose Xi is in classis closer to minus 1, that is when I will get lower value of J. So, because we would be using sgn of W transpose X as ourfinal classifier output.

So, if W transpose X is positive,we will put in plus 1 class.W transpose X is negative, we will put in minus 1 class so thatessentially, if I am learning a W, so the W transpose Xi is close to plus 1 when our Xi is in plus 1 class.W transpose Xi is close to minus 1 when ourXi is in minus 1 class.Then, such a W will do well as a classifier too. So, I can simply take y i is to be plus 1 minus 1. Find a W that minimizes this.With that W, I can implement a classifier is simple, say if Xi is transposes W is positive then,I will put in plus 1 class.X i transpose W is negative, I put in minus 1 class.

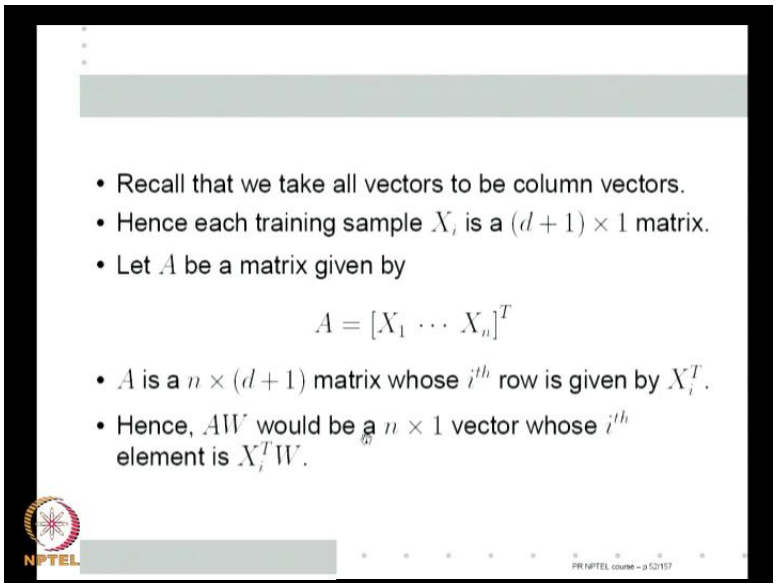
(Refer Slide Time: 24:02)



So, minimizing the same criterion function,I can either learn a linear regression function or a linear classifier.Thus, minimizing Jis a good way to learn linear discriminant functions.Right? So, minimizing J is a good way to learn regression functions as well as a good way to learn linear discriminant functions.Now,so let us let us look at

thecomputational problem. Now, I am given training data Xi y i.I have this function J of W defined for every W inR d plus 1.So,I want to set your R d plus 1 to find a minimizer of this function.

(Refer Slide Time: 25:03)



Now, this is a quadratic function, right?So, because this is a quadratic function, it is no surprise that we can analytically find the minimizer. So, what we will do next is to derive the minimizer by using calculus.To do that, we will have to first put this expression into a slightly different formwhere it is easier to differentiate it andfind the minimizer.So, what we are going to do is we going to rewrite J of W into a more convenient form.Recall that weto put it in the in this form,we need to do some vector algebra.We wewe are going to representthis expressionin in in in a vector matrix notation.

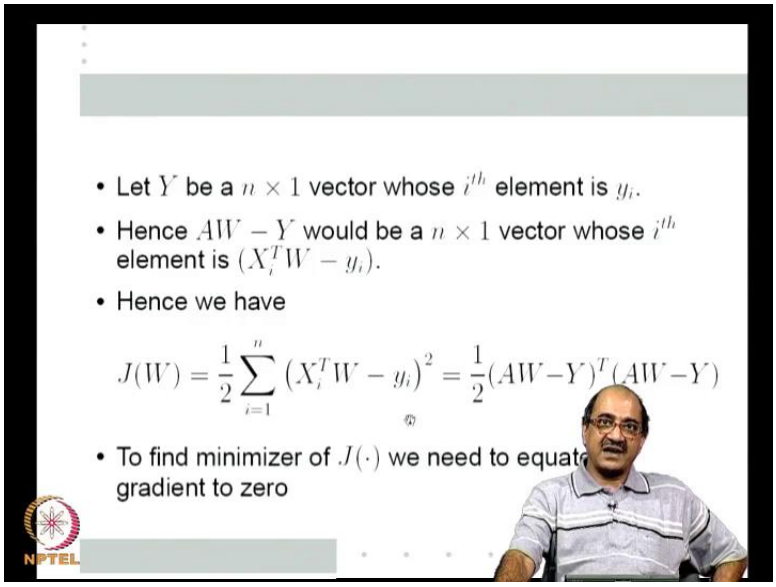
So, so fareven though all our feature vectors are vectors, we never particularly bother about that representation. Butright in the beginning, I told you that in this course, all vectors are always taken to be column vectors.So, we did notaffect us much whether column vectors or row vectors. But now that we need a specific representation;let us recall that all vectors are column vectors. What does that mean?Each training sample Xi is a column vector.A column vector means it is a one column matrix.

So, because Xi has d plus 1 dimensions, each Xi is essentially aa d plus 1 by one matrix.Now, let A be a matrix given by,soyou put each of the X 1 to X n as columns of a matrix and take it is transpose.That means the first row of A will be aX 1 transpose. It

has to be X 1 transpose because X 1 is a column vector.So, the first row of A will be the first training sample arranged as the row vector.Secondrow of A will be thethe secondtraining data vector and so on. So see, each X, each Xi is d plus 1 dimensional.

So, without the transpose, this will be d plus 1 by n matrix. With the transpose, it becomes n by d plus 1 matrix and whose ith row is X i transpose.You have to write it as Xi transpose because you have to make it a row vector. By bybydefinition,Xi is aby definition x i iscolumn vector.So, the ith row of the matrixis given by X i transpose.Now, if I do AW where W isA is a n by d plus 1 matrix.W is a d plus 1 vector. So,AW will be n n by 1 vector.A's rows are represented as X1 transpose,X2 transpose, so on.So, in this n vector, when I am, when I take this matrix vector multiplication, the ith element will become Xi transpose W.So,AW will be a n vector whose ith component is Xi transpose W.

(Refer Slide Time: 27:50)



Now, let the capital Y denotes also a n vector whose ith element is y i. See, I have n training samples XiYi.So,I take all thetargets y 1 y 2 y n.Arrange them as a vector. Then, call that y.Now, if i make a new n by 1 vector; see,AWA is An by d plus 1 matrix.W is a d plus 1 by 1 vector.So,AW is A n by 1 vector.y is also an n by 1 vector.

So,AW minus y will be n by 1 vector whose ith element will be what?The ith element here is Xi transpose W.Here is y i.So,ith element of a W minus y is Xi transpose W

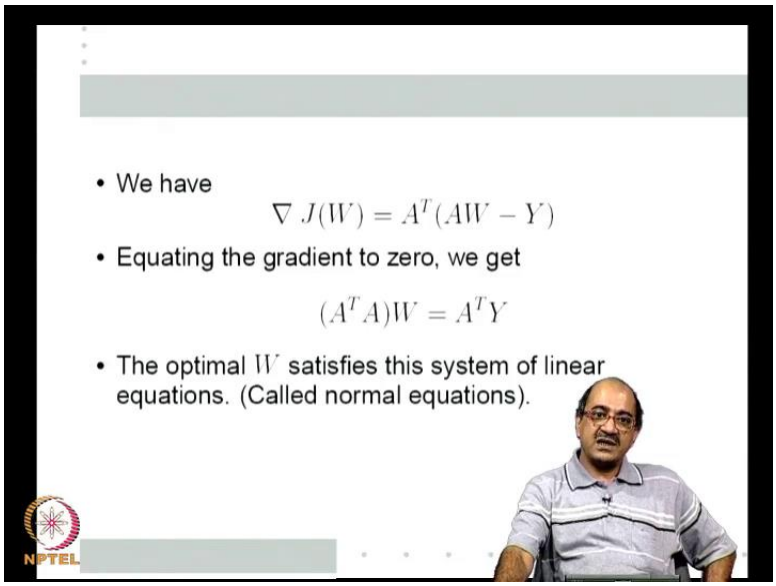
minus y i.Now, we are ready to rewrite the functionJ. J ishalf ofXiXi transpose W minus y i whole square.

Now,I have a vector here,AW minus y whose i-th component is exactly this; Xi transpose W minus y i whole square. So, this sum is actually, you take each component of this vector,square it and submit.That is nothing, butthe norm square of this vector. So, the norm of that vector is simply AW minus y transpose W minus y. So, I can write JW as half of AW minus y transpose, AW minus y. Just a small point which I should have mentioned in beginning; the only reason we put a half in this AW, it isit is it ismore out oftostick to convention.

When you differentiate with respective to W because the square here, they will be 2 that comes out, so that we would not have any constants in yourderivative that two will be cancelling with this half otherwise this half makesno difference.I amI am minimizing J.So, whether I put this half or do not put this half, the minimizer will be the same.It is just that putting that halfis traditionso that thistwo from this square,Idifferentiate,we will cancel this half.That is the only reason the half is there.

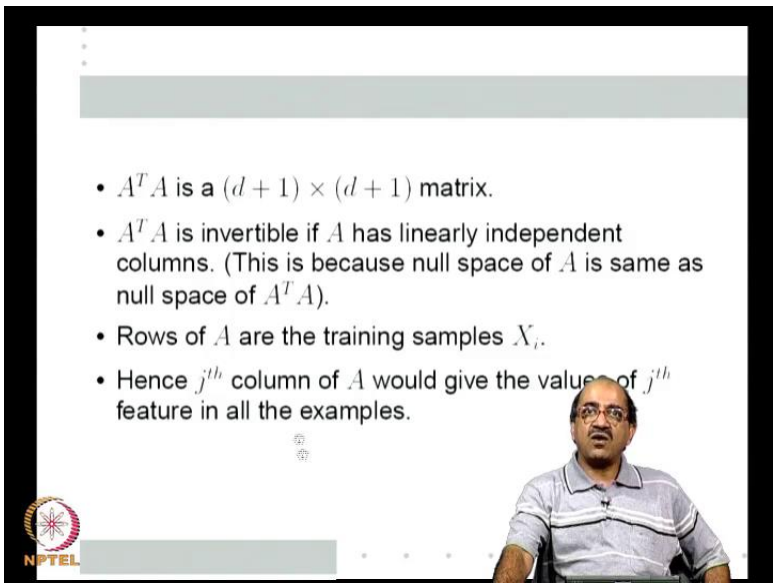
So, anyway now we have written JWin terms ofvector matrix notation as half AW minus y transpose AW minus y where A is a matrix,whose rows are the augumented feature vectors X1 and X2, all the way upto X n. yis all the targets. y 1 y 2 y n arranged as a n vector. W is the d plus 1 vector that I II want to find by minimizing this.So, to find minimizer of J, we need to equate its gradient to 0. JW is given by this.So,I need to find the gradient of this with respect to W.This is you know,AW minus y transpose,AW minus y.So, if Iif Idifferentiate with respect to W I, get AW minus y and multiplied by the coefficient of W. So, of course, we have toproperly take care of either A comes or A transpose comesbecause it is a vector vector matrix differentiation.

(Refer Slide Time: 31:01)



If we do all that, we get the gradient as a transpose times AW minus y. That has to be because AW minus y is An by 1vector.So,I can only multiply with a transpose and not with A.So,I know that when I differentiate,of course, the half will go away. When I differentiate, I get a transpose AW minus y.So, the gradient turns out to be a transpose into AW minus y.So, if I equate the gradient to 0, I get a transpose AW is equal to a transpose y. So, what we now know?Because we want to minimizeJ to find the optimal W, this is the gradient of J with respect to W.

(Refer Slide Time: 32:39)

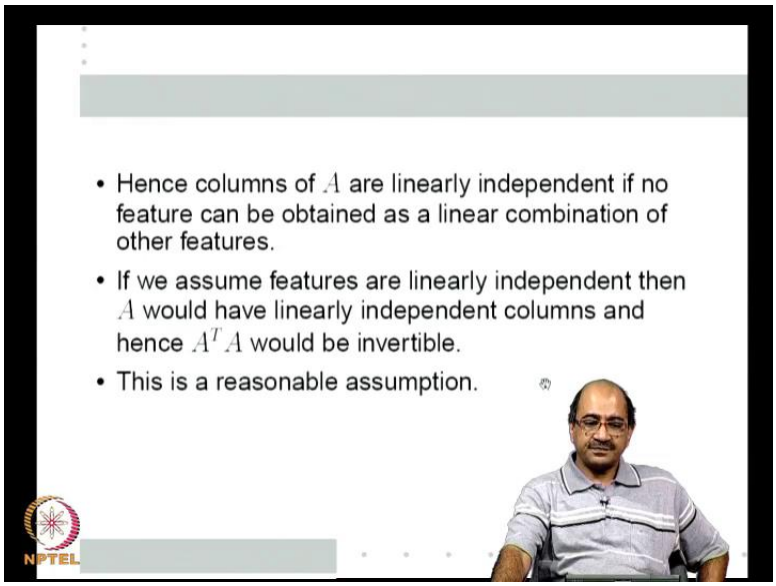


The minimizer of j W has to satisfy this equation.Now, this is some matrix, right? This is also,me given vector.So, this is a linear system of equations.So, what we can say is that the optimal w satisfies this system of linear equations.These are often called the normal equations. If you solve this system of linear equations, we can getW. A matter of fact, if A transpose A is invertible,I can pre-multiply both sides with A transpose A.I can write an expression for W.

Now, what does a transpose Abeing invertible mean?A transpose J is A d plus 1 by d plus 1 matrix. Recall that a transpose is A dA is A n by d plus 1 matrix. So, a transpose is a d plus 1 by n matrix. So, a transpose by A is A d plus 1 by d plus 1 matrix.A transpose A is invertible. If A has linearly independent columns, if A has full column rank then,A transpose A would be invertible.This is easy to showessentially by showing that the null space of A would be same as the null space of a transpose A.Anyway, ifeven, if you do not understand null space, does not matter.

A transpose A is invertible ifand only if A has linearly independent column; that is A has full column rank. Now,let us ask when will A have full column rank. Rows of A are the training samples Xi. So, what will be the ith or jth column ofA?So, in the first row, jth column would be the jth value in the first training sample. So, that is the value of the jth feature in the first training sample. The second row jth column will have the value of the jth feature in the second training sample. The jth column of A would give us values of the jth featurein all the examples.

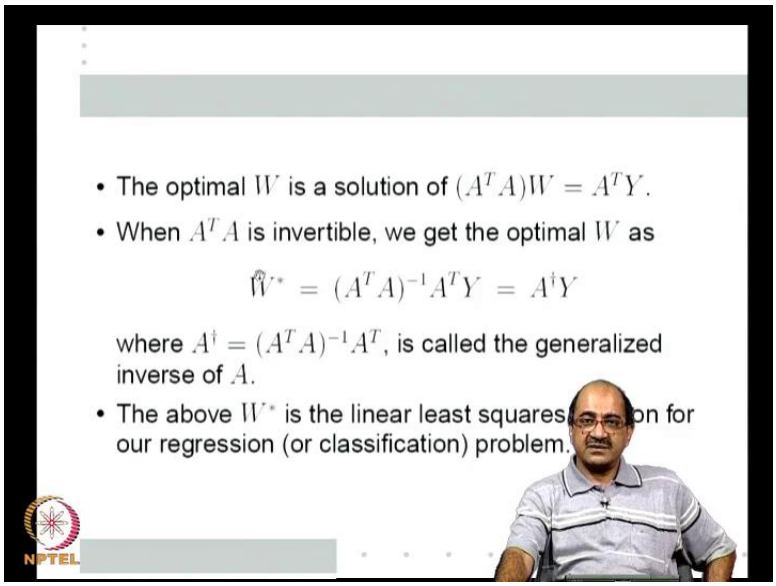
(Refer Slide Time: 34:29)



So, each column ofthe matrix A will give me the values of one particular feature across all.Example, if I think of X i's as feature vectors and and there are d features or d plus 1 feature say, the augumented case. So, each column of A will essentially put together values of one particular feature. In all the examples,what does columns of a being linearly independent means? No one column can be retained as a linear combination of the other columns.

Here, columns give you feature values.So, the columns will be linearly independent. If no feature can be obtained as a linear combination of other features, if one of the feature values can be predicted as a linear function of the other features, only then, the columns f A will not be linearly independent.But, that looks ridiculous because when we are looking at a linear classifier. If any one feature is a linear function of the other features, that feature is useless.

(Refer Slide Time: 35:40)



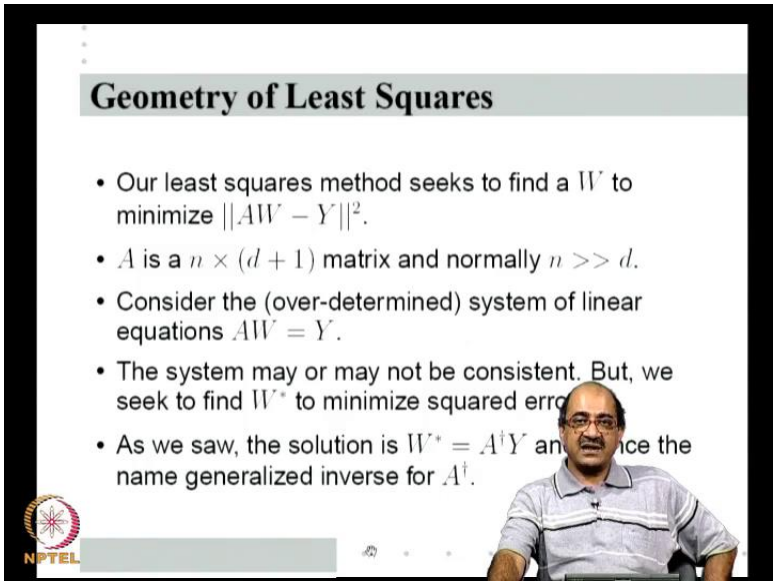
So, it isaaaa,it is reasonable to assume that our features are linearly independent. If I assume that the features are linearly independent, then A would have linearly independent columns. Hence,A transpose A would be invertible. As I just now said, this is a reasonable assumption, that is assumption, that a transpose A would be invertible because that simply means that my features are linearly independent.That is a very very reasonable assumption ineither pattern recognition or regression problem because I am learning a linear model.

If one of the features is obtainable as a linear function of the other features then, it should not be putting that featureat all in new model.Now, as you seen in the optimal W satisfies A transpose A into W's A transpose Y. So, if A transpose A is invertible, I can write my optimal W, which I will write as W star as A transpose A whole inverse into a transpose Y.This A transpose A whole inverse into a transpose, that matrix is often calledA dagger.

This sign is called dagger.This is called A dagger.So, I can write W star as A dagger Y.This A dagger is often called generalized inverse of A.Because A is a rectangular matrix, so it does not have an inverse.But, this A dagger is called generalized inverse of A.We will currently see why it is called the generalized inverse of A.But, any case starting from my normal equations; if A transpose A is invertible,I can write W star as A transpose A whole inverse A transpose into Y.

This W star is the linear least square solution to a regression or classification problem.Essentially, if I am given my feature vectors, the my exampleexampletraining patterns, that is I am given XiYi then,I can form the matrix A.I can form the vector capitalY.Then,I can calculate A transpose A whole inverse A transpose intoY.That will give me W star.

(Refer Slide Time: 37:16)



So, given the training data,I I can use this expression to compute W star. So, this W star is your linear least square solution. That is your final regression function of classifier.So,let us look a little bit intowhat the solution means.What is our least squares method trying to do?As we have seen thatthat function j is simply the norm square of the vector AW minus Y, right?Now, what is AA, is A n by d plus 1 matrix.W is a d plus 1 vector.Y is A n vector.Normally see, d is the dimension of the feature vectors. n is thenumber of examples.

Normally, we have many more examples that the dimension of feature vector has otherwiseis really meaningless.If you give me, you know one point inR 2 and ask me to find a classifier,it is hardly meaningful.I should get many many more points; certainly more than two points, so really say whether linearly separable or not.So, in most normal problems of function learning or classifier learning, n will be greater than d. n is often much larger than d.Now, let us look at this.AW is equal to Y system.This is a system of

linear equations.W is the d vector that is the unknown.d plus 1 vector, that is the unknown.

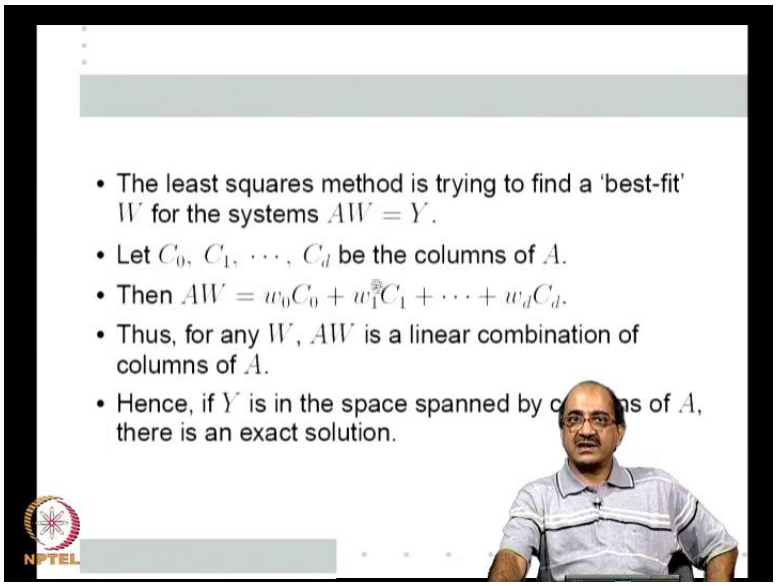
Now,A is a rectangular matrix, n by d plus 1.So, in a linear system like this, n the rows of A, n could be the number of equations. The columns of A that is d plus 1 same as the dimension W, those are the unknowns.Because n is much larger thand,this is an over determined system of equations.We have more equations than unknowns. Now, the system may be consistent, may not be consistent.As you know, when we given an over determined system of linear equations since the system is not consistent would not have a solution. Beforehand, we have no way of knowing their solution.

They may not even be because we are just fitting a function. The actualXiYi may not be on a line on a hyper plane. So,i may not be,i may not even be able to satisfyW transpose Xi is equal to Y I, for each i.So, the system may not be consistent. Whether it is consistent or not, we are asking find AW, but even if it does not exactly solve this;it is all set in the sense of minimizing errors over the training data.That is, find a least square solution of this.

So, if I cannot exactly satisfy AW minus YA equal to Y, what I am saying is atleast finds me some W, sothat the norm AW minus Y whole square is small.That is, themean square error solution for this over determined system of equations.As we sawthat solution is, one can write it as W star is equal to a dagger Y. So essentially, I am solving this linear system with A as a rectangular matrix.So, really A does not have an inverse. But, if A had an inverse, i would have return W is equals to a inverse y.

Ultimately, when I did the least square solution of finding AW such that AW minus Y norm square is minimum, my solution is given by W star is some matrix times A. That matrix is given the symbol A dagger. That is the reason why A dagger is called the generalized inverse.It is like an inversefor a solving a linear system of equations, using the inverse of the coefficient matrix. We are essentially doing the same thing except in a mean square sense. The solution once again is W star, is some matrix into the right hand side. So, that matrix is A dagger.

(Refer Slide Time: 40:56)

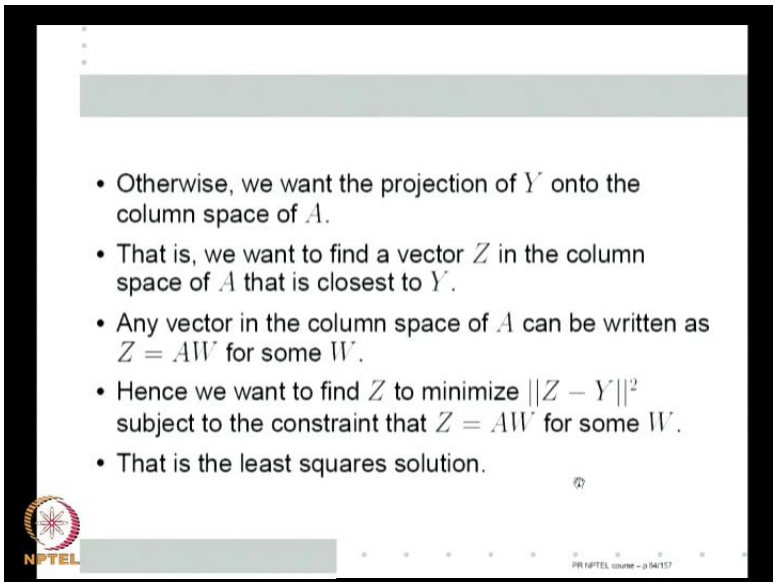


So, A dagger is called the generalized inverse of A.Let us look a little moreinto the solution.So, the least squares method is tryingto find the best fit W for the system of linear equations. AW is equal to Y. What does the best fit solution mean? Suppose columns of A are C0, C 1, C d.There are d plus 1 columns of A.Let us call them,C0, C 1, C d. Now, when I multiply anymatrix on the right with a vector; essentially, what I get is another vector. It is a linear combination of the columns of the matrix whose linear combination coefficients; are the coefficient of the vectors.

So, any AW is simply, if W has components,W0, W 1, W d, then, AW will be written as W0C0 plus W1 C 1 plus W d C d, so the AW vector is simply W0C0 plus W 1 C1 plus W d C d vector.So, every AW is essentially a linear combination of the column vectors of A.So, when I search over all possible W,all I can get out of AW are linearlinear combinations ofcolumns of A.So, for any W, AW will be a linear combination of columns of A.

So, when can I solve this system exactly?If Y is in the space spanned by columns of A; space spanned by the columns of A is called the column space.What do we mean by space spanned by the columns of A?If you consider this,the vectors space obtained as all possible linear combinations of columns of A.All vector that can be obtained as linear combination of columns of A, that is called the column space of A.

(Refer Slide Time: 42:51)



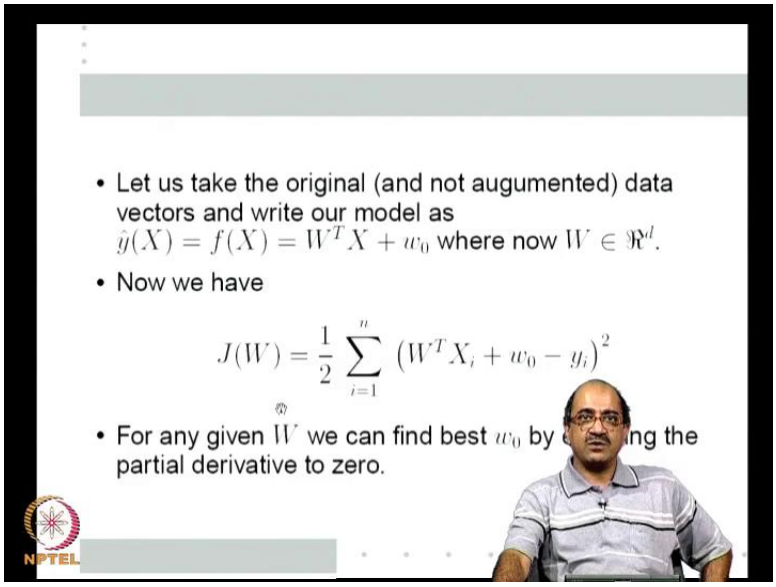
So, if the vector Y happens to be in column space of A, then we have an exact solution.There will be some linear combination of columns of A that will give me the Y.That linear combination is the W vector, butsuppose, we do not Y is not in the column space of A, then what is the best we can do?Then, we are asking project Y onto the column spaceof A.What do you mean by project Y on to the column space of A?

That is, we want to find a vector Z, which is in the column space of A and that is closest to Y.Because AW can only give me vectors in the column space of A,I cannot now satisfy AW is equal to Y anyway.So, the best I can do is to find a vector in the column space of A that isthat is that has least distance to Y.That is the difference between that vector and Y is the smallest.So, that is what is meant by project Y on to the column space of A.

How do I find such a Z?To find such as Z now, any vector in the column space of A can be written as AW for some W.Thus,that is what this definition of column space of A.So, finding a vector Z in the column space of A that is closest to Y is same aswe want to find Z to minimize Z minusY norm square.We want to find Z to minimize Z minus Y norm square. That is the distance between Z and Y under the constraint that Z has to be AW, for some W because,Z has to be in the column space.So essentially, projecting Y on to the column space of A is same as minimizing.

Now, because Z has to be AW,I can put this as AW.So, minimize AW minus Y norm square.So, minimizing AW minus Y norm square is same as projecting Y on to column space of A, that is the least square solution.A matter of fact, what it means is, if I have A vectors of space defined through some vectorsX 1, X 2, X n. So, if I, if there is a vector Y and I want to project it on to the space spanned by X 1, X 2, X n, then,I can organize those X'sas rows of matrix A. Take A transpose,A whole inverse,A transpose and multiplypost multiply with this. Then, that will be the projection.

(Refer Slide Time: 45:42)



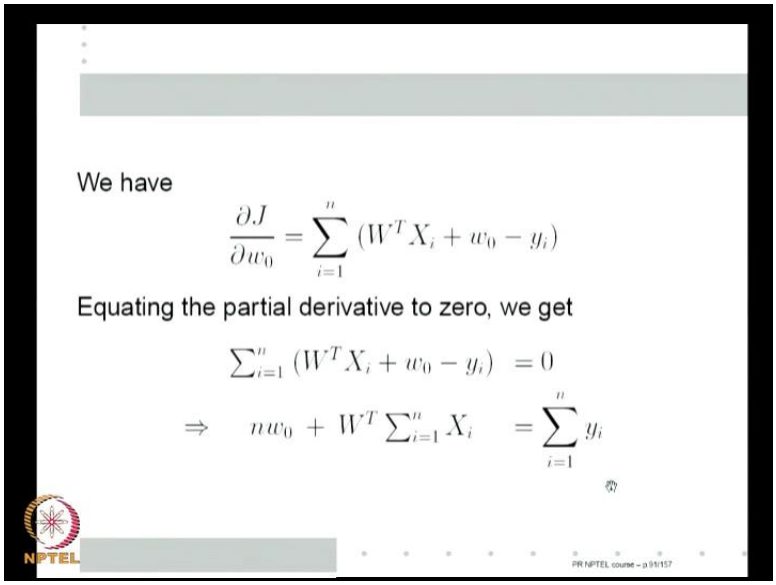
So, as a matter of fact, this A transpose,A whole inverse,A transpose is also called a projection matrix.It projects Y on to the column space of A.So, least square solution can be looked at as obtainingan approximate solution to an over determinant system of equations, which may not be consistent hence, essentially projecting this vector Y on to the column space of A. The projection matrix is given by A transpose, A whole inverse, A transpose.Now, let us look at a few more things.

We have beenwe have been saying that we can either use augumentedaugumentedrepresentation or representation of the linear function as W transpose X. So,let us ask what the extra constant does.See, essentially if I have this f X defined as W transpose X plus w naught is not a linear function, in the sense f of X 1 plus X 2 is not equal to f of X1 plus f of X 2 because of this w 0.Of course, is called an f find function.It is just aaconstant plus a linear function. We artificially made into a linear

function, not artificially.We made it a linear function by simply working in the augumentedspace. But, suppose you want to work in the original space and ask what is this extra constant doing?Why did I need this constant?

Why couldn’tI have simply done W transpose X?To see that,let us go back to what we minimize by this time.Let us not work in the augumented space, butwork in the original space.So, I will write J W ashalf i is equal to 1to n.W transpose Xi plus w naught minus y i whole square,so, minimizing this, minimizing this.Now, with respect to both w and w naught, I should have put w naught also here. But, anyway that does not matter.

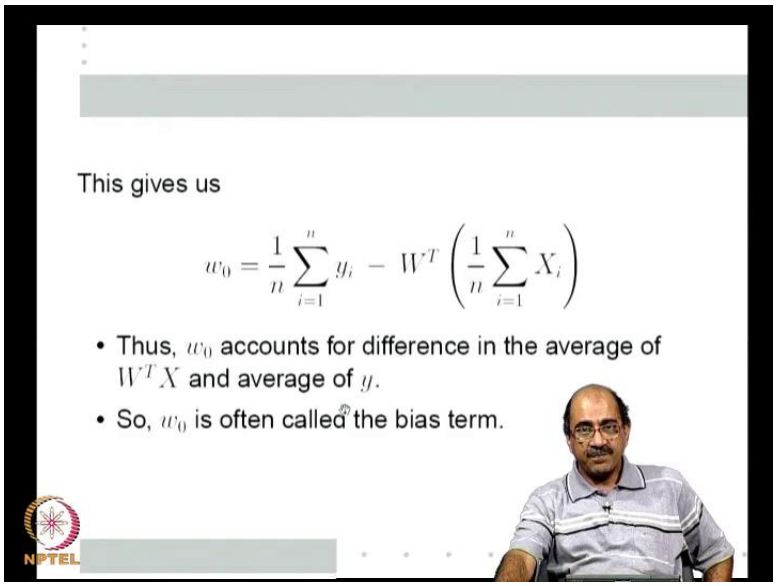
(Refer Slide Time: 47:17)



So, let us ask if I fix AW vector, what is the best wnaught, I can get?For a fixed w, if I want to get the best w naught then,I have to find the partial derivatives of J with respect to w naught and equate it to 0. That will give me the best w naught. So,let us look at thatbest w naught.So, what is delJ by del w naught?This is J.If Idifferentiate it with respect to w naught,I get this 2 the half cancelssummation; this into W transpose Xi plus w naught minus y I, because, w naught has no other coefficient;that is the whole derivative.

So,del j by del w naught is given by this. So, if I equate this to 0, what do we get? This is equal to 0. i is equal to 1to nW transpose Xi plus w naught minus y equal to 0.Now, youtake the summation inside i.Get n times w naught plus W transposetimes summation x i. You take y i on this side, is summation y i.

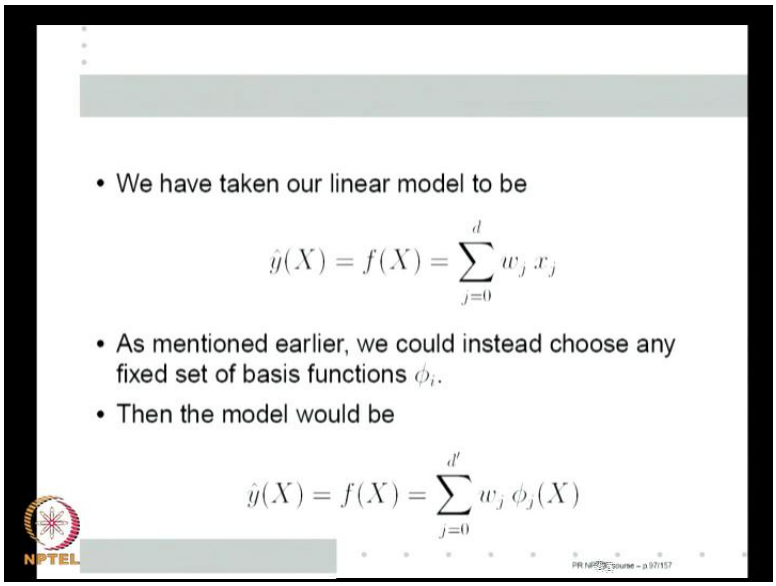
(Refer Slide Time: 48:00)



So,I can solve this to get w naught. That gives me w naught is 1 by n.See this, n will come this side.If Idivide this by n, is1 by n summation y i minus 1 by W transpose summation Xi, which I can write it as 1 by n summation y i minus w transpose 1 by n summation Xi. What is this 1 by n summation Xi?This is the average of Xi.This is the average of y i.Soessentially, w naught accounts the difference in the average of W transpose X and average of y.

If I ultimately wanted to represent y as WtransposeX, assuming essentially thinking of y and X asrandom, what it should mean?This expected value, y should be equal to W transpose, expect value of x.So, if expected value of y, is the average of y is not equal to W transpose expect value of X. Then,i have to account for this difference because that cannot be accounted for my linear function W transpose X.That difference is accounted for by w naught. That is the reason, why we need a constraint essentially.

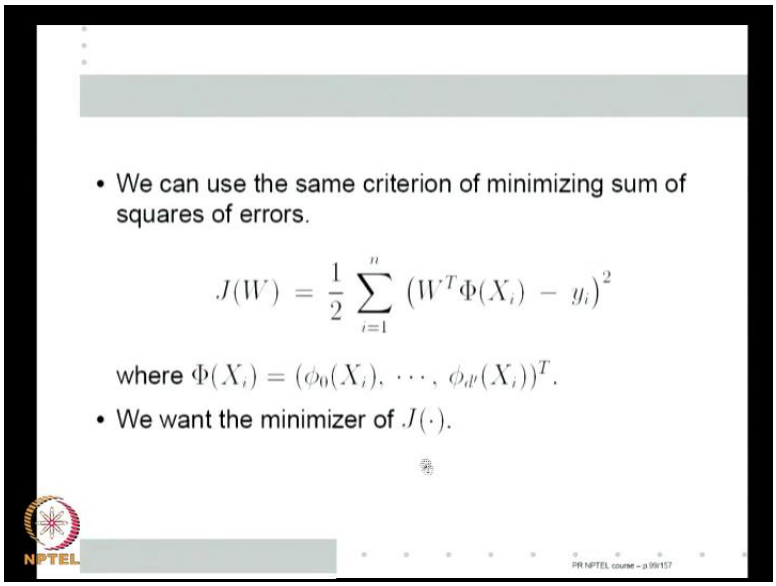
(Refer Slide Time: 49:32)



If some of these these averages can be matched, otherwise then we do not need the constants.So, w 0 is often called the bias term because it is time to adjust between the averages of, average of y to the average of W transpose X.So,that is why whenever we writea general linear model as W transpose X plus w naught; the w naught is called the bias term.A few more things about our linear model.We are taking our linear model once again.Let us put w naught back into this and take X of 0 to be 1, so that, we are working back in the augumentedspace.So, our model is y hat X is summation J is equal to 0 to d w j x j.

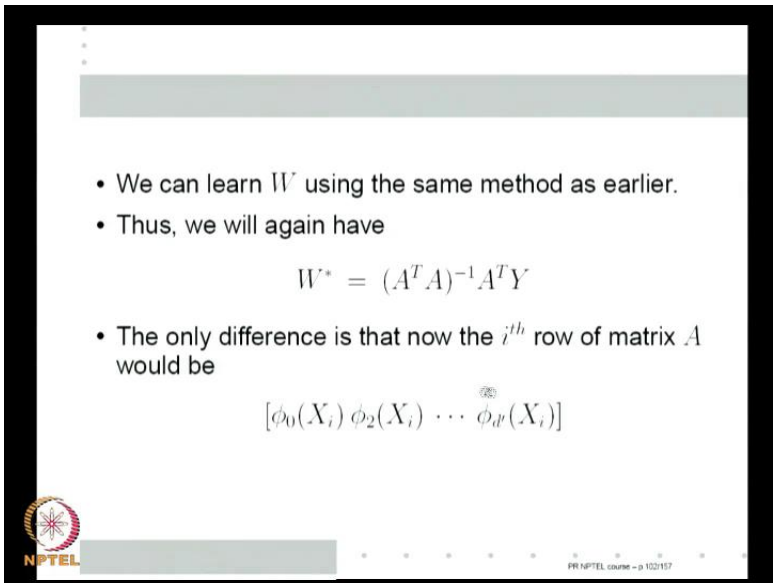
Now,as mentioned earlier, we do not have to use x j.We can use any fixed sets of basis functionsy i. What does that mean?The model could have been y hat x is equal to that is f of x is equal to j is equal to 0 to no d prime.It could be any number of phi functions; need not naturally be equal to the dimension of x j is equal to 0 to d prime.w j phi i of x jy hatcan be written as j is equal to 0.d prime w j phi j of xj is equal to 0.d prime,this d becomes d prime. So, this as we seen is also a linear model.

(Refer Slide Time: 50:53)



A matter of fact, if I have chosen this as the model using any functions phi exactly the same method that we have seen so far will work, right? Why I can now take the same criterion minimizing sum of squares of zeroes?

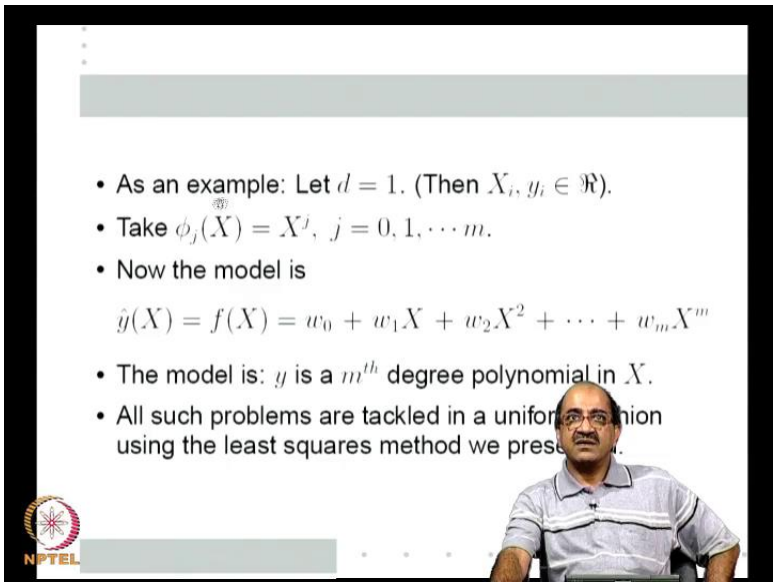
(Refer Slide Time: 51:47)



That is half W transpose, say now,I call it phi X. See, the thethethethe sum is w j phi j x that is w 0 phi 0 x, w 1 phi 1 x, w 2 phi 2 x and so on.So, if I write it as w transpose phi x I, this vector phi x i should have components phi 0 x iphi d prime x i. So,I can write j as i is equal to 1to n w transpose capital phi of x i minus phi whole square.So, if I put capital

phi of Xi in place of x i, thenexactly the samealgebra goes through.We want the minimizer of j.The minimizer of j can be learnt using the same method as earlier.We will once again get a transpose A, inverse A, transpose y. The only difference is that now, the rows of A earlier, the i-th row of A is x i.

(Refer Slide Time: 52:19)



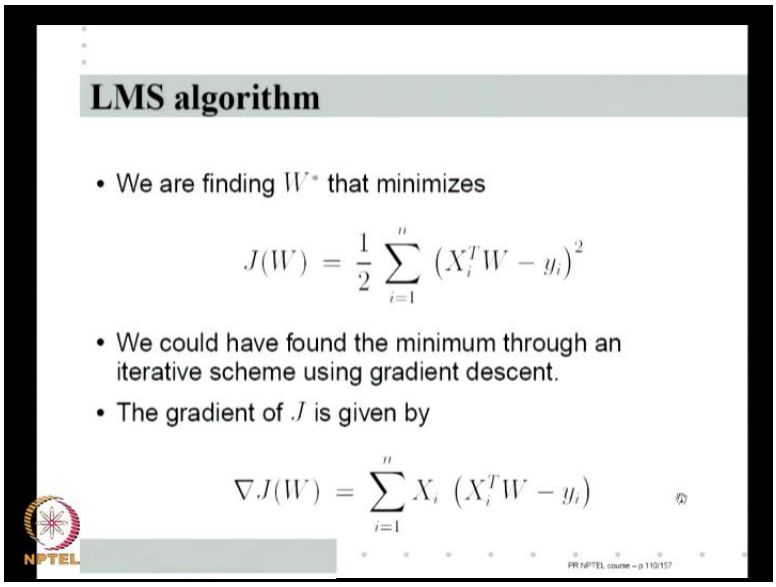
Now, instead of that, it will be capital phi of X I, which is phi 0 of x i, phi 1 of x I, all the way upto phi d prime of x i. So, the only difference is that i-th row of matrix A would bewould be now the capital phi of x I, except for this exactly same thing go.So,let us look at one example of this generality.Let us take d is equal to 1.1 dimension case, so that means both x i and y i are belonging to r.So, my training data said, this is simply real numbers pairs of real numbers, x and y, x 1 y 1, x 2 y 2, x n y n.You want to find a function relation with x and y.

What is this problem?This is thefamiliar curve fitting problem that all of you would had.So, you want to put to y is equal to f X.Then,you find theleast squares curve fitting.The least squares curve fitting is you take f Xi minus phi i whole square, sum over i and minimize that.You, most of you would have done the linear feet that is,you do y yi minus m x i minus c whole square, differentiate it with a fine (( )), which is equal to m and c. Then, find the optimal m and c. That is the least square's curve fitting.So, the problem that we have is a least square's curve fitting.

Let us say, we want a generic least square's curve fitting what kind of curve.Let us take phi j of X to be X to the power J. J is equals to 0to n.So, what does the model mean now? y hat x is f of X , w 0 phi 0 of x phi 0 of x is 1 and w 1 of phi one1 x phi 1 x is x to the power 1 and so on. So, y hat x is a w 0 plus w 1 x plus w 2 x square plus w m x power m.So,what is this?This is nothing but a polynomial index.So, if I want to model y, as the mth degree polynomial on x,I can simply think ofit as a linear least square's problem say, mth degree polynomial is settling at an linear curve.

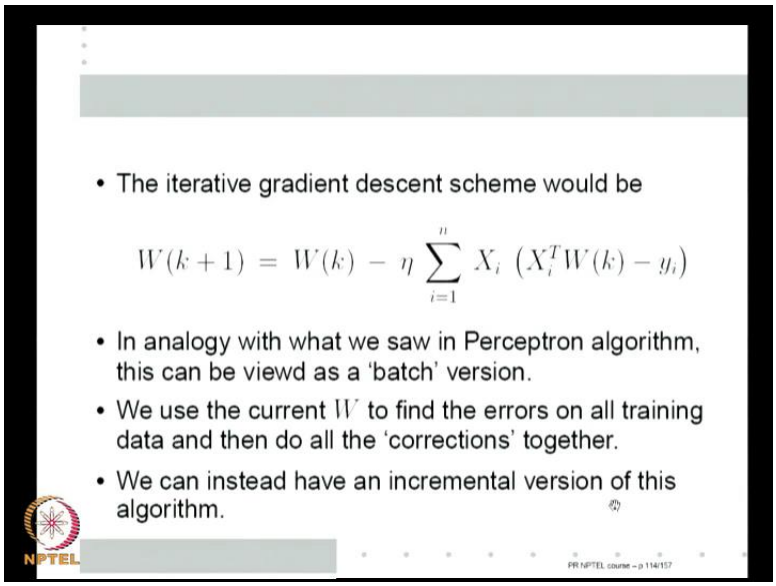
But, I can think of it as a linear least squares problem by simply working with phi j of x is equals to x power j.So, the model is,I want to model y as an nth degree polynomial on x.I want to find the best coefficients for the polynomialw 0, w 1, w m.I can solve it using the same linearmethod that I have of this,the generalized inverse method.

(Refer Slide Time: 54:48)



So, the all such problems are tackled in a uniform fashion using the least squares method that weproposed here.So, this is what essentially the generality that you get with respect to phi means.We are finding.Let us just look at a little more structure in this.We introduce one more algorithm what is called the LMS algorithm.We are findingAW star to minimize this.To minimize any function,I can use gradient descent.If I want to minimize a function f of X where X is a vector then,III can do an iterative algorithm X of k plus is X of k minus theta times gradient of f of X k.

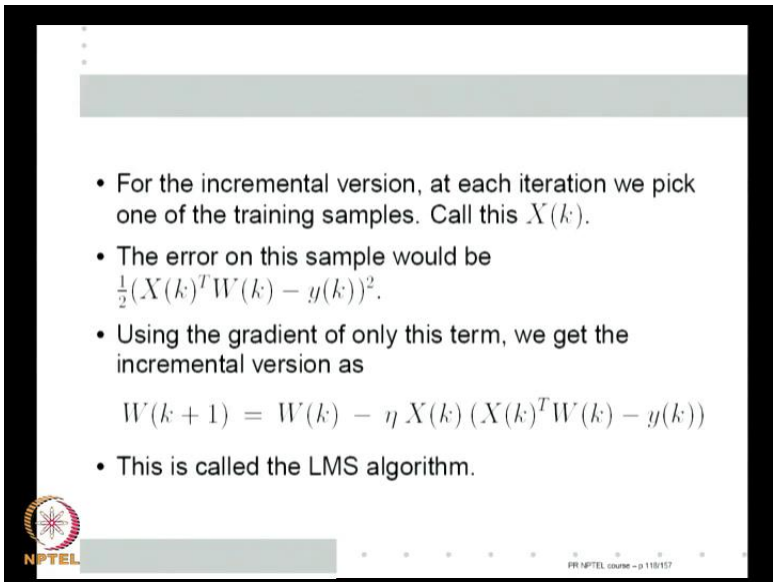
(Refer Slide Time: 55:43)



That algorithm converges to the local minimum. So, i can find minimizer using gradient descent. So, we could have found the minimum through an iterative scheme using gradient descent.What is the gradient of j?Gradient of j isonce again the 2 cancels with half. So,I get Xi into x i transpose W minus y i. So, what will be an iterative gradient descent scheme for this?So,W k plus 1 is equal to W k minus some step size eta into my gradient is this. i is equal to one n Xi into Xi transpose W minus y i.

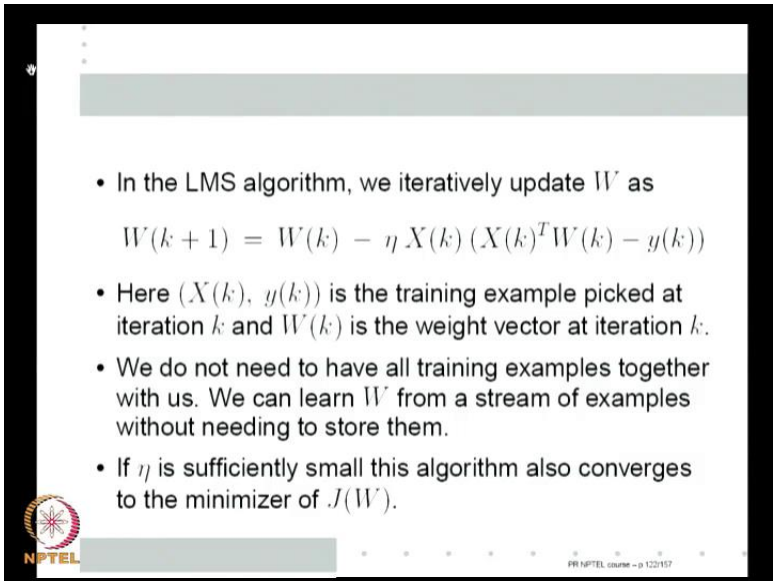
Now, if you look at it, in analogy with perceptron, it looks like a batch version of the algorithm. Given a current W, these are the errors I am making on each of the training samples. On i training sample, my error is Xi transpose W k minus y i. So, I calculate the gradient with respect to all the errors and then do correction together.So, that is a batch version.I could as well have done it incrementally.So, here we are first using the current W.We find errors on all the training data.Then, do all the corrections together instead of that we could do an incremental version of the algorithm.

(Refer Slide Time: 56:35)



What will the incremental version mean?In the incremental version, at each iteration, we pick one of the training samples (( )) X k and correspondingly Y k.Then, the error on this sample will be X k transpose W k minus y k whole square.So, using only that gradient in the gradient descent,IgetW k plus 1 equal to W k minus theta X k into X k transpose W k minus y k. That is the gradient. This is called theLMS or the least mean square algorithm.

(Refer Slide Time: 57:14)



So,LMS algorithm is essentially the linear least squares method, where I minimize J using an iterativegradient descent but, using anincremental algorithm.So, that is the LMS algorithm. We update W iteratively like this. Here, X k Y k is the training sample with data iteration k. W k is the weight vector at iteration k. So, like in the perceptron case because is incremental, we we do not need to store all the examples. We can have examples coming in a stream.

We can do from there, if it has sufficiently small. This algorithm also converges to the minimizer of j (( )). Such an algorithm is called the LMS algorithm.So, next class, we will start with the LMS algorithm andsee. LMS algorithm was also as old as perceptron.It also a come from a simple modelcalled Adaline.So, we look at the LMS algorithm, various ramifications of LMS algorithm and then look at more structurein the linear least squares method.

Thank you.