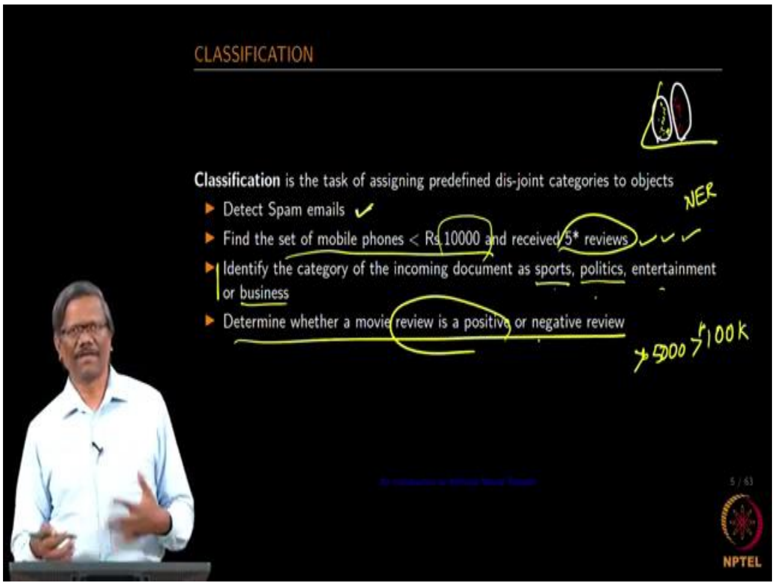
Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 31
Linear Models for Classification

(Refer Slide Time: 00:15)



CLASSIFICATION

Classification is the task of assigning predefined dis-joint categories to objects

- Detect Spam emails ✓
- Find the set of mobile phones < Rs 10000 and received 5* reviews ✓ ✓ ✓
- Identify the category of the incoming document as sports, politics, entertainment or business
- Determine whether a movie review is a positive or negative review

The first task as I mentioned earlier is going to be the classification task ok. So, we need to define what classification is, we have done this earlier. You remember we have taken the BBC data and then try to figure out what kind of topics that we have in the corpus right. In this case again ah we are going to be talking about the classification with respect to machine learning. But before that you know I will just give you some little definition related to that which I have not done earlier. And then move into the perceptron and then see how perceptron can be used to classify few things ok. So, we know very well that at the task of assigning predefined, dis-joint categories to objects right.

So, given a set of objects; so I want to classify that this is a different class and this has a different class right, it is a very simple explanation. So, we want to detect spam emails. So, this something all email boxes have implemented. So, we should understand what a spam email is in order for us to really identify whether it is a spam or a mail that you really want to read ok.

So, there are certain examples and parameters that are available for you to distinguish between the spam and the no spam email. So, one of the tasks that you want to perform is, this is something that we do very often right. So, we are very good in terms of replacing mobile phones every two years right so.

So, we keep looking at the newer models and then try to figure out based on the budget, which could be the best within that budget and then what kind of reviews you have seen for those and then based on the reviews based on you on your friend's reviews and so on you finally, pick upright.

So, this is something which can also be called as a classification problem; I want to find all the mobile phones which are less than 10000 rupees and received 5-star reviews. So, my budget is only this I do not want to go beyond that, but I want to do within this pick up 5 or 6 phones and then go through all the reviews and then find them.

So, this is something that we want to perform. So, in this case, what happens there are E-commerce sites that provide you all the details in terms of you know click click click and then you get to this and then you also have reviews related to that, there are just 5-star reviews without any text in related to that mobile phone. And there are reviews are which are written in English or you want to read them and then find out what some unknown person said about the review. So, this is again a classification task that we want to do.

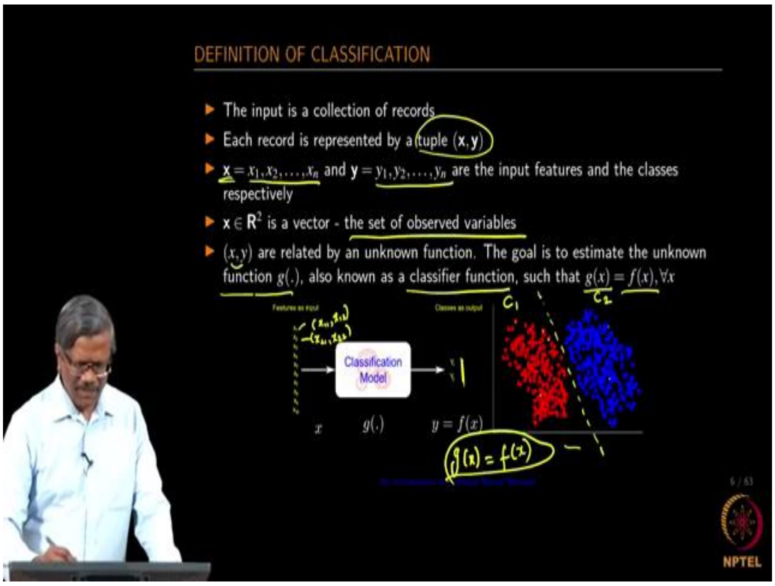
So, suppose if you are given a given about 1GB of review data right, which contains reviews about all phones in English text with respect to the mobiles ah, with respect to the features of the mobile with respect to the price all that in all English text; it contain data from letting us say 5000 to let us say sorry 100k for and then it is a huge pile of text and you want to classify that with respect to the reviews in terms of 5-star reviews or you want to classify that with respect to the price or you want to classify that with respect to the manufacturer and so on right. So, this is something that you can perform using the classification tasks.

So, what is that you require; you require some kind of a corpus that contains the reviews about all the phones; you should be able to extract certain features set from the corpus and feed that as input and finally, classify that based on what is asked ok. So, this is going to be using some kind of a NER. So, NER will extract the features for you and then using the features you can classify ok.

This something with that we saw as I mentioned earlier of using some corpus where we have data related to sports, politics, entertainment or business. So, when you have taught the system to recognize these based on the corpus when a new data is coming in you want to find out whether it belongs to any one of these right; and then you want to figure out whether a review of a movie is positive or negative.

Now, without really looking at the dashboard, you want to really go read some reviews you know some people are very good at writing a review. So, you want to really go read that and then find out what that person has said about that and so on ok. Apart from that you want to classify that based on what the review is all about its positive or negative. So, this is another type of classification that you would do in the natural language processing ok.

(Refer Slide Time: 06:48)



DEFINITION OF CLASSIFICATION

- The input is a collection of records
- Each record is represented by a tuple $(\mathbf{x}, \mathbf{y})$
- $\mathbf{x} = x_1, x_2, \ldots, x_n$ and $\mathbf{y} = y_1, y_2, \ldots, y_n$ are the input features and the classes respectively
- $\mathbf{x} \in \mathbf{R}^2$ is a vector - the set of observed variables
- (x, y) are related by an unknown function. The goal is to estimate the unknown function $g(.)$, also known as a classifier function, such that $g(x) = f(x), \forall x$

So, let us give some simple definitions for classification. So, supposing you know look at the figure at the bottom here right. So, there are two classes one with the blue squares; and then another one with red circles; it is very easy to draw a line; oops and then say there are two classes you know usually we can figure out that but supposing if we have gotten the data, you know with respect to all those points you know sequentially right, so where the points are mixed.

So, we want the system to really draw a boundary and then tell us whether this point belongs to class 1 or class 2. Let us assume that this is class 1 this is class 2; and then how will it really draw the boundary and then say that anything to the left of the boundary is class 1 anything to the right of the boundary is class 2 ok.

So, for that we require two things; one is for the training sake we require the point and then we require to which class it belongs right. So, what we have to figure out it based on the tuple that we have, we need to find out this light ok. Assuming that we have two-dimensional points so, the line separating or the plane separating these two classes is a 1-D line.

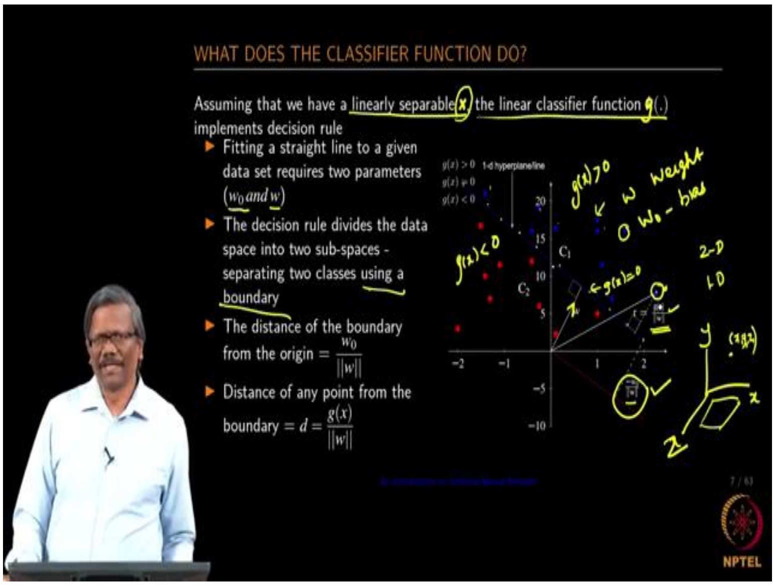
So, again what we know here as I mentioned earlier, we know the set of points x 1. So, this could be $x = \ x_1, x_2, x_3, \dots \dots x_n$ ok; each one is like this it is a two dimensional it is a point right. So, we have so many inputs which we are giving as input; and then we know to which class each point belongs to and we have to estimate this model right. So, this is one of the tasks that we keep doing in the machine learning ok.

So, here we have to create a model, when you create the model and when you are done with the estimation of the model it if you use g of x will also provide ok. So, this is your ideal model building exercise ok. Let us see what we can do with this, let us assume that there are set of points which are given as input for the tuple and then y is the set of features and x is the set of input ok. The next $X \varepsilon R^2$ and then is a vector or you can call it as this set of observed variables we know that x and y are related by an unknown function.

So, we only know that x is related to y; X_1 is related to Y_1; XY is related Y_1 and X_{10} is related to Y_{10} and so on. If you have two classes like this, we know that they are related, but we do not know how they are related right. So, and that is what we are going to be estimating. So, the classification exercise is to really estimate this model ok.

So, the goal is to estimate the unknown function g; it is also known as the classifier function such that $g(x)=F(x)$ for all inputs that you have provided. So, this is a very simple definition of the classification model. So, using this model, we want to roughly estimate g(x) ,and g(x) will give you this decision boundary ok.

(Refer Slide Time: 11:59)



WHAT DOES THE CLASSIFIER FUNCTION DO?

Assuming that we have a linearly separable x the linear classifier function $g(.)$ implements decision rule

- Fitting a straight line to a given data set requires two parameters (w_0 *and* w)
- The decision rule divides the data space into two sub-spaces - separating two classes using a boundary
- The distance of the boundary from the origin $= \frac{w_0}{||w||}$
- Distance of any point from the boundary $= d = \frac{g(x)}{||w||}$

Let us see how it is done. So, assume that we have a linearly separable x; so, it is very crucial this linearly separable x in this case ok. This set of points should be the linear classification function implements the decision rule that should beg right let us be consistent with the notation. So, if you have a linearly separable set of variables and each variable is a two-dimensional point such as this; it is going to give you a straight line that separates the point.

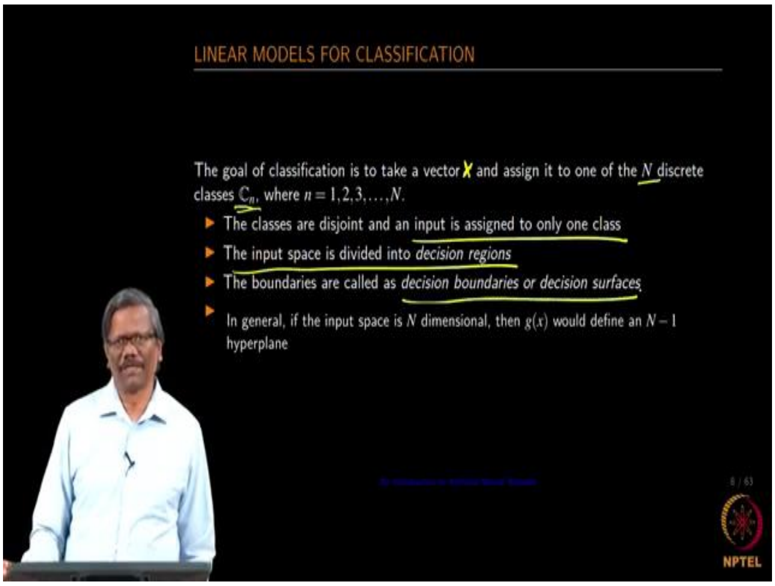
So, we required two parameters for this fitting a straight line to a given data set requires two parameters; one is called w naught another one is called w ok. So, in this case we call the was weight w naught as bias. The decision rule divides the data space into two subspaces right. So, initially when the points are plotted; it is one single space. So, once we have estimated that function, it divides that space into two different sub-spaces using a boundary.

So, the distance of the boundary from the origin is given by this relationship ok. So, note that w naught is the bias and then the one at the denominator is the norm of the weight vector ok. So, once you have the decision boundary, the distance from the origin is given by this relationship. So, one this is your boundary the blue line that you have here is your decision boundary and then there are points on either side of the boundary right.

So, you take one point and then drop a perpendicular to that that is going to be the distance of that point ok. So, this particular point that I am talking about from the surface it is given by this g which you are estimating divided by the or its the ratio of the classifier function to the norm of the weight ok; and then this is your normal this is normal w is normal to the decision boundary ok.

So, we can call this as a line separating these two classes or a 1-D hyperplane since the input parameters are 2-D the decision boundary will be 1-D. If the input parameters are in this space and if you are defining this by X_1 , Y_1 is Z $_1$ then you are going to be separating this by a straight line I am sorry by a 2-D surface.

(Refer Slide Time: 15:44)



LINEAR MODELS FOR CLASSIFICATION

The goal of classification is to take a vector X and assign it to one of the N discrete classes C_n, where $n = 1, 2, 3, \ldots, N$.

- The classes are disjoint and an input is assigned to only one class
- The input space is divided into *decision regions*
- The boundaries are called as *decision boundaries or decision surfaces*
- In general, if the input space is N dimensional, then $g(x)$ would define an $N-1$ hyperplane

6 / 63

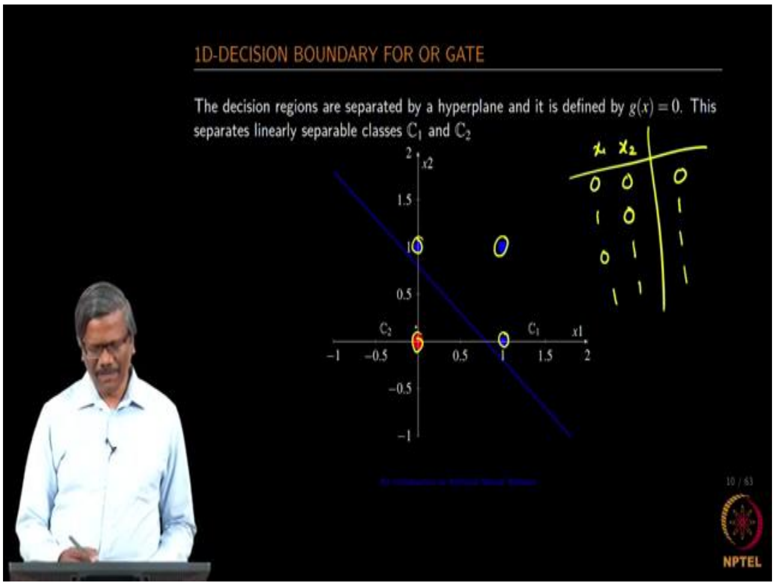
Again some more definitions as I mentioned earlier, the goal of the classification is to take a vector X; and assign it to N discrete classes C n where n =1 So, in the previous case we had two classes so, but you can also utilize this for n number of classes. The classes are disjointed and input is assigned to only one class.

So, this is something we should be aware of. The input space is divided into decision regions ok, the region that we had shown right the classifier had created that one d line is our decision region.

The boundaries are called decision boundaries or decision surfaces. So, one more thing also you may want to know the points on the line, where you will have the value of g(x)=0 So that means, also let us say that this particular point is on the boundary and then the g(x)=0 for this point.

And then for these points which are away from the decision surface or this 1-D line ok. So, the point such as these, their values would be greater than 0; that is how we subdivide the input region into two different regions and then in this case where the points are to the left of the decision boundary we have to go less than 0. So, we kind of set a threshold so that the line is estimated based on the threshold that we are setting up there ok.

(Refer Slide Time: 18:03)



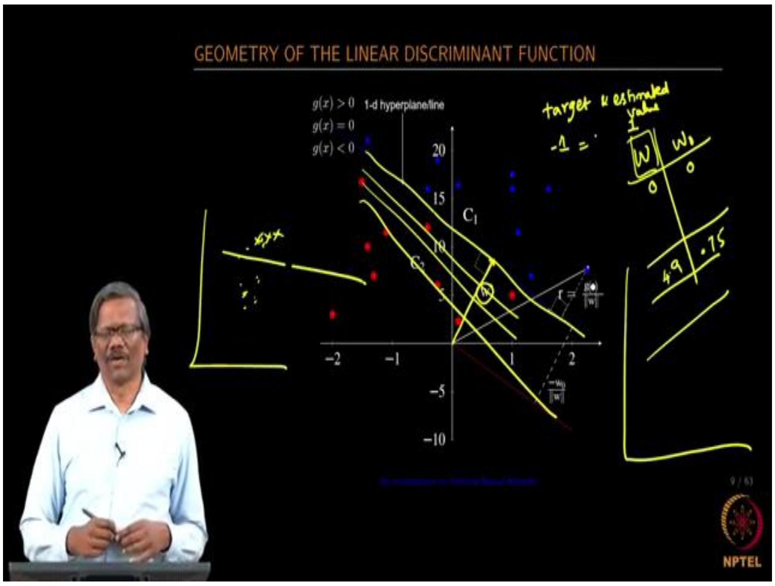
1D-DECISION BOUNDARY FOR OR GATE

The decision regions are separated by a hyperplane and it is defined by $g(x) = 0$. This separates linearly separable classes C_1 and C_2

x_1	x_2	
0	0	0
1	0	1
0	1	1
1	1	1

I think this is some plot in the enlarged space. So, we have seen some examples right especially in terms of how these points are divided.

(Refer Slide Time: 18:18)



GEOMETRY OF THE LINEAR DISCRIMINANT FUNCTION

So, one more thing I just want to explain before we go to the next one is. The line that we have here there the decision boundary that we are talking about it is estimated in an iterative fashion. So, initially when you start with the boundary line could be anywhere you know initially since we do not know exactly the model.

So, the model parameters that we have are w and w naught. So, we initially start with 0 and then given the input value and the kind of parameter estimation that we have the line would be somewhere here and then slowly and steadily based on more and more input that you are providing, it will start shifting parallel to this surface and finally, settle down in this line ok.

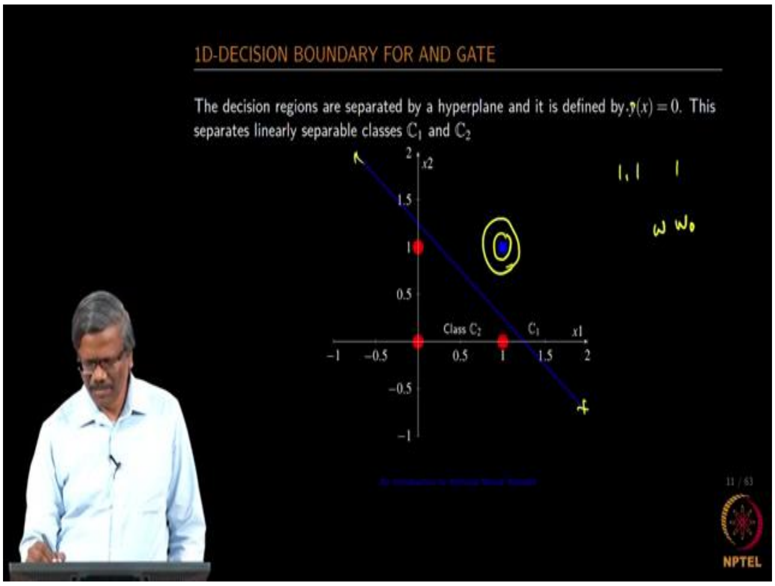
And then the distance from the origin is decided by as I mentioned the bias ok; and the orientation of that is given by the w. So, it could be you know points some points could be like this right. So, I want to have a line going like this from the origin to. So, it should be decided by the bias and also by this w with respect to the slope and with respect to the distance from the origin, and this is estimated in an iterative fashion.

So, initially it will be 0 and then finally, let us assume that you know these points are given by the equation ax; let us say 4.9 and then point 75. So, when you reach this point, you will have a line of these types I think the line would be this fashion if you have something of this type ok. So, how do we estimate this? So, this is estimated based on the error that we calculated you remember we spoke about the target and the estimated value right.

So, let us assume that for this point we know to which class it belongs to; right. Let us assume that this is equal to 1, and then if the estimated value is - 1 we know that we have to adjust win such a way that it increases the weights for all the positive values and if it is -1 and this is 1; we have to reduce the weights in the direction so that it comes closer to the target value ok.

So, again I am going to give some small examples here, would you know what this would you know what these points represent? So, we have a 0 here and there is a point 0 and 0; let us say this is x and y_1 have used x_1 and x1 we have x_1 as 1 and then 0. So, we are talking about the decision boundary for an OR gate right again if you look at this all the 1s are on the right side of the decision boundary and the 0 is on the left side of the decision boundary. So, we are able to linearly separate this right.

(Refer Slide Time: 22:33)



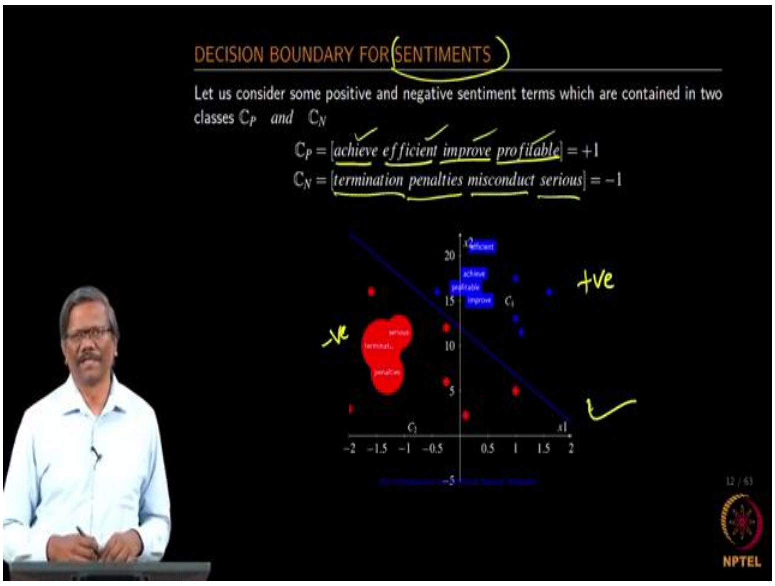
1D-DECISION BOUNDARY FOR AND GATE

The decision regions are separated by a hyperplane and it is defined by $y(x) = 0$. This separates linearly separable classes $\mathbb{C}_1$ and $\mathbb{C}_2$

So, this is and we have 1 only for 1 comma 1 rest we have 0 right. So, again we are able to linearly separate this point from the other points.

So, there is no problem for us again; if you want to estimate this you can start by initializing the values of the parameters w and w naught and slowly and steadily estimate this particular decision boundary or estimate this model g of x equal to 0; ok.

(Refer Slide Time: 23:17)



DECISION BOUNDARY FOR SENTIMENTS

Let us consider some positive and negative sentiment terms which are contained in two classes C_P *and* C_N

$$C_P = [achieve\ efficient\ improve\ profitable] = +1$$
$$C_N = [termination\ penalties\ misconduct\ serious] = -1$$

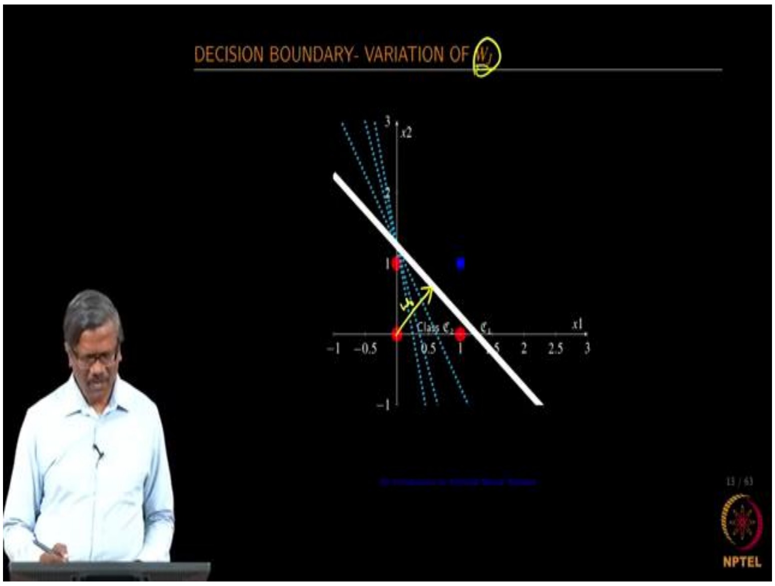
So, the decision boundary for this sentiment so, we need to map it to our space especially in the natural language processing right. So, as I mentioned earlier, what are the problems that we are talking about in terms of the classification of these sentiments. So, we have the sentiment analysis where we want to find out whether a given sentence as a positive influence or a negative influence in terms of the words that are present in the sentence right ah.

So, for example, if we have a set of words, they represent the positive influence and we have set of words that represent the negative influence so; that means, we would be able to again split the set of words, which are in one input space into two different sub-spaces as a positive region and negative region. So, all those negative words would fall to the left of the decision boundary and all those which are positive would fall on the right side of the decision boundary.

So, this is one example, so how do we do this? These are all words. So, how do I really provide the input, in earlier cases you know it was? So, simple we can provide the input as is in terms of the integers or real numbers and so on.

So, in this case we required some transformation of the input parameters. So, we will talk about that there is one example that I will be showing later ok. So, again to explain that in the case of natural language processing, we can utilize these classification principles to decide whether a given sentence is a positive sentence or a negative sentence.

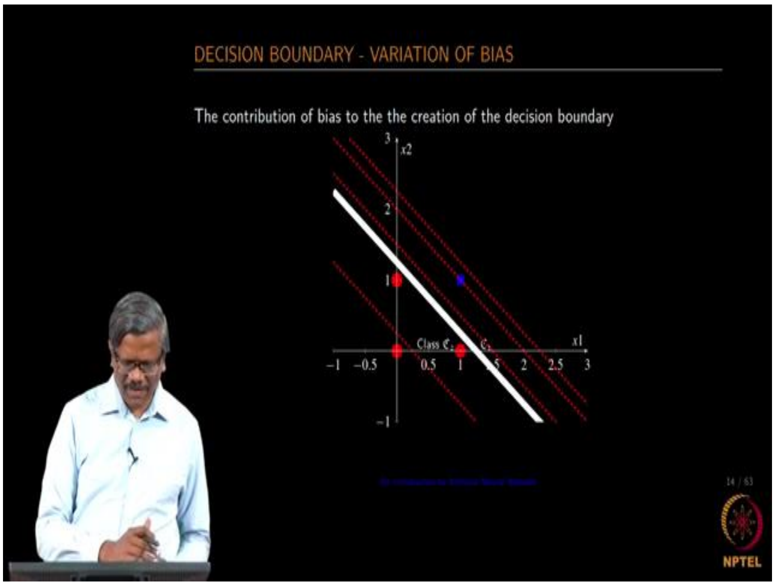
(Refer Slide Time: 25:17)



DECISION BOUNDARY- VARIATION OF W_j

So, it is possible for us to apply the same principle in natural language processing as well ok. So, as I mentioned earlier, the variations of the Wright in terms of this slope are decided by the W_j; and then the distance is decided by the w naught.

(Refer Slide Time: 25:38)

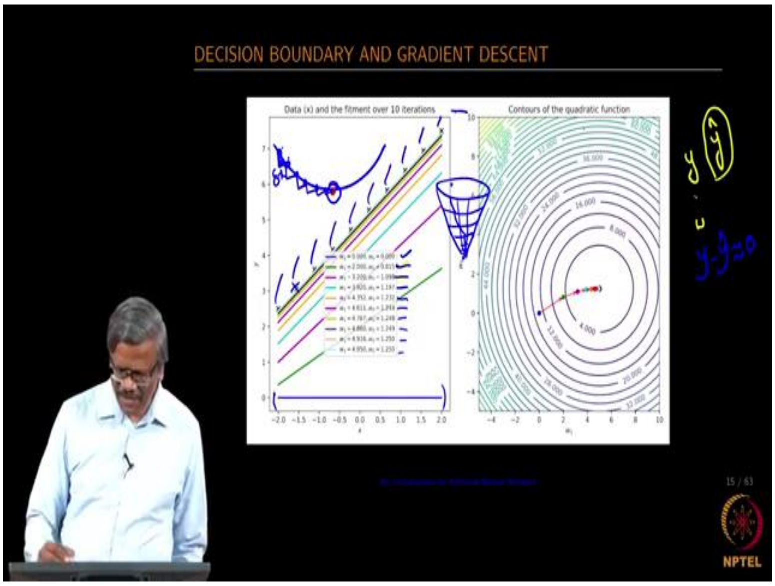


DECISION BOUNDARY - VARIATION OF BIAS

The contribution of bias to the the creation of the decision boundary

So, there can be you can actually you know write a program to show how the boundary changes you know during the iterative process. So, that is what I have shown here.

(Refer Slide Time: 25:52)



DECISION BOUNDARY AND GRADIENT DESCENT

Data (x) and the fitment over 10 iterations
Contours of the quadratic function
y
ŷ
y-ŷ=0
15 / 63
NPTEL

And then this is another example where I have taken a certain set of inputs about 10 points ok; and then I know to which class it belongs to, what I do not know is how to fit that the fitment is not known; the fitment is learned during the iterative process.

And then you can see that the lines starting from here this one which shows the plus here I will use a different color right. So, these points are my original points. So, this is where the target lies, since I know only the relationship between the input and the output; so, I start to estimate the classification function or the classifier using the iterative process.

So, initially, I start with 0 0 and then you see based on the input slowly and steadily it goes towards the original set of values and then on the right side what you see is the trajectory through which the iterative process mode; you remember we have a target and the estimated value.

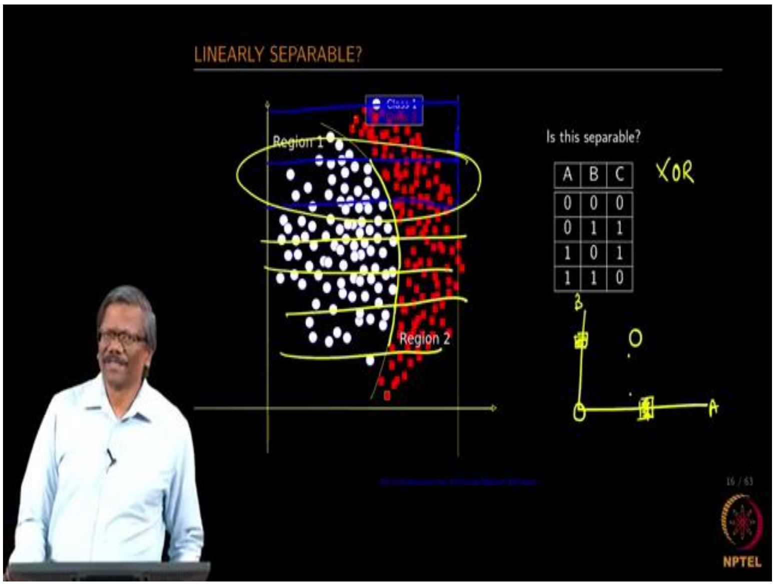
So, let us call the target as y and the estimated value as y hat right. So, when you start with y hat is not going to be equal to y that you are expecting. So, there is going to be some error and we propagate the error back to the model and ask the model to learn that parameter right.

So, that is where w_1 keeps changing you can see the variation here right for each of those see how it varies for every iteration. So, once it is closer to the point or once it reaches the minima, we stopped the iteration or there is no more change that I can bring in; that means, they minus y hat is very close to 0; that means, there is no more learning than I can provide I have learned the system. So, that is where you can see that the iteration process moves in this direction; and slowly and steadily reaches the final point ok. So, you can view this in terms of a dimensional diagram ok.

So, we started somewhere here because this is the error we have maybe I will draw a different like this. We started from here and this is our destination and then we moved here and so on. If you do not have a differentiable function it is not possible, and then at the end you know it is almost flat.

So, you do not have any delta x that you can move and that is when it stops you can think of this as a cone-like this right and then you start your iteration from somewhere here and then slowly you come down; and reach this point and that is what is shown from the top level is the top view of this contour, which gives you the quadratic function that changes every time when you change the values of w and w naught.

(Refer Slide Time: 30:33)



LINEARLY SEPARABLE?

Is this separable?

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Is it possible to separate this? Now we have been talking only about linearly separable a set of variables right. So, in this case you know well that it is only separated by a curve and not by a straight line. So, I am going to give an exercise to you; you need to go and then find out whether it is possible to do this; let us assume that this function is a piecewise linear curve.

So, can I first take only this region and say there are points, which lie to the left of it to the right of it and then next take another small piecewise linear region another one and so on.

So, you keep dividing this into various regions such as this and then finally, provide that the set of the point that we have here on the right side of the curve belong to class 2 and then the one on the left side belong to class 1. So, if is it possible to use whatever approach that we had shown.

So, this is an exercise for you and then the second one is this is an XOR; is it linearly separable you know that we have A here B here 1 1. So, we have points which are like this right. So, is it possible to linearly separate this? If I draw a line here this one is to the left of this. So, and this is not right. So, can I draw like this? No, so, how do I separate this function? So, is it possible to use the existing approach that we had followed to solve this problem; think about it?