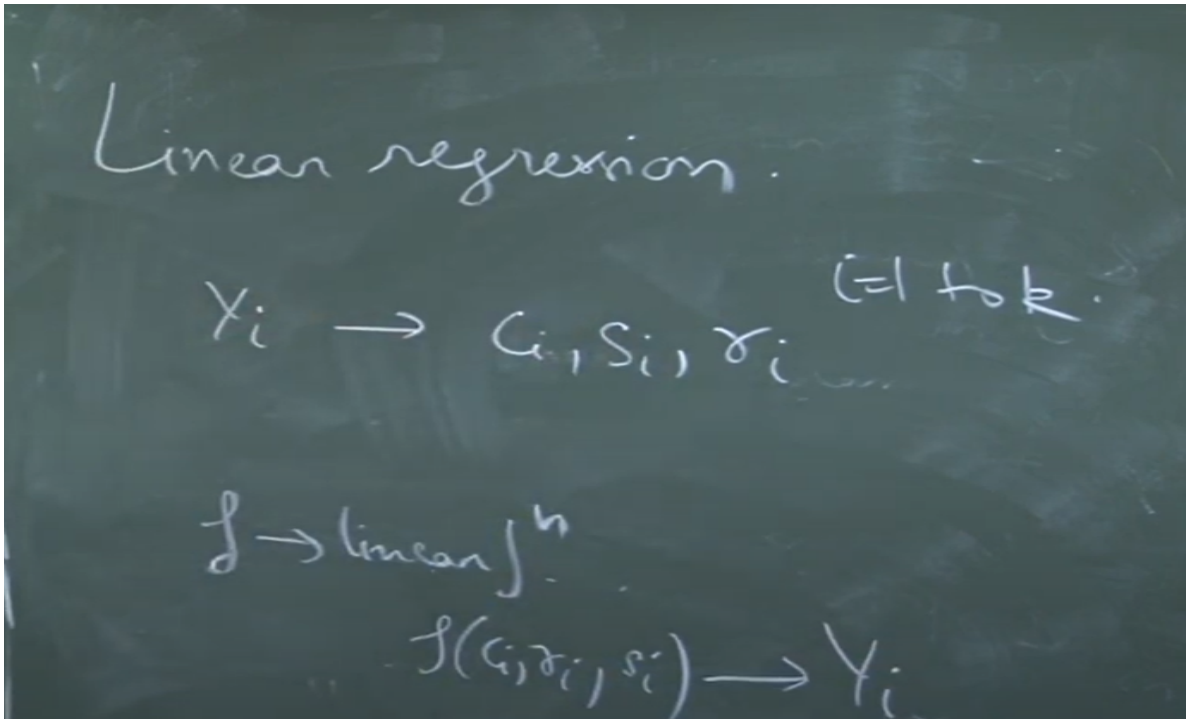
Linear Programming and its Applications to Computer Science

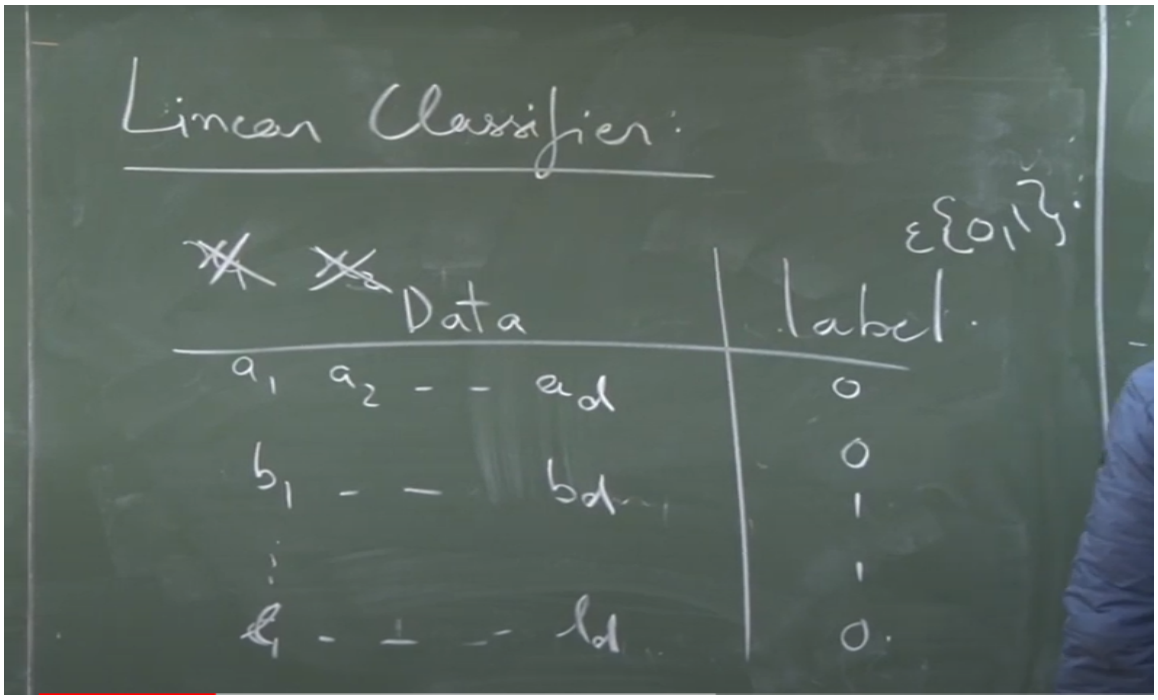
Prof. Rajat Mittal
Department of Computer Science and Engineering
Indian Institute Of Technology, Kanpur

Lecture – 49
Linear Classifiers through LP

We were already talking about machine learning and linear programming. Linear programming is our favorite because we are studying this topic and machine learning is the favorite for everyone and we wanted to see how to relate these two. We already saw one application which was linear regression.



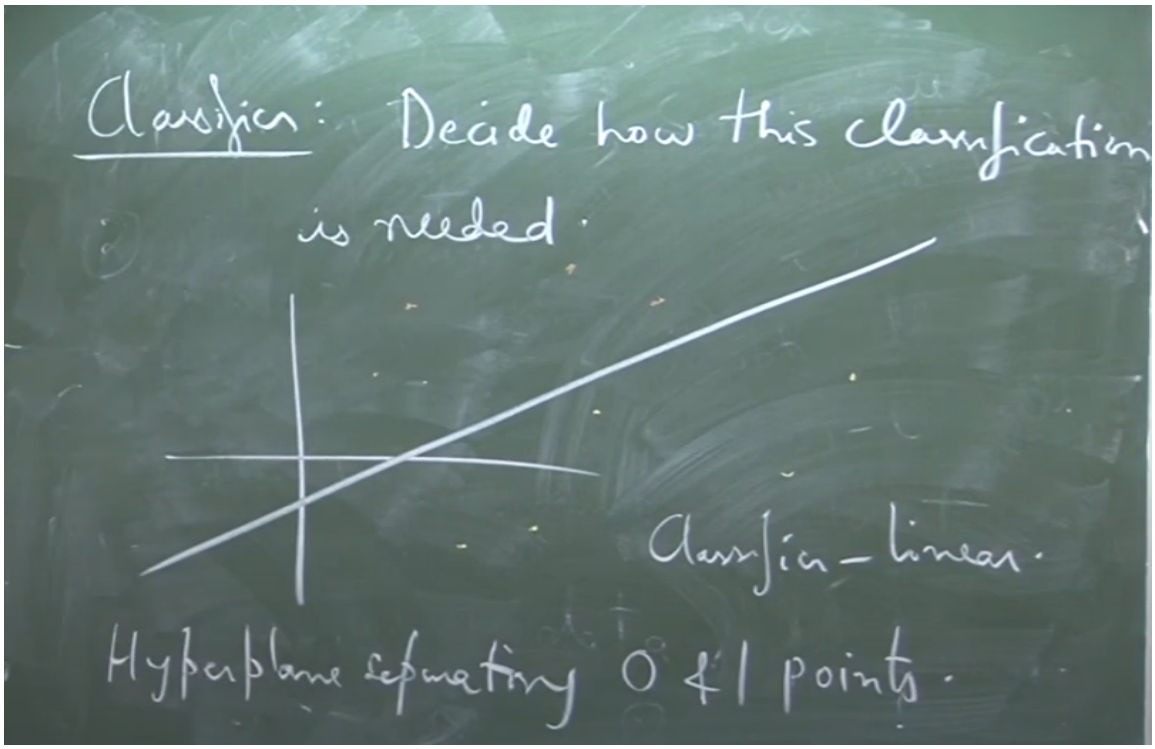
The idea was we were given these data points for let us say i is equal to 1 to k and our task was to find a linear function such that f of ci, ri, si is very close to Yi.



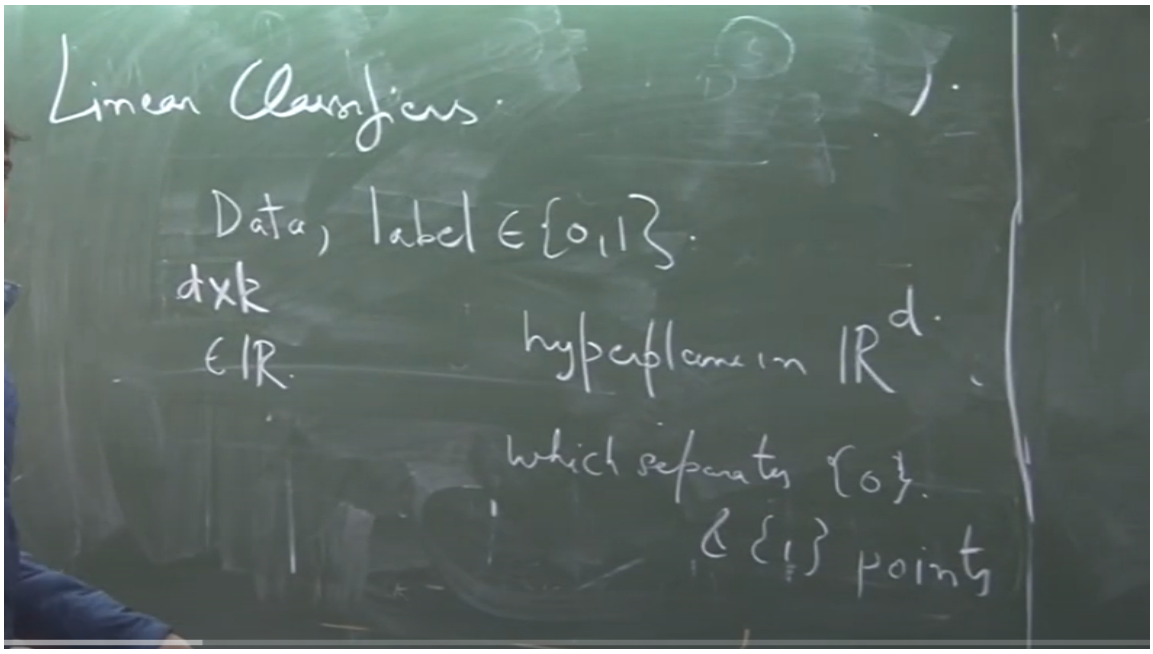
This is called a linear regression problem. We are going to study a slightly different problem which is called a linear classifier.

In the case of linear classifier, the data is almost going to be the same. We are going to have x1, x2 or probably I should say. Let us say we have these data points in d dimension. I have d dimension d parameters. These are given. This is my data. The difference in the case of linear classifier is that my now label is going to be an element of 0, 1. It is going to be a Boolean value. You can think of it as yes or no.

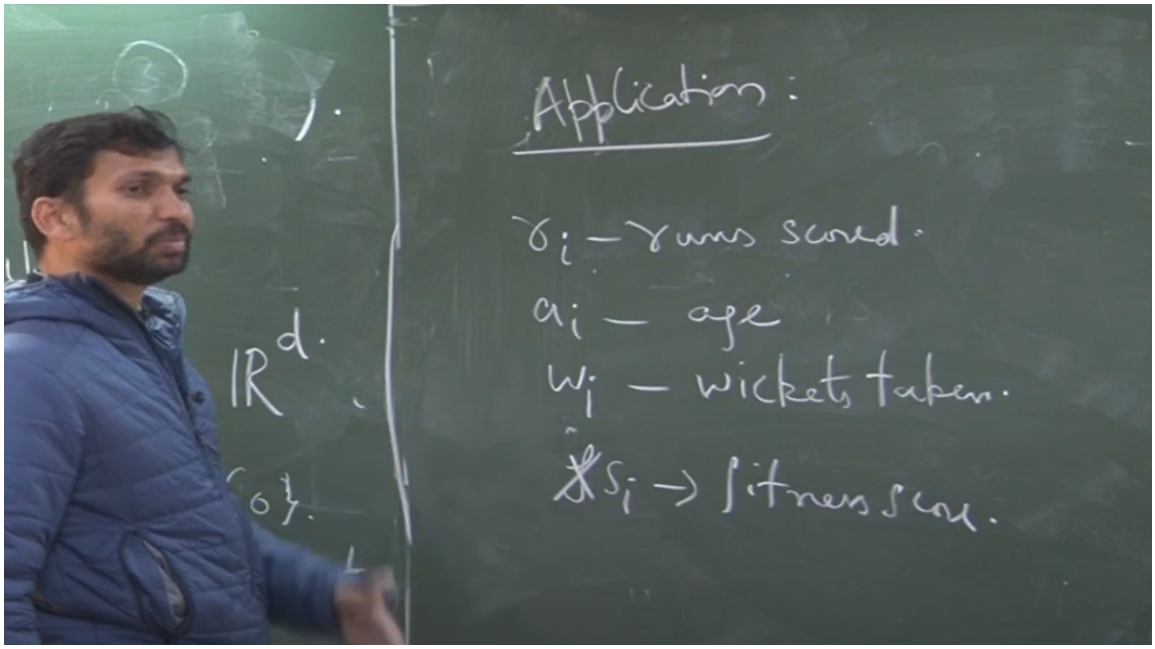
You can think of it as true or false, but some of the data is going to 0. Some of the data is going to be 1 is to decide how this classification is done. If we take q from the regression problem, we need to find a function whose output is Boolean 0 or 1 and it should be 0 on the data points here and 1 on the data points which are 1. Again if we are looking for a general function, things become really difficult.



Let us say I describe the 0 points by orange dots and 1 point by yellow dots. I want to figure out how to separate them. How do I decide which points are yellow, which point are orange and the way I have drawn this, you might see that this is a nice way to decide which point is orange, which point is yellow. Again by applications, if we are given that our classifier is linear, then we got the linear classifier problem where you want to find a hyper plane separating 0 and 1 points or we can even ask the easy question is there a hyper plane which separates the 0 and 1 points. Let me write the problem once again clearly.

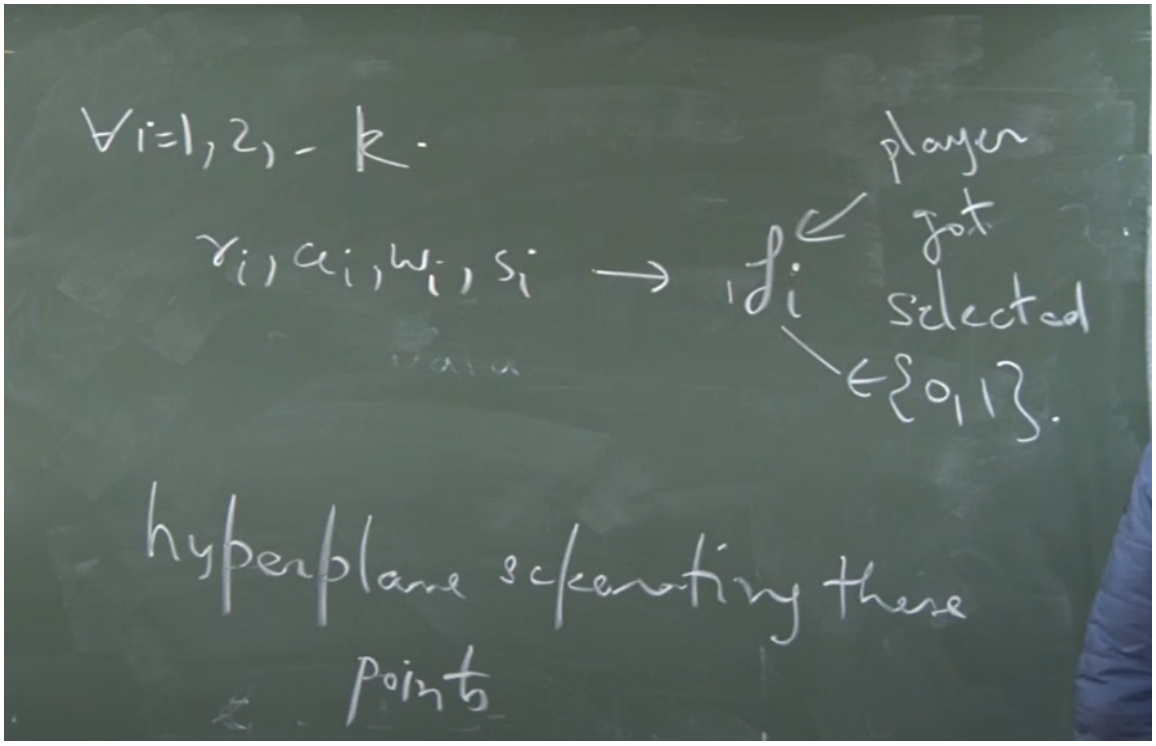


I am given the data, I am given the labels. Again you can think of the data as an m cross n matrix or in this case since there are d dimensions and k data points, it is going to be a d cross k matrix of real numbers. Given this data and labels, you want to come up with a hyper plane in r to the d separates 0 and 1 points. This is the problem we are looking at. I am already talking about machine learning.



This lecture is going to be really famous. The only thing now to do is to take application from cricket and this video will become India famous. So, let me take an application in cricket. To clarify what we are doing, let us say a player whether a player gets selected or not depends on 4 factors which are runs scored, age, wickets taken. I hope in Indian context I do not want to explain what are runs and what are wickets and

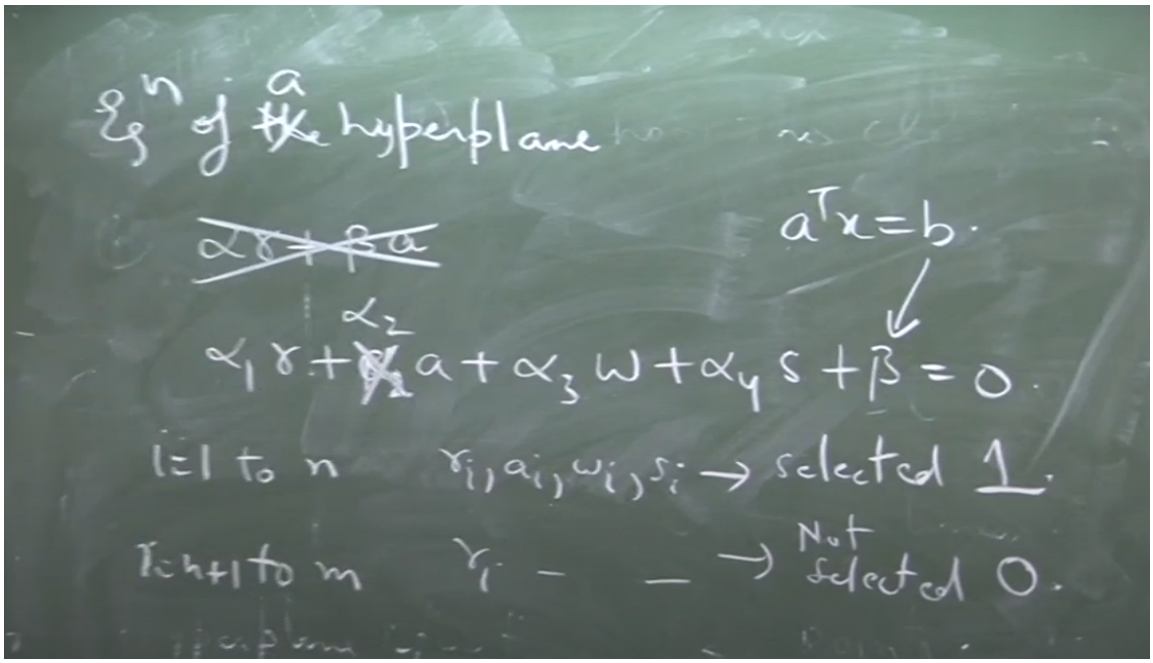
then some fitness score because we are in new age and fitness also matters.



Given these 4 numbers, we know in past which players were selected, which players were not selected. That means for k players, I have ri, ai, wi, si and final fi which is whether the player got selected or not. This is an element of 0 comma 1. This is an example of linear classification. We are given now in future some other ri, ai, wi, si need to figure out whether that player will get selected or not.

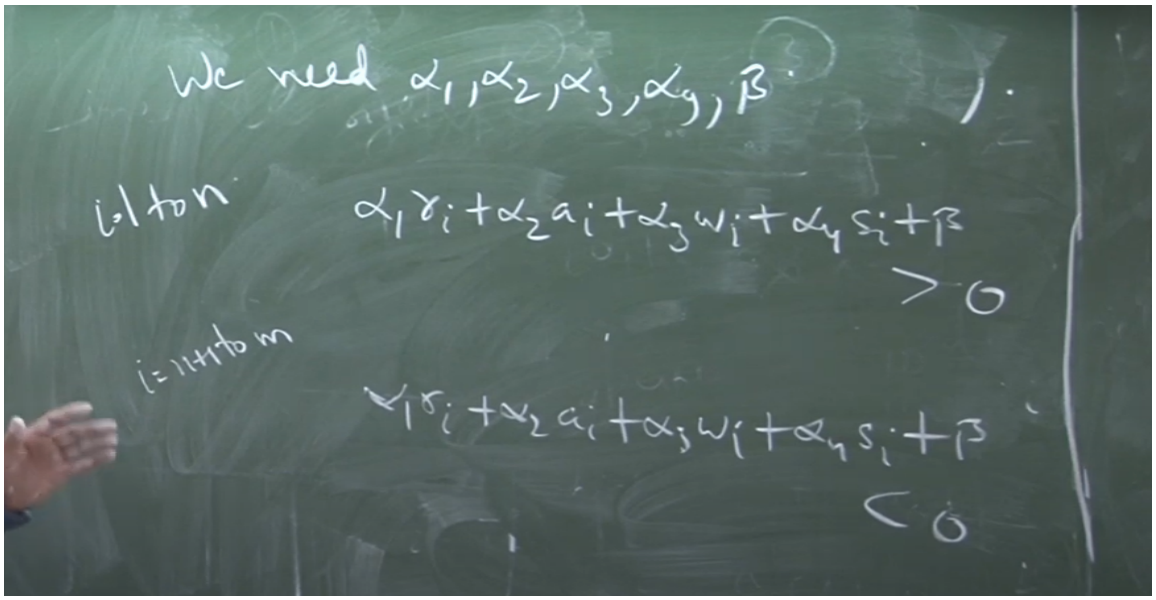
This is one way to decide whether the player will get selected or not using machine learning. And again, we are going to see a linear programming solution to this problem. You might guess that linear programming should be involved because we are looking for a hyperplane calculating these points. First question is what is the equation of a hyperplane? In d dimension or let us keep it as 4 dimension, the equation would be alpha r plus since there are lot of these and I do not want to use many Greek symbols. Let us say alpha 1 r plus beta 1 a alpha 2 a alpha 3 w plus alpha 4 s with a constant equal to 0.

This is the equation of an hyperplane. We have seen this. Generally, we write it as a transpose x equal to b, b is beta here, the coordinates of a are denoted by alpha 1 alpha 2 alpha 3 alpha 4. This is the equation of hyperplane and what we are looking for is a hyperplane separating these points. Let us make our life easier.



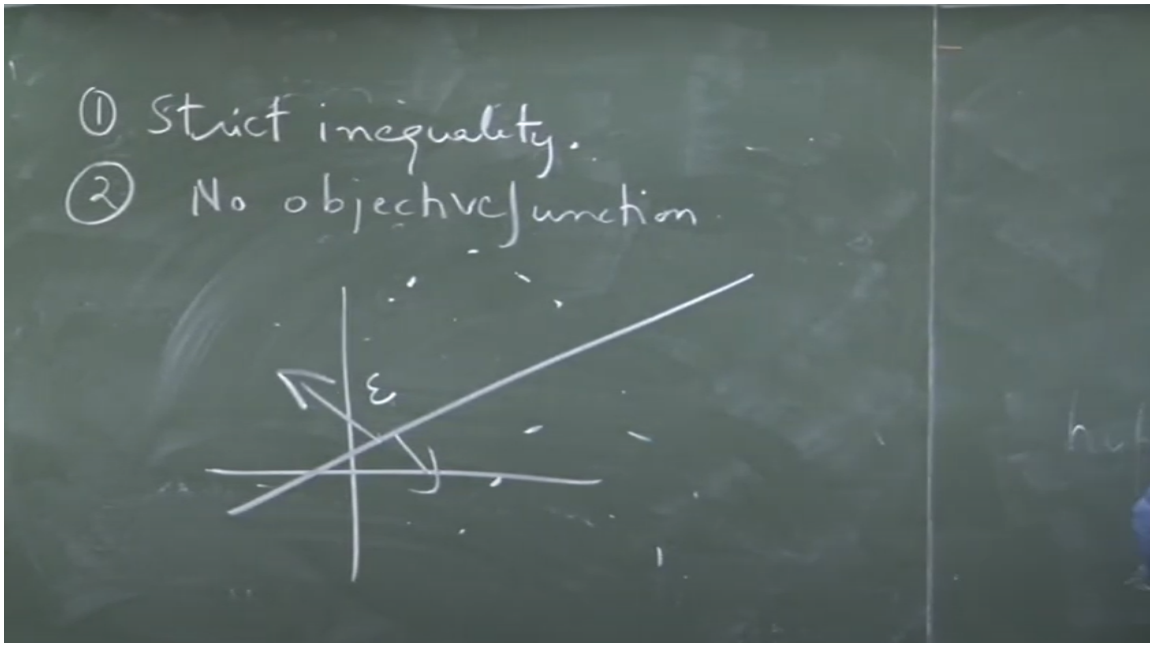
Let us say for i 1 to n, we can sort our points such that for all these, these are the selected players or the value was 1 and from n plus 1 to m, these are not selected. The value was 0. We want to distinguish between these 2 set of points. Writing it mathematically, we need a hyperplane. What is the hyperplane for any point between 1 to n alpha 1 r i plus alpha 2 a i plus alpha 3 w i beta should be strictly greater than 0.

And for i, we would want the same quantity to be less than equal to 0. This is the problem we want to solve. We want to find alpha 1 alpha 2 alpha 3 alpha 4 and beta such that this



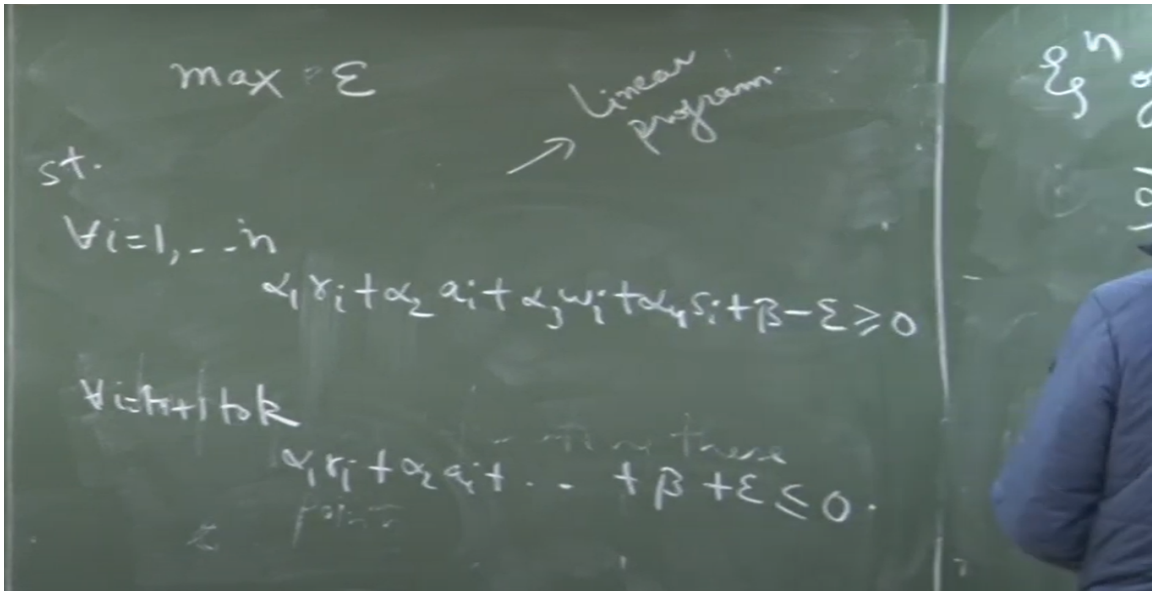
quantity is greater than 0, this quantity is less than 0. Question is does such alpha exist? If such alphas exist, can we find them? When I say alpha, any alpha is in beta. Again it

almost looks like linear program.



There are only constraints and the inequalities are strict. Let me just write out 2 problems, strict inequality and no objective function. In some sense, this looks like a feasibility problem, linear feasibility problem. We will take care of both of them in a single go. What we are going to say is that let us say this hyperplane which separates these points is at some distance away.

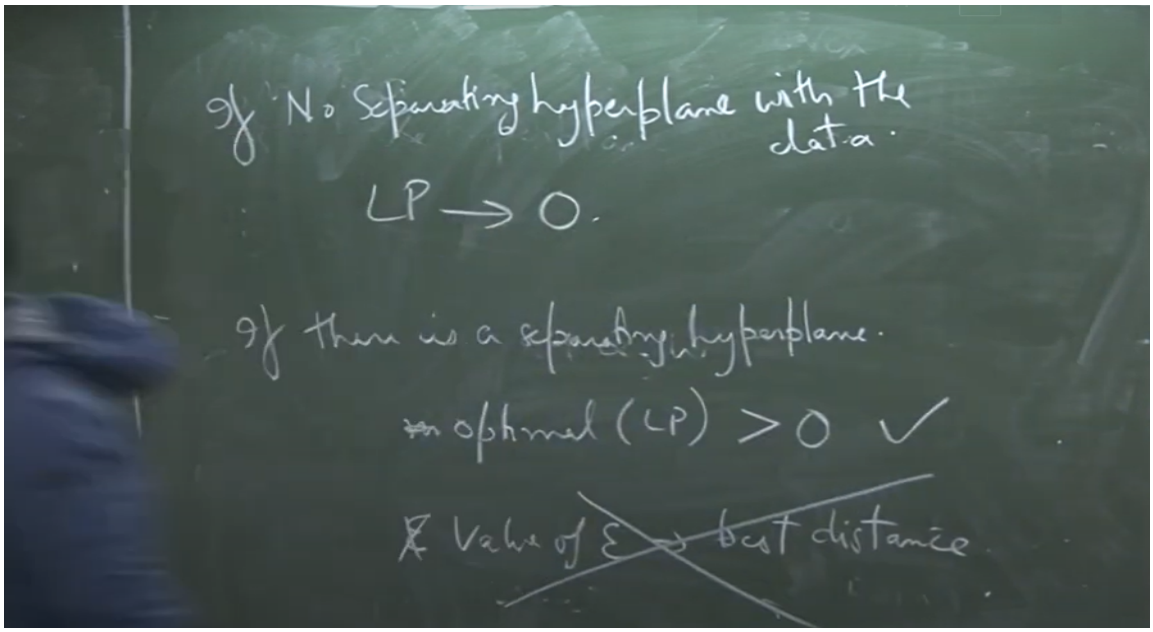
Again I am not notifying the notion of distance. Let us say it is some epsilon away. That basically means I can say that I want a hyperplane actually which maximizes this epsilon. It is like a really nice classifier. I do not want it to be here.



I do not want it to be here. I want the classifier which is in between all these points. So

I want to maximize alpha, maximize epsilon such that i to n minus epsilon is greater than equal to 0 and for i is equal to n plus 1 sorry I have k data points for I plus 1 to k is equal to 0 is less than equal to 0 and this is clearly a linear program. But what does it tell us about the classifier problem? This looks like it is solving the linear classifier problem. Let us go slightly into detail and see what it exactly does.

So first thing is and this is something which I am giving you as an easy exercise. If there is no separating hyperplane between the data, then this LP has value 0. Similarly I cannot have a value greater than 0. If there is no separating hyperplane, I cannot have a hyperplane such that all these quantities are positive and all these quantities are negative. So clearly my value will be less than equal to 0.

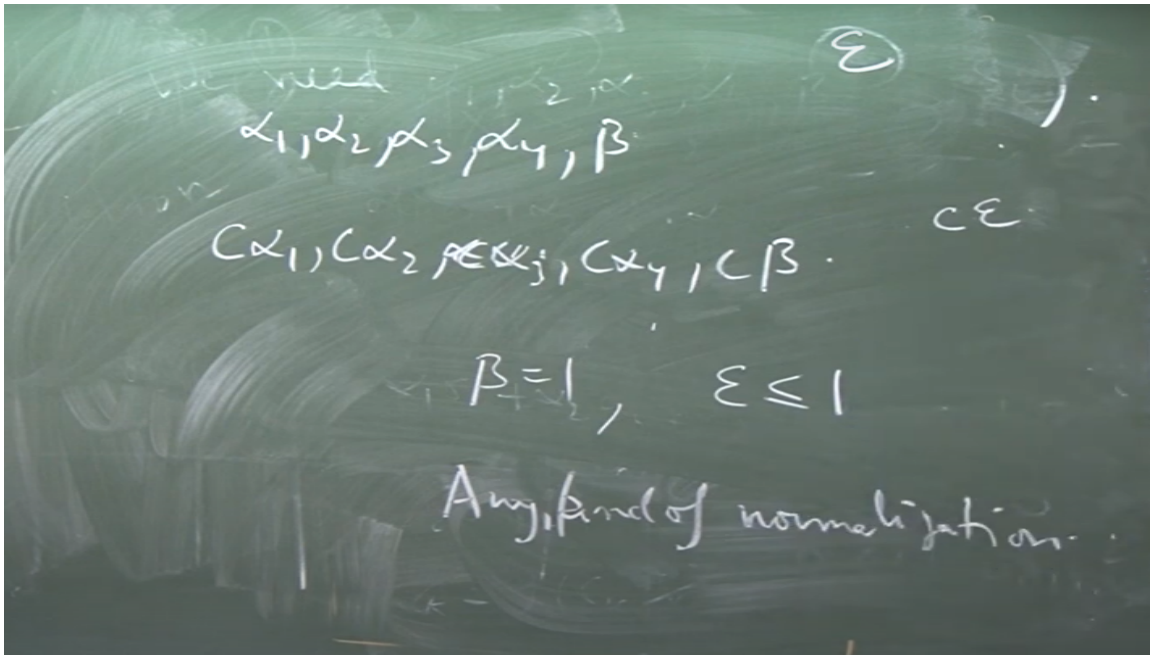


What you can show is that you can always have a hyperplane which has the exact value 0. This is the trivial case. On the other hand, if there is a separating hyperplane, what happens? Now you might say that in some sense epsilon will give the maximum distance possible or two epsilon will give the maximum distance possible between the two set of points. So once more the optimal of our LP will be strictly greater than 0, correct? And then intuitively you might say the value of epsilon gives us the best distance.

This is true. This unfortunately at the current form is not true. So let us see what is going wrong here. Unfortunately this hyperplane does not take care of scaling or this linear program does not take care of scaling. For example, let us say I have a hyperplane defined by these quantities; I get the distance to be epsilon. Alpha 1 to alpha 3, alpha 4 they are feasible solution of these equations and for I 1 to n this hyperplane is away by epsilon, for I n plus 1 to k it is below 0 by epsilon.

What about the same hyperplane with the modified constants? I can just take this is the

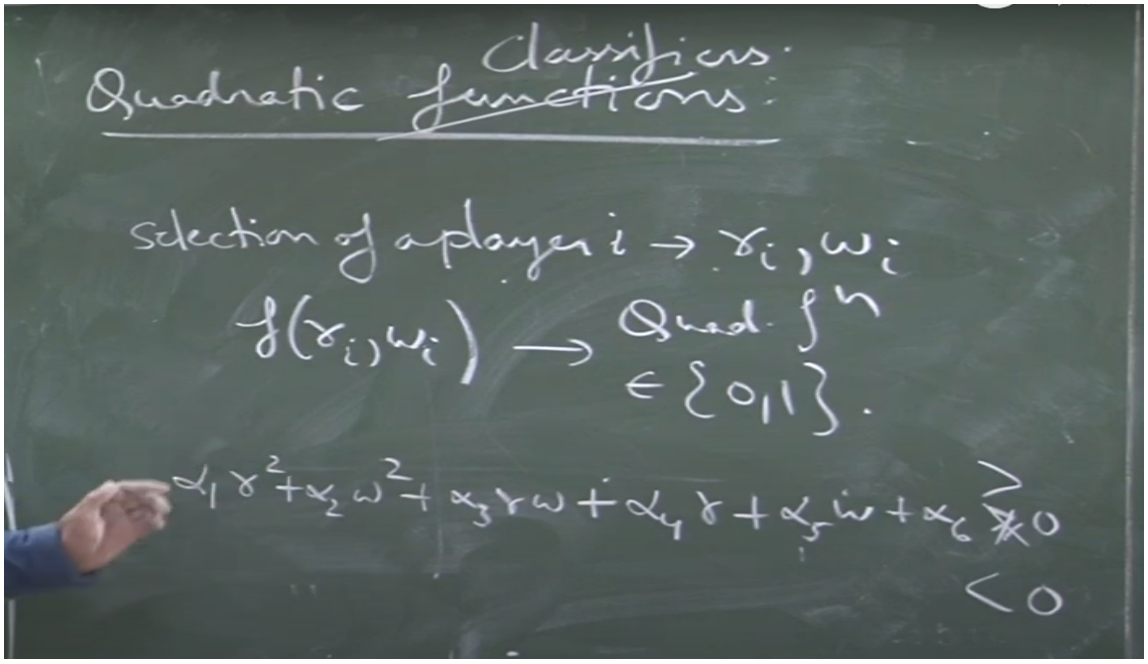
same hyperplane as before, but now its distance is going to be c epsilon. So what is going to happen is that if for this LP if there is no separating hyperplane we get LP the value as 0, this is an easy exercise. If there is actually a separating hyperplane my optimal will become unbounded, ok but that is not a big cause of worry if you want to figure out whether it is bounded or not and you do not want your LP to have unbounded value actually in your LP solution you can always stop as soon as you get a value higher than 0. In your simplex method, in your flip side method as soon as you have got a point where the LP has value more than 0 you know there is a separating hyperplane you can stop there. On the other hand if depending on the application if you really want your LP value to be not unbounded you can say either beta equal to 1, you can say epsilon less than equal to 1.



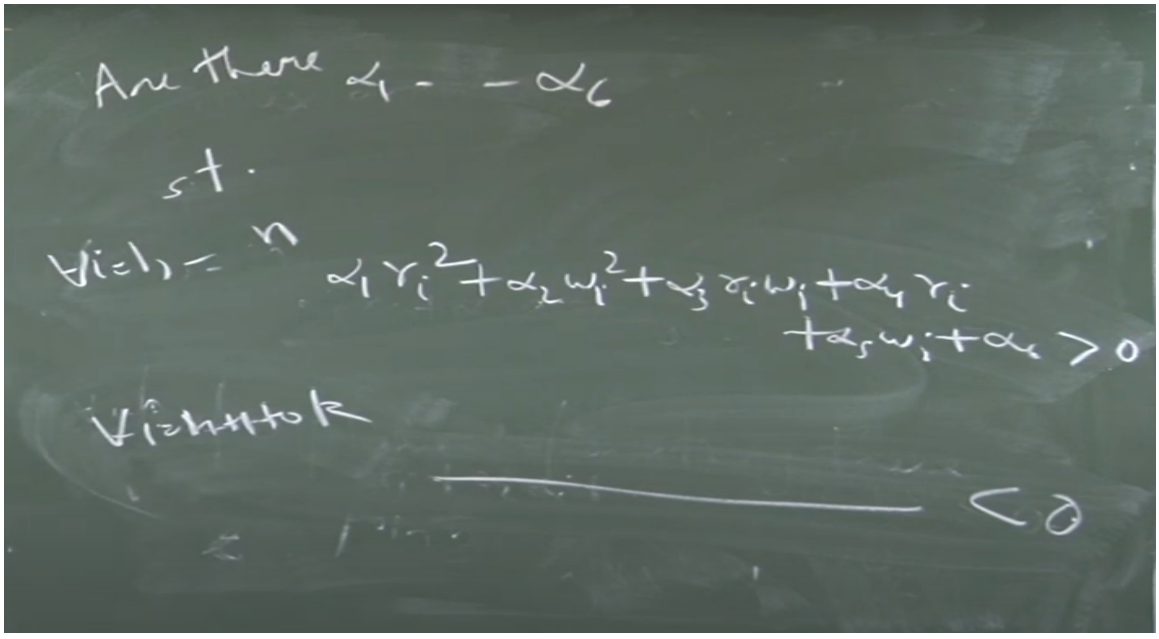
Any kind of normalization will make this value finite and it will give you some hyperplane best depending upon the equation here. Depending on beta equal to 1 you will get some notion of best hyperplane which separates these points for epsilon less than equal to 1 again you will get some notion so on and so forth. But definitely one thing is quite clear if you want to figure out is there a hyperplane separating these points these n points from these k minus n points this can be done by solving this linear program. If we get the optimal value to be 0 then there is no separate hyperplane if at any point we get a value more than 0 then we know there is a separating hyperplane. And then there are ways to make the LP value bounded.

It might seem like we can only classify using linear functions, but there is a small trick which can even extended to quadratic functions or quadratic classifiers. What does it

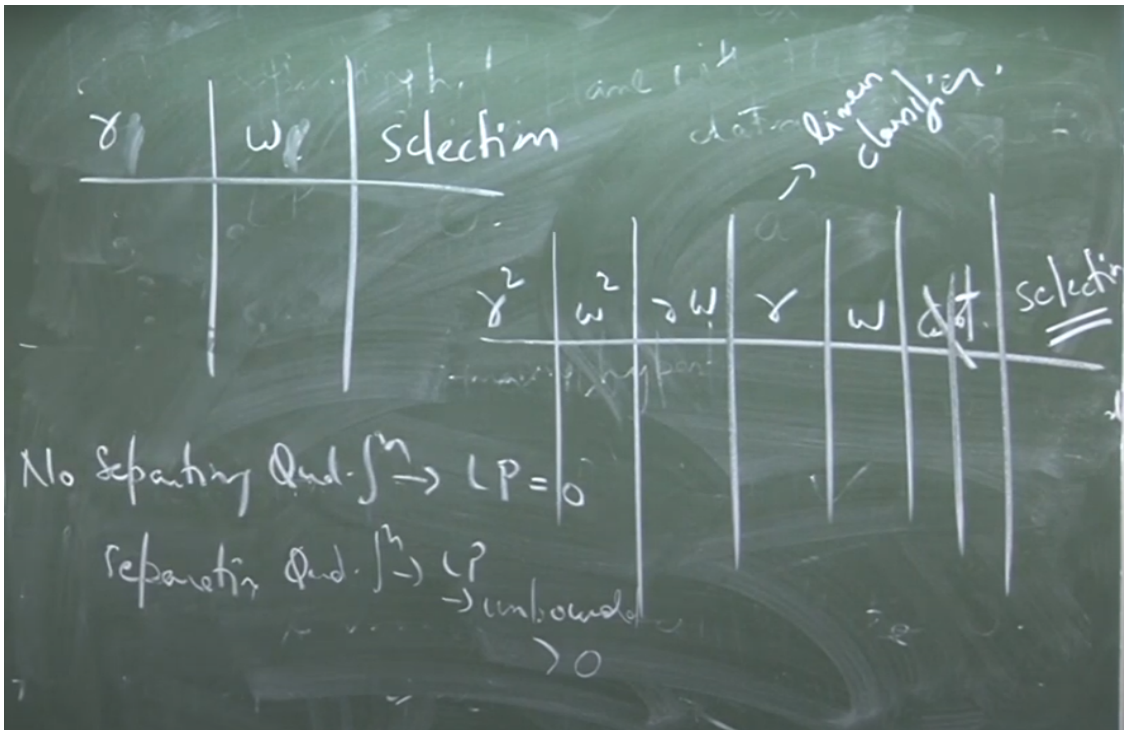
mean? Let us take a small dimension example because 4 dimension examples will be too big. Let us say the selection of a player depends on just 2 quantities number of runs score and number of wickets taken. And this is a quadratic function whose value is in 0 commas 1. What it means is that I have a function which is something like alpha 1 r square plus alpha 2 w square plus alpha 3 r w greater than equal to 0 I should say or less than equal to 0 right.



So, I am replacing a coefficient here r square w square r w these are the all possible quadratic or less than quadratic terms. It means there is there are some coefficients alpha 1 alpha 2 alpha 3 alpha 4 alpha 5 alpha 6 such that for selected points it should be greater than 0 for non selected points it should be less than 0.



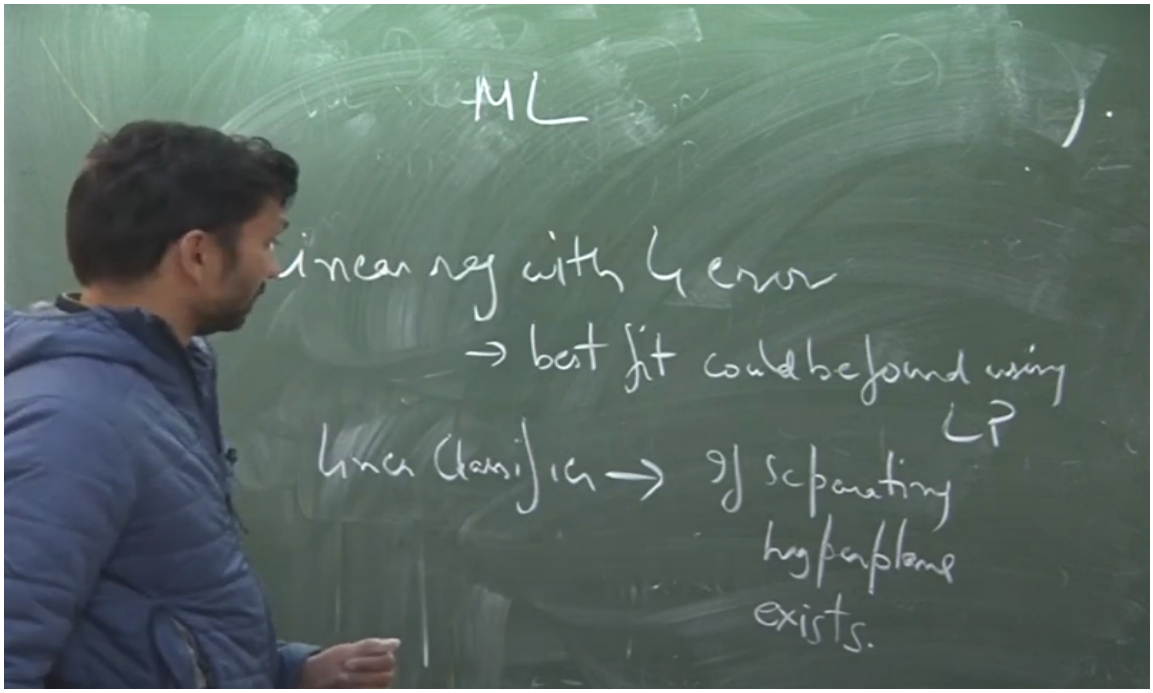
I already have the problem written here if I want to change it I want to write are there alpha 1 alpha 2 alpha 3 alpha 1 2 alpha 6 such that and a similar equation for n plus 1 to k this should be less than 0. The only thing we have to remember is that we need to increase the dimension of our data remember we were only given r i w i what we have done is that by increasing the dimension we have changed quartet classifier to a linear classifier problem. So, I should say r w selection and selection dependent upon these 2 as a quadratic function.



Now, we slightly changed it we said r square w square r w r w constant actually it is not worry about the constant this will be there in the linear classification also and there is a selection. This data forget about this data can be transferred into this table and once we have converted this data into this table now I have a linear classifier problem on this data. That means, I want to find a 6 dimensional hyper plane which satisfies this and we know how to do this we will say maximize epsilon less than equal to 0. Now, putting these I can solve this equation again the same things if there is no separating quadratic function I will get the LP value to be 0. So, no separating quadratic function LP value is 0 if there is a separating quadratic function LP value will be unbounded strictly more than 0.

So, this and again if you are uncomfortable with the unbounded value you can reduce some normalization to fix it and this is just for quadratic function, but if you think of cubic and everything by raising your dimension suitably all that can be handled. That means, we can solve linear regression quadratic regression all that with the linear program. To summarize for ML we saw 2 applications linear regression with L1 error this could be written as a linear program and we can solve we can find the best fit

according to the L1 error best fit could be found using LP for linear classifier problem. We know that we can find if separating hyper plane exists the separating hyper plane exist if my LP value is strictly more than 0 if my LP value is 0 then there is no separating hyper plane. So, this was just a flavor of many applications of linear programming in different domains you saw game theory you saw communication complexity now you saw machine learning this is definitely not the end of it just in computer science there are.



So, many applications you can even talk about complexity measures complexity theory and you can see many applications I encourage all of you to go to the internet find other applications of learn linear programming and learn about it.