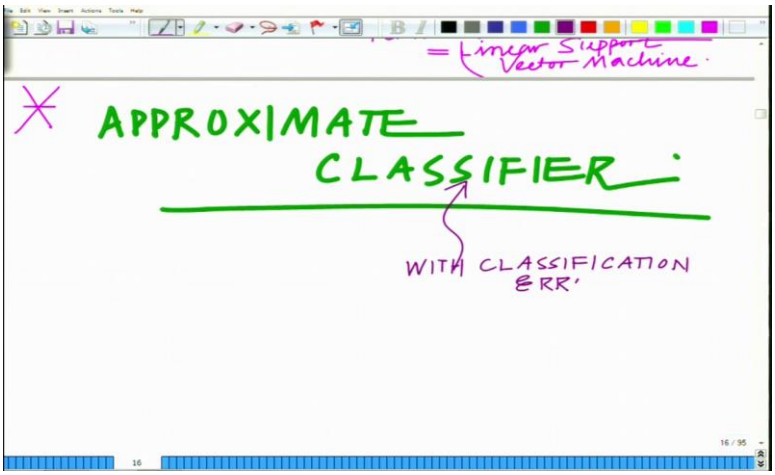
Applied Optimization for Wireless, Machine Learning, Big Data
Prof. Aditya K. Jagannatham
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture - 62
Practical Application: Approximate Classifier Design

Keywords: *Approximate Classifier Design*

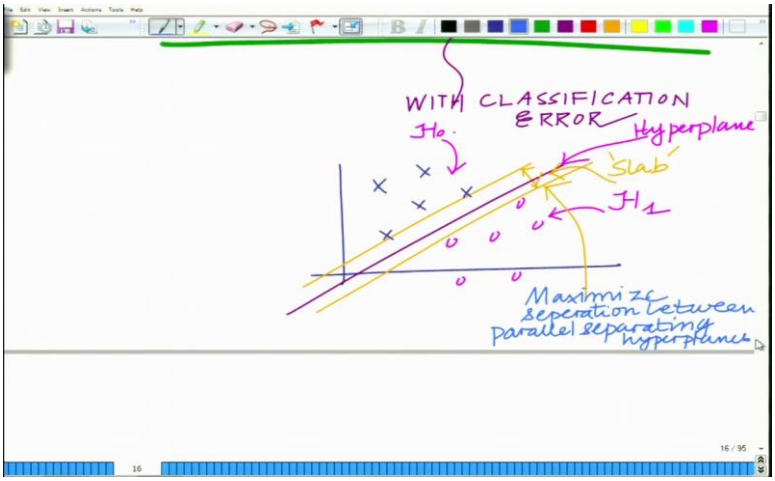
Hello, welcome to another module in this Massive Open Online Course. We are looking at the linear classifier and we have seen how the support vector machine for linear classification of two sets of points can be formulated in a convex optimization problem. So in this module, let us explore the possibility of approximately classifying two sets of points.

(Refer Slide Time: 00:50)



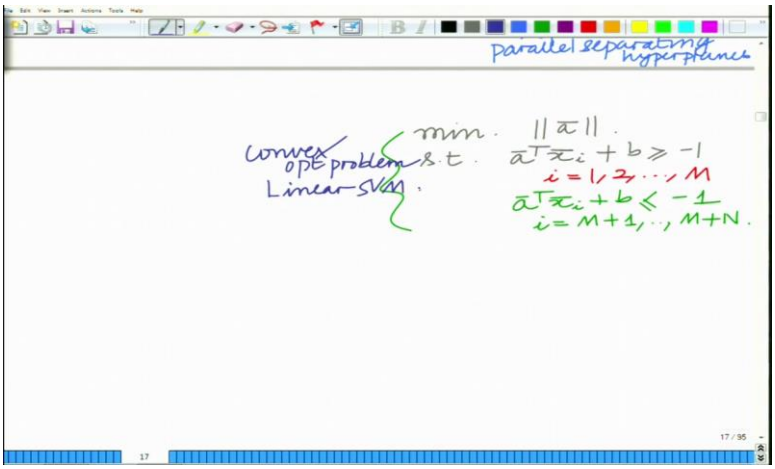
So let us look at building an approximate classifier in the sense that this has some classification error.

(Refer Slide Time: 01:34)



So far we have seen two sets of points and we are trying to separate them. So this set corresponds to hypothesis H_0 and hypothesis H_1 as shown in slide and we said we can separate them using a hyper plane. But that results in the trivial solution. So what we said was we are going to fit the thickest possible slab and maximize the separation or between the parallel hyper planes. The optimization problem for this can be formulated as follows.

(Refer Slide Time: 03:02)



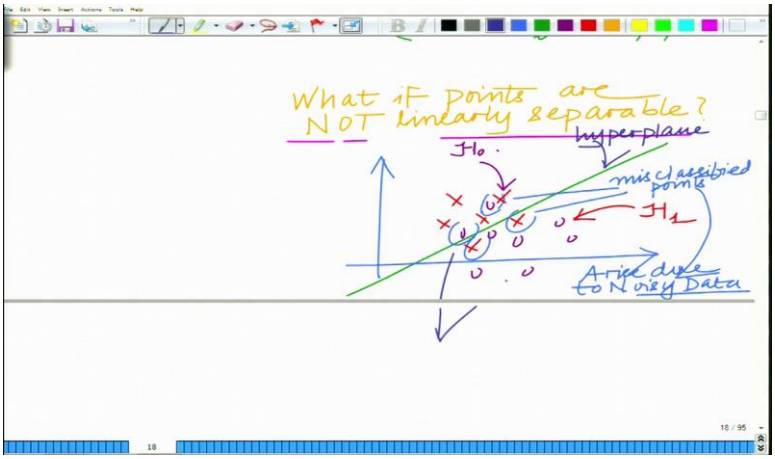
We said the distance of separation is $\frac{2}{\|\bar{a}\|}$. So if you want maximize the distance we want

$$\min \quad \|\bar{a}\|$$

to have $s.t \quad \bar{a}^T \bar{x}_i + b \geq -1 \quad i = 1, 2, \ldots, M$ and this is your convex optimization problem for the $\bar{a}^T \bar{x}_i + b \leq -1 \quad i = M+1, .., M+N$

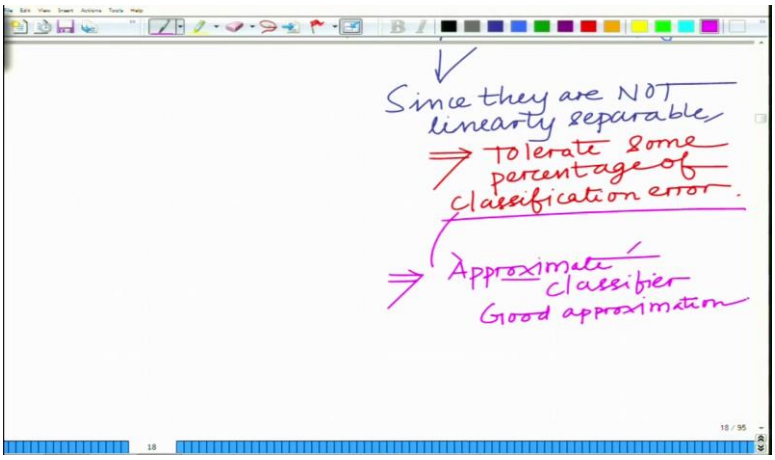
linear SVM and now, the problem arises if the sets of points are not linearly separable.

(Refer Slide Time: 04:26)



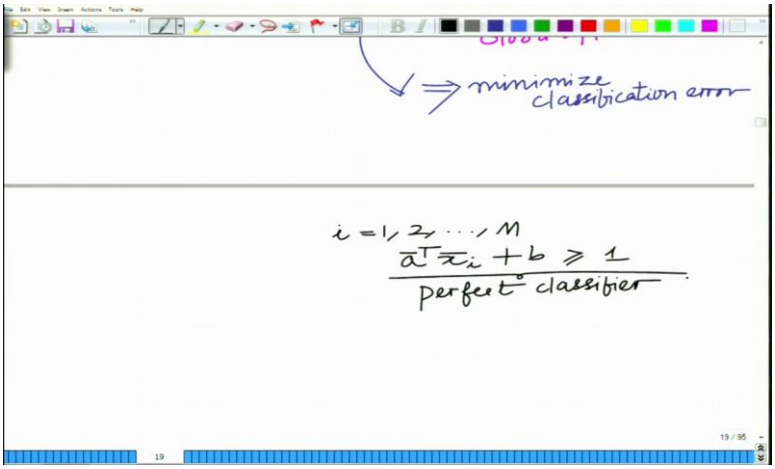
For instance, you have a situation where you have some points belonging to hypothesis H_0 and at the same time you have some points belonging to hypothesis H_1. But if you try to separate them by any plane, you are going to have some classification errors. So you might get misclassified points and these can also arise due to noisy data. So sometimes there might be noise in the system and some of the H_0 observations are closely clustered with H_1 and some of the H_1 observations are closely clustered with H_0. So it is not possible to find a plane or it is not possible to fit a nice slab between them which means one has to tolerate a certain amount of classification error.

(Refer Slide Time: 07:24)



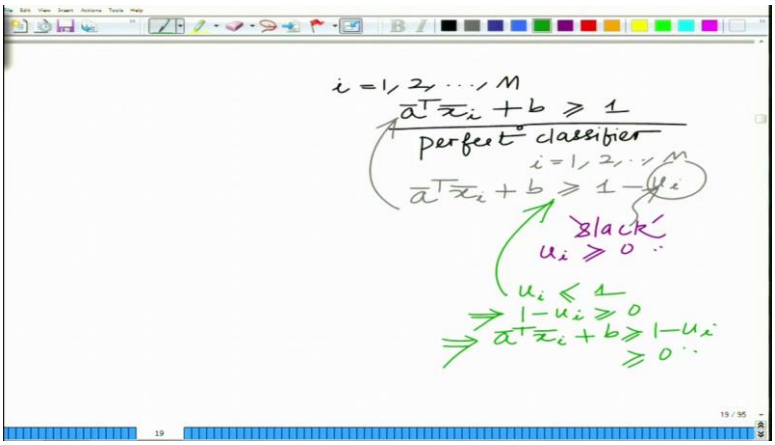
So this implies that our classifier is only going to be approximate. So it is going to be an approximate classifier, but we want to have a good approximation. So we want to design an approximate classifier that minimizes the number of misclassified points or minimizes the classification error.

(Refer Slide Time: 09:28)



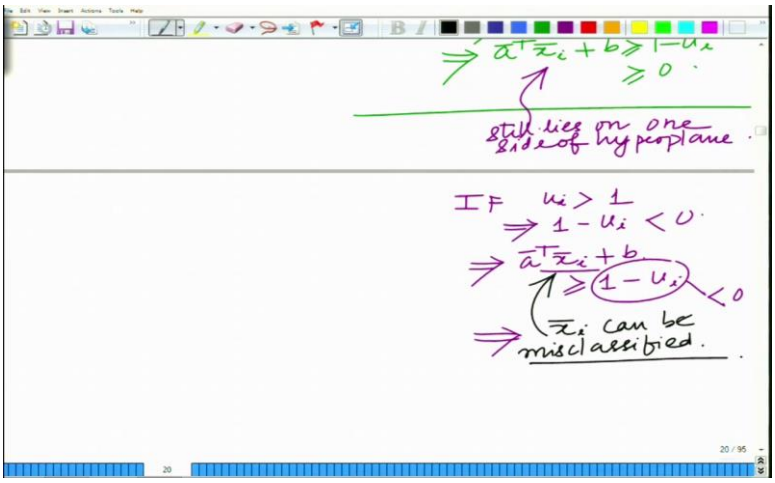
So this is given as shown in slides.

(Refer Slide Time: 10:27)

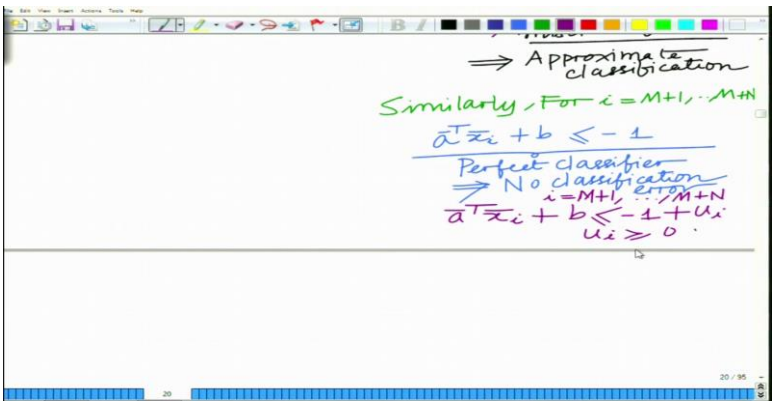


So you allow for a certain slack in the constraint that is the constraint need not be exactly satisfied.

(Refer Slide Time: 12:10)

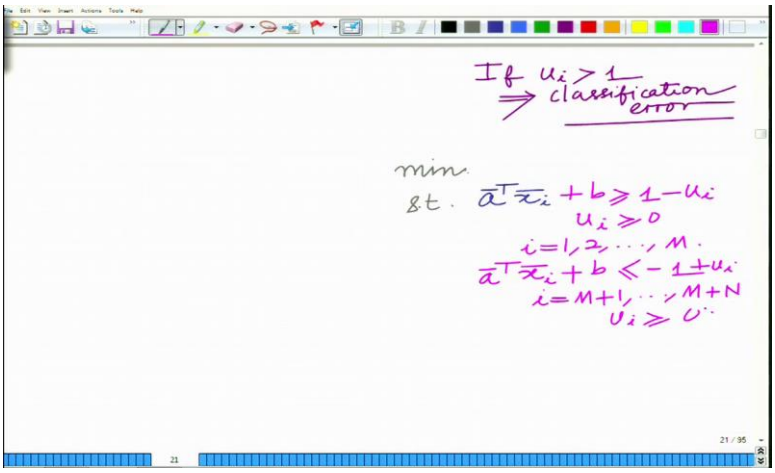


(Refer Slide Time: 13:30)



So 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. Now, for a perfect classifier there is no classification error. However, there is classification error, even when you want to allow the possibility of point being misclassified, again due to noisy data perfect separation is not possible.

(Refer Slide Time: 16:05)

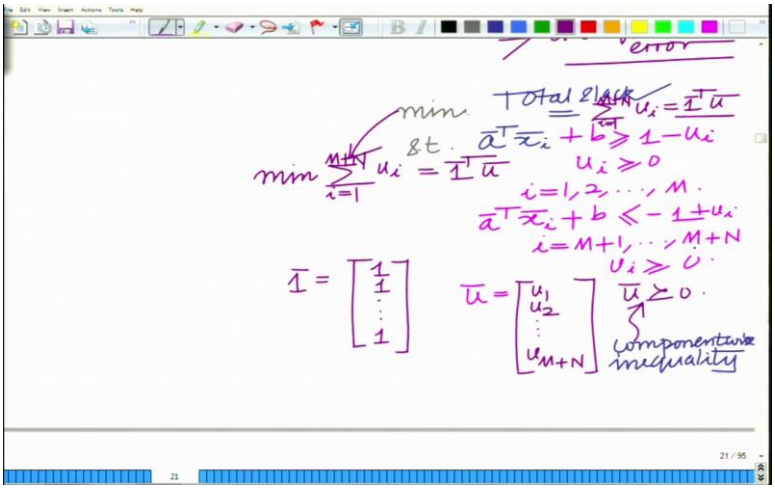


So by introducing the slack in these constraints, you are allowing the possibility for few of the points to be misclassified. So you want to build an approximate classifier, but the best approximate classifier. So we have

$$\min \quad \left\| \bar{a} \right\|$$
$$s.t \quad \begin{aligned} &\bar{a}^T \bar{x}_i + b \geq 1 - u_i \\ &i = 1, 2, \ldots, M \\ &u_i \geq 0 \\ &\bar{a}^T \bar{x}_i + b \leq -1 + u_i \\ &i = M+1, \ldots, M+N \\ &u_i \geq 0 \end{aligned}$$

.

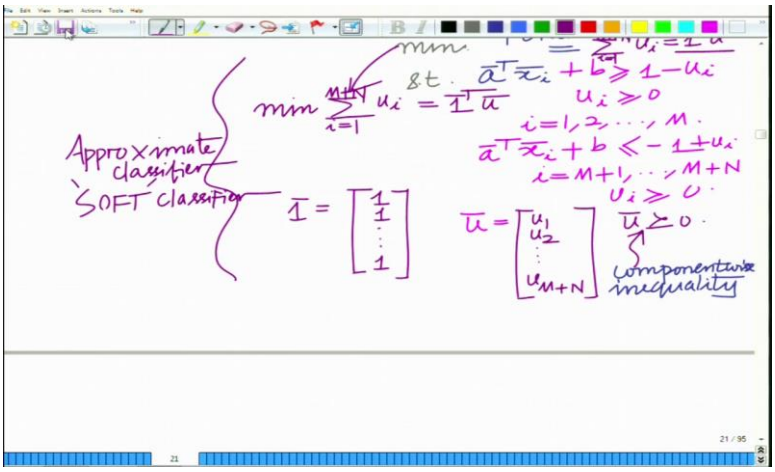
(Refer Slide Time: 18:00)



And now, if you write this as a vector you can combine all these things and this is the component wise inequality. So you can simply minimize the total slack. Now, the best approximate classifier is basically the one which minimizes the total slack at the same time, we do not want to allow too much of slack or too much of tolerance, we want to keep the tolerance as low as possible which means we have to minimize the total slack.

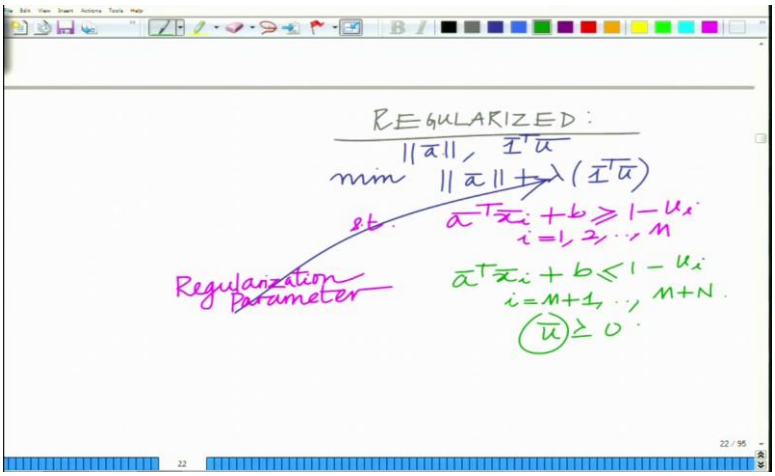
So this is given as $\min \sum_{i=1}^{M+N} u_i = \bar{1}^T \bar{u}$ $s.t$ $\bar{a}^T \bar{x}_i + b \geq 1 - u_i$ $i = 1, 2, \ldots, M$ $u_i \geq 0$ $\bar{a}^T \bar{x}_i + b \leq -1 + u_i$ $i = M+1, \ldots, M+N$ $u_i \geq 0$. So that is basically the approximate classifier or you can also think as the soft margin classifier.

(Refer Slide Time: 20:21)



So you can regularize this.

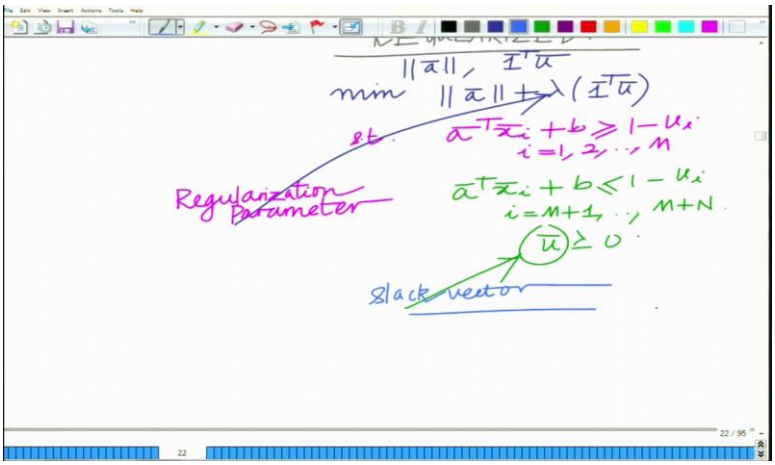
(Refer Slide Time: 21:12)



Previously, we wanted to fit the thickest slab. So there are two objective functions and you can consider a combination of them. So we use the regularization parameter λ , and this is given as

$$\begin{aligned} \min \quad & \left\| \bar{a} \right\| + \lambda \left(\bar{1}^{T} \bar{u} \right) \\ s.t \quad & \bar{a}^{T} \bar{x}_i + b \geq 1 - u_i \\ & i = 1, 2, \ldots, M \\ & \bar{a}^{T} \bar{x}_i + b \leq -1 + u_i \\ & i = M+1, \ldots, M+N \\ & \bar{u} \geq 0 \end{aligned}$$

(Refer Slide Time: 22:43)



Now, if linear classification is possible, then these components of the vector $\bar{u}$ will either be 0 or close to 0. But of course, if linear classification is not possible, then several of these u's will be greater than 0, in fact, some of these might even be greater than or 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 or equal to 0 as possible that is you want the slacks in general to be as low as possible, as close to 0 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 separating hyper planes along with the slack. So that fits the thickest slack, while allowing certain amount of misclassification and you are minimizing the linear combination of these two objective functions. This is basically the regularized minimization or you think of this as a regularized classifier. So we will stop here and continue in the subsequent module, so thank you very much.