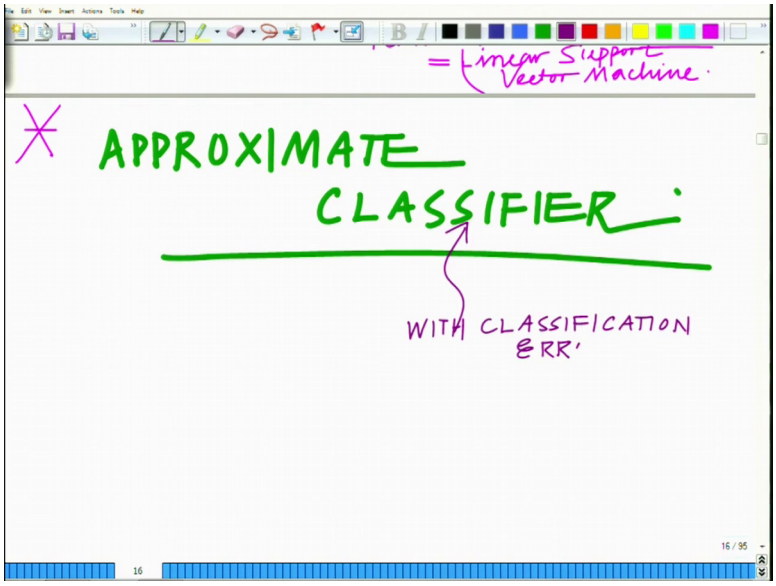
Applied Optimization for Wireless, Machine Learning, Big Data
Prof. Aditya K. Jagannatham
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture - 62
Practical Application: Approximate Classifier Design

Hello, welcome to another module in this Massive Open Online Course. We are looking at the linier classifier and we have seen in how the support vector machine for linier classification of 2 sets of points can be formulated confect optimization problem ok. So, in this module, let us continues you discussion; let us make this paradigm a little bit more sophisticated by incorporating also certain amount of classification error for Approximate Classification all right.

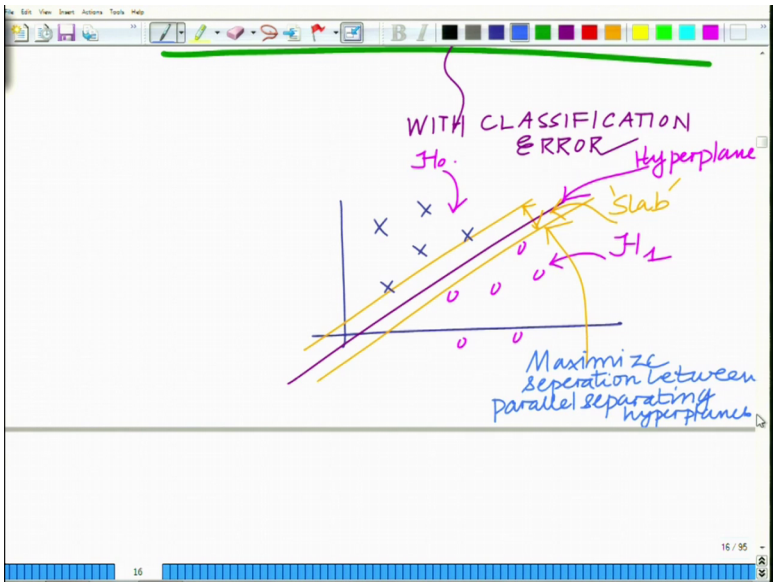
So, what we want to do is I want to explore the possibility of approximately classifying 2 sets of points ok.

(Refer Slide Time: 00:50)



And the reason for that I am going to explain at shortly. So, we have looked at so called hard or exact classification. So, let us now also look at building an approximate classifier and approximate classifier in the sense that there is some possibility for a classification so can incorporate. So, you can say this has with some classification error.

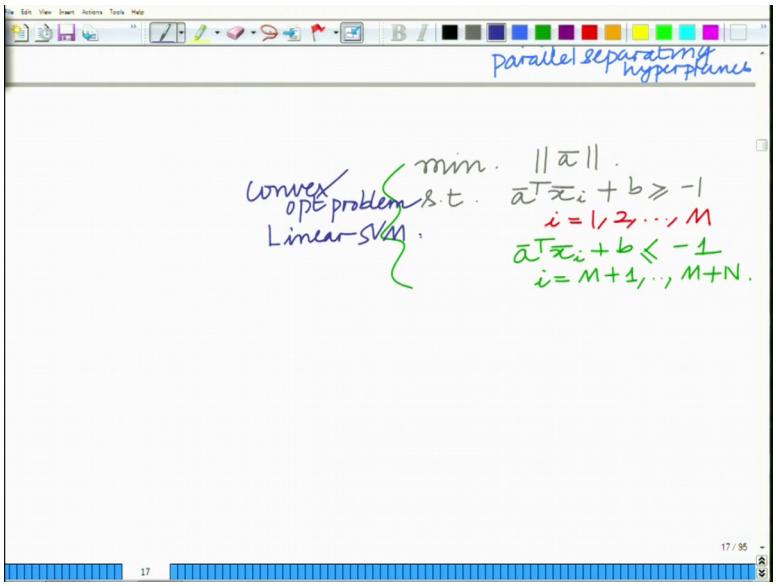
(Refer Slide Time: 01:34)



So, this allows for some classifications error and the reason for that is as follows. Now, so far what we have seen is basically the following, we have 2 sets of point and we are trying to separate them. So, this set correspondence to hypothesis is H 0 corresponds to hypothesis H 1 and we said we can separate them using a hyper plane. But we separate them simply using hyper plane that results in the trivial solution.

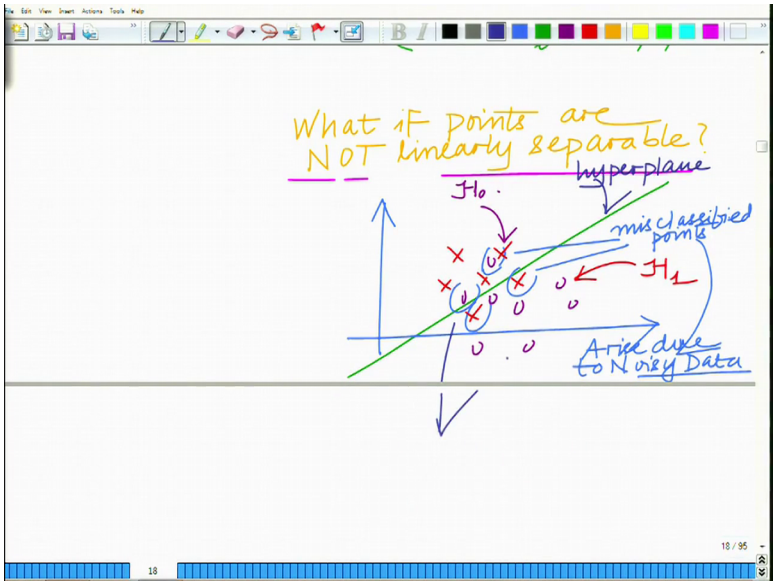
So, what we said was we are going to fit a slab, the thickest possible slab. So, we are going to fit a slab and maximize the separation or maximize the we would use 2 hyper planes and we are going to maximize the separation between the parallel separating hyper planes. Now, the problem with this is the following thing. Now, what we have said is the following thing. We have said the optimization problem for this can be formulated as follows.

(Refer Slide Time: 03:02)



I want to minimize remember the said the distance of separation is 2 over norm a bar. So, if you want maximize the distance we want to we have to minimize norm a bar subject to the constraints a bar transpose x bar i plus b greater than or equal to minus 1 and a bar trans this is for i equal to that is the first set of points i equals 1, 2 up to M and then, we want to have a bar transpose x bar i plus b less than or equal to minus 1 for i equals M plus 1 up to M plus N and this is you convex optimization problem for the linear SVM ok. This is what we have seen. This is for the linear. Now, the problem arises if what if the sets of points are not linearly separable.

(Refer Slide Time: 04:26)



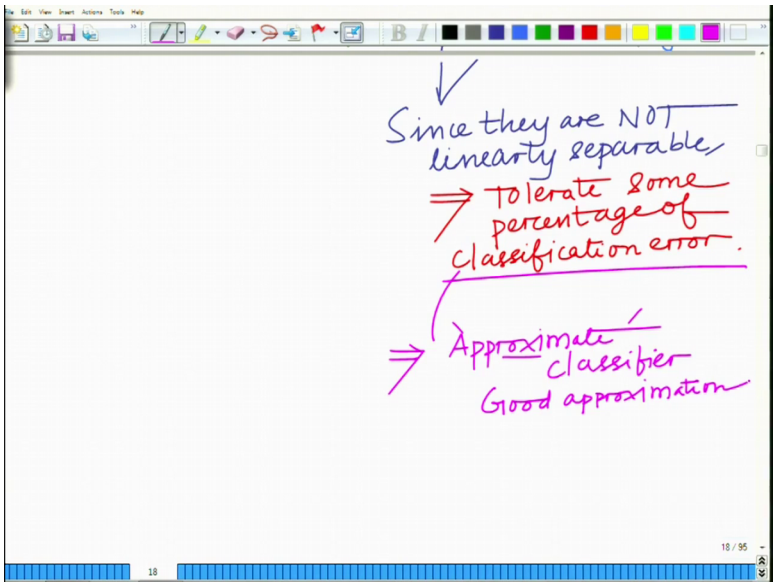
So, what if what happens are not what if they are not linearly separable. For instance, you have a situation where you have some points belonging to hypothesis H 0 and at the same time you have some points belonging to hypothesis H; so these are the H 0 points and these are the H 1 point and then, we can see there are 2 clusters. But if you try to separate them by any plane all right, if you try to separate them by any plane ok, you are going to have some classification errors.

So, if this is the hyper plane. So, you are going to have for instance, this point here this point here which are H 1 point classified as H 0 and this point here this point is the here. So, these are the misclassified points. So, these are your misclassified points or and these can also so and these can also arise due to the following. These can also and this is a very important point these can also arise due to noisy data.

So, what happens is normally this 2 clusters that is belonging to the primary user are absent and primary user present are well separated; sometimes because there might these is noise in the system and that is reality. In practice frequently, we have no observations are noise less; all the observation noisy which means some of the H 0 observations closely clustered with H 1 and some of the H 1 observation are closely clustered or cluster closer to H 0.

So, there is essentially there is an area where there is a kind of grey area where there is a mixing of the paired points corresponding to these 2 data set. So, in the sense, so in the sets that these 2 sets might not be linearly separable that it is not possible to find a plane or it is not possible to fit a nice slab between them. So, that the both the set of points lay on different sides of the slab; which means one has to tolerate. So, they are not linearly separable. So, one has to tolerate certain amount of classification error that is the point.

(Refer Slide Time: 07:24)

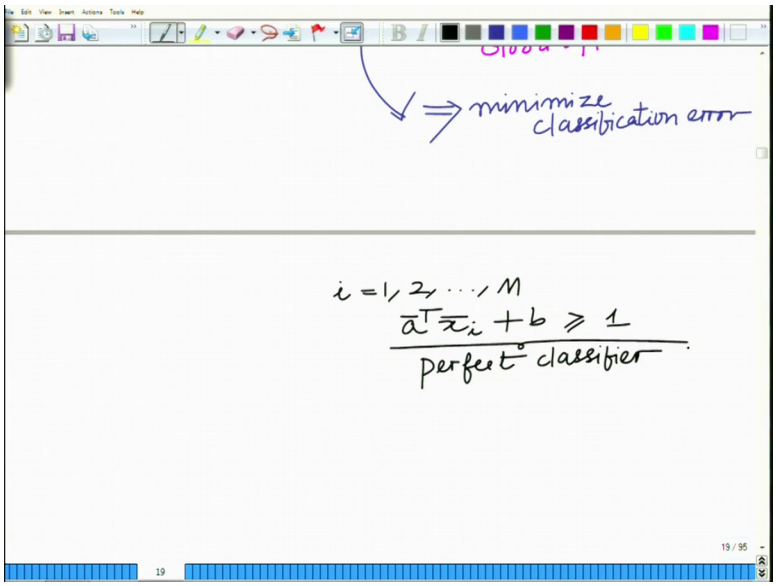


So, since they are not, since they are not linearly separable; since they are not linearly separable, this implies one has to tolerate some percentage of one has to tolerate some percent of classification error all right; which means basically there is not exact classification or hard classification, which implies that or classifier is only going to be approximate.

So, it is only going to be an approximate classifier, but we want to good approximation ok. So, as per as approximations go if you want to have an approximate classifier; then, one can draw any hyper plane all right. For instance, one can draw hyper plane that looks like this ok. But you can see or you can draw hyper plane that looks like this, but we can see although that is an approximate classifier that is bad approximation because it is roughly I mean that the data sets are not well separated all right.

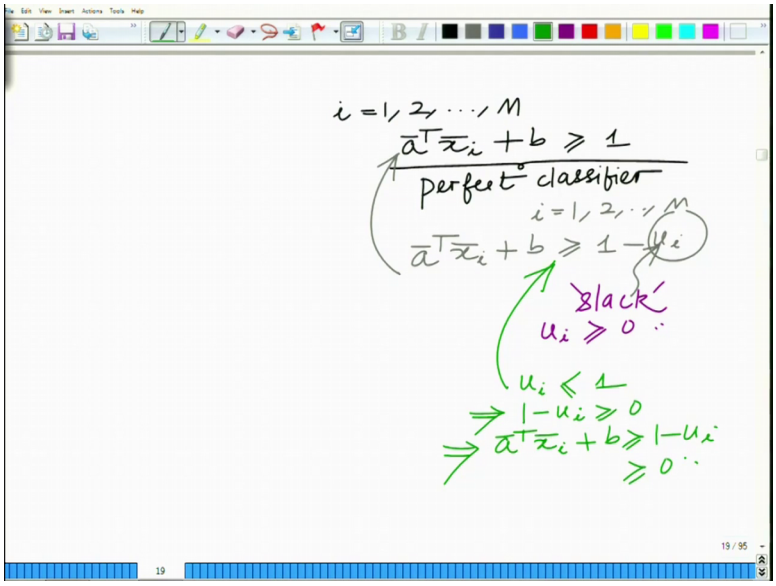
So, we want to design a approximate classifier, but in the same time approximate classifier that is that minimizes the you can say that minimizes the number of miss classified points or minimizes the classification error or that is very accurate all right to a very high degree. So, that what we say we want to design approximate classifier; but a good, a decent approximate classifier that gives you the best classification that is possible that minimizes the classification error that can be a good metric.

(Refer Slide Time: 09:28)



So, minimize you minimize the classification error; minimize the classification error ok. Now, how do we do this? We can do this follows. So, here remember consider the points i equal to 1, 2 up to M, we have the condition a bar transpose x bar i greater than equal to 1 or a bar transfers x bar i plus b greater than equal to 1. Remember this is for perfect classification let us say no, this is for a perfect classifier if perfect classification is possible.

(Refer Slide Time: 10:27)

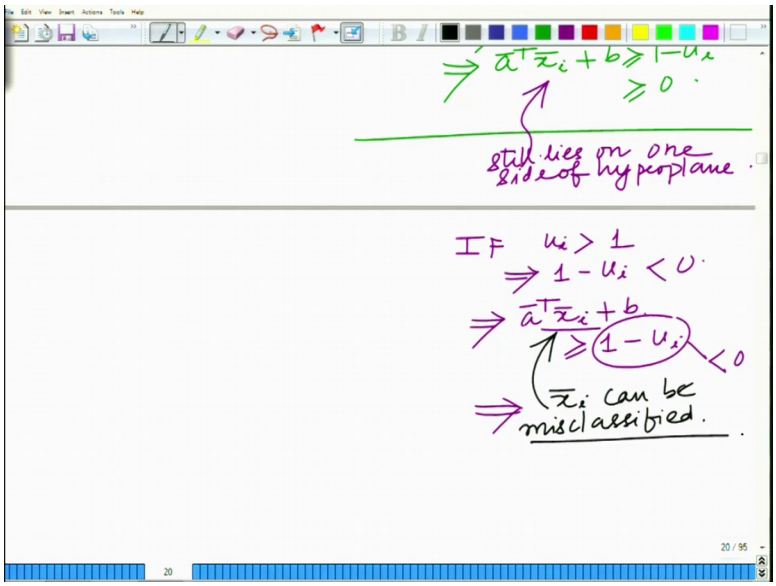


On the other hand, when this is not possible what we will do is we will modify this as follows; a bar transpose x bar i plus b greater than equal to not 1, but 1 minus u I, where this u i remember, we are still talking about i equals 1 2 up to M, where this u i is termed as a slack.

So, you allow for a certain slack in the constraint that is the constrain need not be exactly satisfied that is a bar transpose x bar i plus b is not greater than equal to 1, but some point for some i it can happen such that its greater than equal to 1 minus u I, where u i some kind of slack that you are allowing in this constraint all right. So, this u i has to be naturally greater than equal to 0 because if u i is less than equal to 0, then a bar transpose x bar i plus b is greater than equal to 1 minus u i which is greater than equal to something that is greater than 1 all right.

So, that is not necessary ok. As long as this is greater than equal to 1 or something that likely less than 1 that is also sufficient. Now, you can see this allows for some classification error because considered two cases one is u i is less than equal to one then this implies 1 minus u i is greater than equal to 0, this implies a bar transpose x bar i plus b greater than equal to 1 minus u i greater then equal to 0.

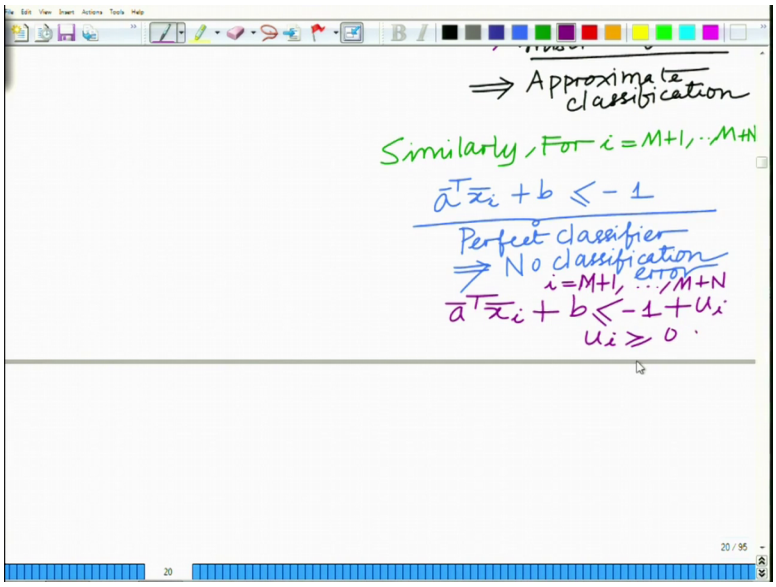
(Refer Slide Time: 12:10)



Which means it still lies on the other side of the hyper plane ok. Still lies on one side of hyper plane that is a bar transpose x bar i greater than equal to 0.

However, if u i is greater than 1, this implies 1 minus u i is less than 0 and this implies a bar transpose x bar i plus b which is greater than equal to 1 minus u i, but this quantity 1 minus u i is less than 0; which implies that a bar transpose to x bar i plus b is less than 0 or which implies that this constraint can be less than 0 which implies that this point can be misclassified this particular x bar i can be misclassified. So, in that sense this is only approximate.

(Refer Slide Time: 13:30)



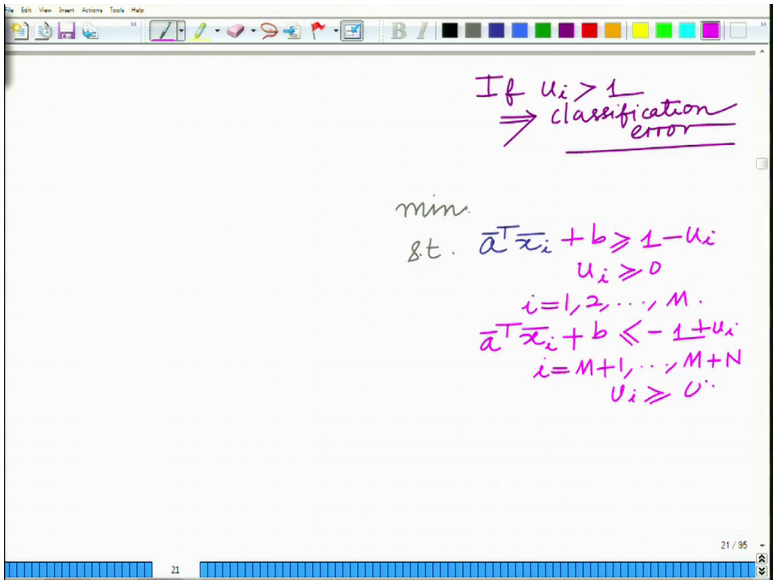
So, in that sense it is only approximate classification in the sense that you are allowing some of the point, you are allowing a slack in this constraint or basically you are allowing some of the points to be misclassified. In the sense that some of the points have slack that is large enough, so that they cross over one side of this hyper plane to the other side. So, some of the hypothesis is 0 points can be classified as hypotheses 1 point. So, you are allowing a certain slack or you are allowing the possibility for classification error.

Now, similarly when you are look at i equals similarly for i equals for i equals up to M plus 1 up to M plus N, then what happens? Remember for perfect classification we have a a bar transpose z bar i plus b less than equal to minus 1. This is for your perfect classification. This is when your classifier, this is for a perfect classifier that is low classification error implies that there is no implies that there is no classification error.

However, there is classification error; even there is when you want to allow the possibility of point being misclassified, again due to noisy data perfect separation is not possible. Then, you can again allows some slack a bar transpose x bar i plus b is less than or equal to minus 1 plus u i ok. Remember we still talking about i equals M plus 1 up to M plus N, where u i greater than or equal to 0 ok.

So, u i is greater than or equal to 0. So, u i less than equal to 1 again, it still has to the other side of the hyper plane; If u i is greater than 1, then you have a problem in the sense (Refer Time: 16:03) cross the over.

(Refer Slide Time: 16:05)



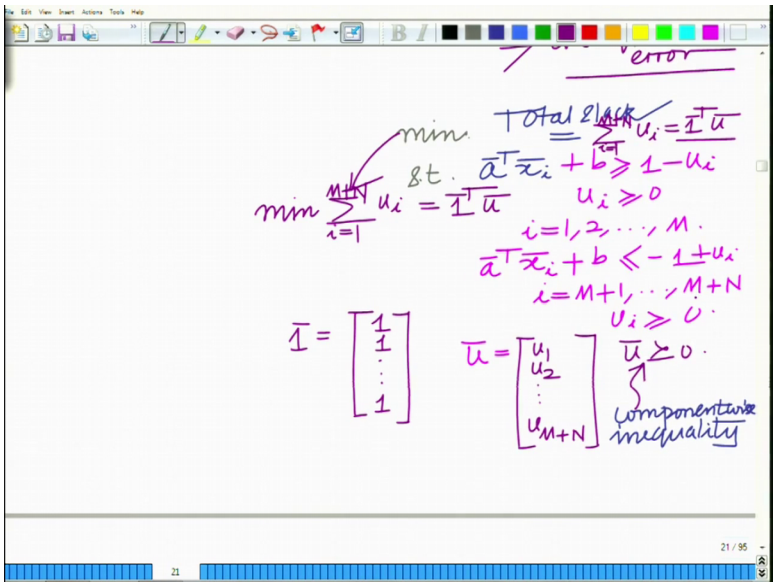
So, if u i greater than 1, this implies there is a classification error. So, by introducing the slack in these constraint, your allowing the possibility that you are recognizing the possibility the data indeed noisy and we allowing for from slack in this constrain that is the possibility allowing the possibility force of you few of the point to be misclassified.

So, therefore now, the optimization problem can be formulated as. Now, remember, we have so for not talked about the objective; remember, you want to build an approximate classifier; but you want to build the best approximate classifier. Now, what is the best of part of it that we will come to shortly?

So, but we have the constraints. So, what are the constraints now? The constraints now a bar a bar transpose x bar i. So, we have the constraints a bar transpose x bar i plus b

greater than equal to 1 which was there previously; now equal to 1 minus u I, u i greater than equal to 0; i equals 1 2 up to M ok. So, now, you have this new constraints on u i. All the u is are non negatives. Similarly, for the other set of x bar i; a a bar transpose x bar i plus b less than or equal to minus 1, i equal to M plus 1 up to M plus or and previously was lesser equal to minus 1. Now, lesser equal to minus 1 plus u i again u i greater than or equal to 0.

(Refer Slide Time: 18:00)



And now, in fact, if you write this as a vector u bar, you can write this as a comprehensive constrained u bar equals u 1 u 2 up to u M plus N; you can combine all this things and say u bar greater than equal to 0. This is the component wise inequality all right. In each component of u bar must be greater than or equal to 0. So, this is your component wise inequality and here, you can simply minimize the total slack.

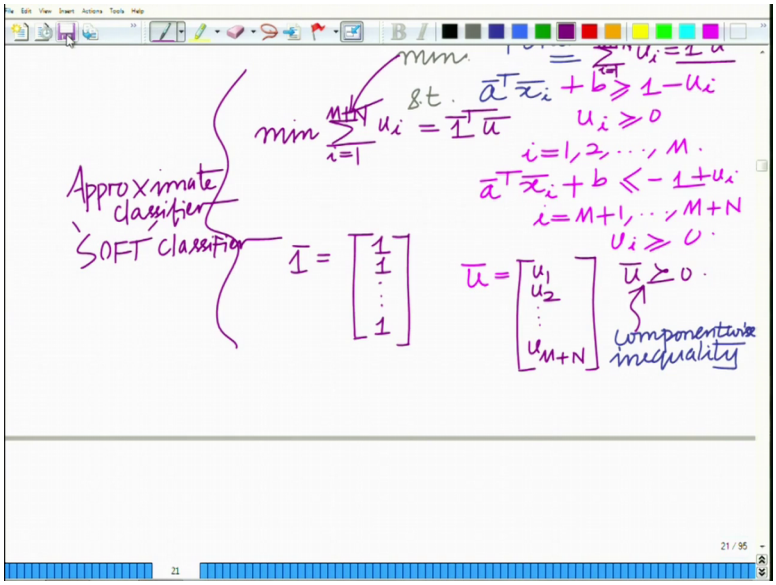
Now, what is the best approximate the classifier? Best approximate the classifier is basically the one which minimize the total slack that is remember we doing the slack because you are helpless right. But at the same time, we do not want to allow too much of slack or too much of tolerance, we want to design we want to keep the tolerance to as low as possible which means we have to minimize the slack or in other word we can minimize the total slack ok.

So, I can minimize the total slack summation i equals 1 to M plus N u i which I can write as 1 bar transpose u bar, where 1 bar is simply the M plus 1 vector of this is the M plus 1

vector of all ones ok. So, that is your objective function; minimum 1 bar transpose u bar. Let me write this again clearly. Minimize summation i equal to 1 to M plus N u i which is nothing but the summation of all components of the vector u bar which can be written as minimized 1 bar transpose u bar all right. Remember each component of u bar each u i is non-negative that is u bar component wise greater than equal to 0 ok.

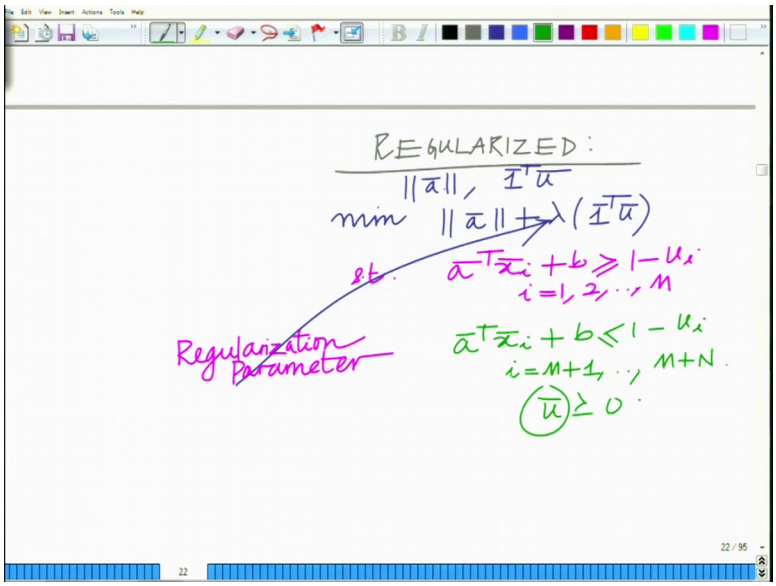
So, that is basically this is basically or approximate classifier or you can also think as the soft margin classifier and so on.

(Refer Slide Time: 20:21)



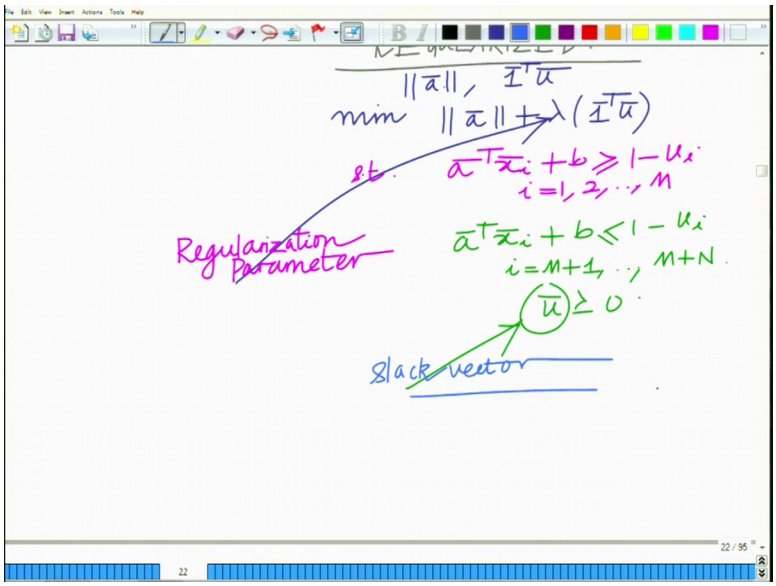
So, basically this is approximate or this also can be if the previous one is the hard classifier, this can also be known as a soft classifier; in the sense that it is soft that is not, it is not a hard partition between these, it is the soft one its porous one through which some points can pass through all right, but it is not the same thing as have no partition at all which is basically where all points are all either (Refer Time: 20:00). So you try to build the best approximate classification and now, you can extend this to a regularized or so, you can regularize this.

(Refer Slide Time: 21:12)



Previously, we had we wanted to fit the thickest slab. So, there are 2 objective functions a bar and one that minimizes the slack, you can consider a combination of them. So, norm a bar plus lambda times 1 bar transpose u; you are aware of this lambda similar to many optimization problems. This is the regularization, this is the regularization parameter. Constraints are the same subject to a bar transpose x bar i plus b greater than equal to 1 minus u i, where i equals 1 2 up to M; a bar transpose x bar i plus b less than 1 minus u i for i equals M plus 1 up to M plus N and u bar component wise is component wise greater than equal to 0. Remember u bar this is the slack vector, each component of that refers to the represents the slack.

(Refer Slide Time: 22:43)



Now, if linear classification is possible, then naturally these components of the vector, this component of the vector u bar will either the 0 or close to 0 all right, similar to what we have seen before. But of course, if linear classification not possible, then role of the use will be greater than 0. In fact, some of these might even be greater than equal to 1 which shows that basically some of points are misclassified.

However, we want these elements to be as few of them to be greater than equal to 0 as possible. That is you want the slacks in general to be as low as possible that is why we are minimizing the total set. In fact, in this case we are minimizing a weighted combination of the distance between the separately high preference plus the slack.

So, that fits the thickest slack, while allowing certain amount of misclassification all right and you are minimizing linear combination of these two objective functions, alright. This is basically the regularized minimization or you think of this as a regularized classify alright. So, we will stop here and continue in the subsequent module, so.

Thank you very much.