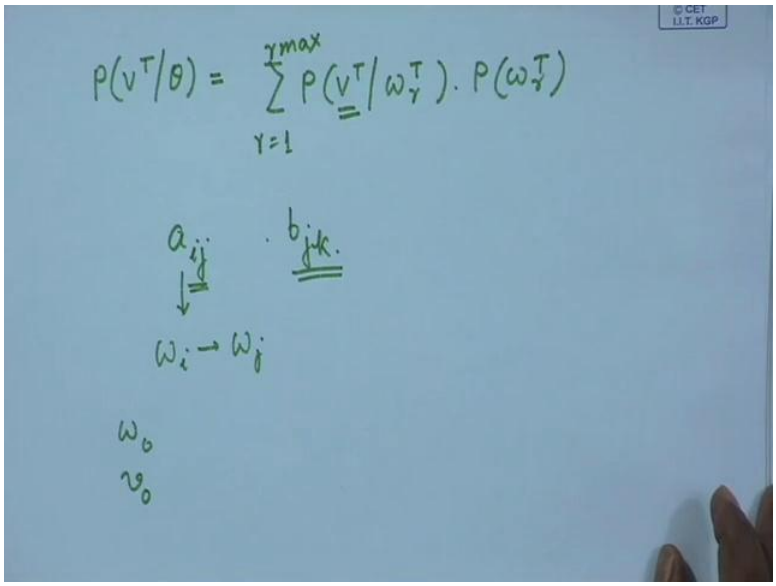
Pattern Recognition and Image Understanding
Prof. P.K. Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 40
Hidden Markov Model
(Contd.)

Hello. So, we are discussing about the temporal pattern recognition, that is the patterns which unfolds in time. We have started discussion on the tool, which is used for recognition or classification of such temporal patterns and this tool as we are discussing is what is known as hidden Markov model. We have seen earlier that this hidden Markov model has three central issues.

The first issue is evaluation problem where we have said that evaluation problem is nothing but if we are given a particular hidden Markov model say theta and you are given a sequence of visible symbols, then we have to find out what is the probability that the model theta has generated this sequence of visible symbols V T by any path. So, when I say path, it is the sequence of hidden states through which the model makes transition while generating the sequence of visible symbols say V T.

(Refer Slide Time: 01:50)



And we have seen that how to estimate this probability and this probability we have said that it is P V T given theta, and we have estimated this as sum of P V T given omega r T

into P omega r T, where this r will vary from 1 to r max, but r max is the total number of possible paths or total number of possible sequences of the hidden states through which the model can have transition while generating this sequence of visible symbols or sequence of visible states called V T.

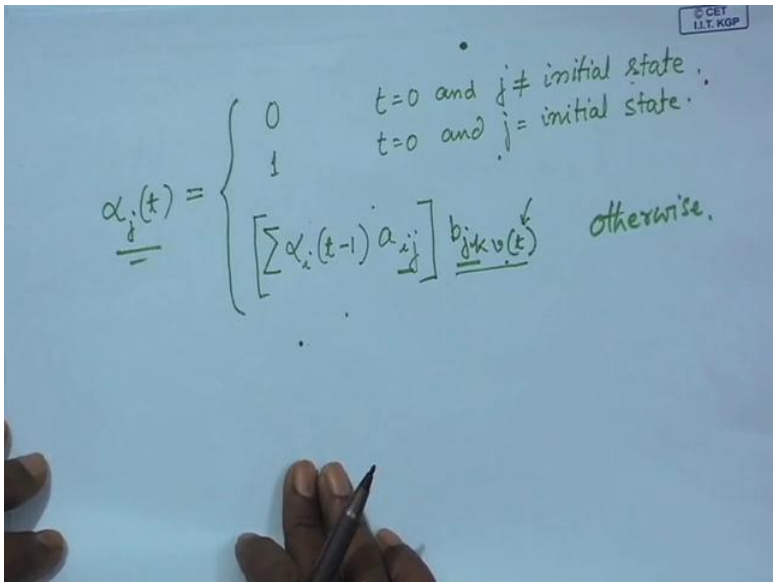
We have seen that there are two other major issues. The second major issue that we have already discussed is what is called the decoding problem. We have said that the decoding problem is, the aim of the decoding problem is to find out the most probable sequence of hidden states through which the machine will make transition to generate this sequence of visible symbols V T. The third issue that we are now discussing, we have just started that in the next class is the most important issue, which is learning of the hidden Markov model. So, by learning what we mean is suppose we have a course structure of the hidden Markov model that is we know that what are the hidden states and we also know what are the visible states or visible symbols.

So, given these two and given a set of sequence of visible symbols supposed to be generated by this hidden Markov model, we have to find out the transition probabilities a ij and b jk, where a ij we know that this is the transition probability from state omega i to state omega j, where both omega i and omega j they are hidden states and the transition probability or symbol emission probability b jk is the probability that the machine generates a symbol b jk when it is in state omega j.

So, these are the two transition probabilities that we have to estimate from a set of known sequences, which are to generated by this hidden Markov model and we have the course description, course structure of the Markov model in terms of that hidden states and in terms of the visible states. As we have already said that out of these hidden states, one of the hidden states say omega naught, this is called the final state that once the hidden Markov model enters this state omega naught; it cannot come out of this state omega naught. Any further transition it will make, it has to make within state omega naught only and while the machine is in state omega naught, it will generate only one visible symbol, which is represented by v naught.

So, learning problem is to estimate a ij and b jk, these two transition probabilities or the transition probability matrixes from a sequence of set of visible symbol sequences. In order to do that, we have used two types of probabilities.

(Refer Slide Time: 06:10)



One of the probabilities which we have estimated from using the forward algorithm that is we have defined like this that is alpha j t and we have said that this alpha j t actually tells the probability that the model will be in state omega j at time step t by generating t number of visible states, the first t number of visible states given in a sequence of visible states v t. The definition of alpha j t is it is like this that alpha j t will be equal to 0 if t equal to 0, and omega j is not an initial state and alpha j t will be will be equal to 1 if t equal to 0, and omega j is an is an initial state otherwise it will be defined like this. So, this will be otherwise.

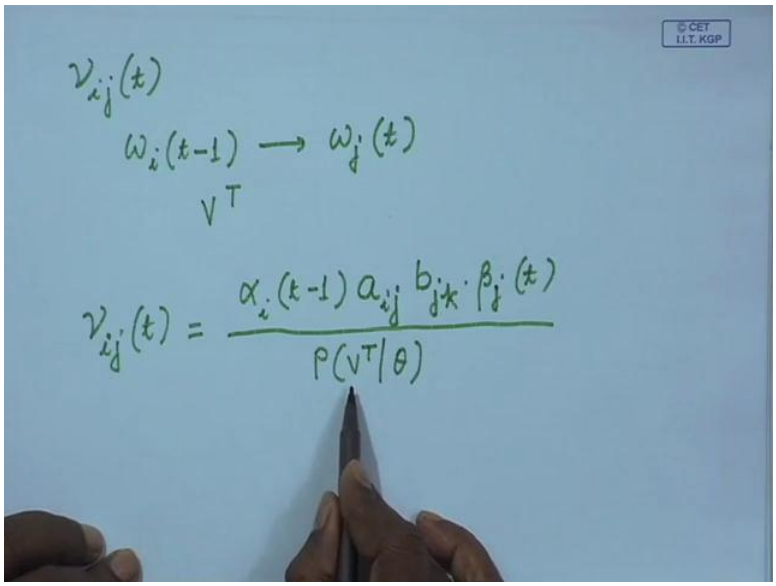
Similarly, using the backward algorithm, we have defined beta i t as given by this and this beta i t tells what is the probability that the model theta will be in state omega i and will generate the remaining symbols of sequence of visible symbols v t. That means, what is the probability that will be in state omega i and will generate the remainder of visible symbols starting from t plus 1 to to capital T. So, these remaining symbols of all visible sequence v t will be generated by this machine and under that situation, what is the probability that the machine will be in state omega i.

So, that is what is given by beta i t and we will see in a short while from now that both this forward probability and backward probability will be used for estimation of the probability, transition probabilities a ij and b jk. Now, whether this beta i t, which is obtained by backward algorithm or this alpha j t, which is obtained by forward algorithm

both of them uses a ij and b jk. So, beta i t uses a ij and b jk alpha j t also uses a ij and b jk, but the problem is this that this a ij and b jk that we are going to estimate that is what is the learning of hidden Markov model. So, initially I do not know what are the proper values of a ij and b jk.

So, the estimates of this alpha j t or beta i t, they are not the correct estimates because I do not know what are the exact values of a ij and b jk. So, these estimates are only some approximation, they are not exact as, a ij and b jk are not exactly known. So, what is the way out? We will do it this way that we define a probability of transition from state omega i to state omega j in the time step t minus 1 to time step t.

(Refer Slide Time: 10:13)



So, I define gamma i j t that is the probability of transition from state omega i at time step t minus 1 to state omega j at time step t, I define this. This is defined for a particular tuning sequence say V T. So, what I can do is I can define this omega i j t, this will be simply alpha i t minus 1 that is the probability that the machine will be in state i at times step t minus 1. It will make a transition to state omega j. So, I have this transition probability a ij it will emit a symbol V T. So, that will be b jk, which will be effective only for the k symbol emitted at t at time step t, which is equal to v k times beta j t.

So, this is the probability of transition from state omega i to state omega j in the time in the time step t i minus 1 to t while generating the sequence V T and this divided by P V T given theta. So, this will be my probability of transition from omega i to omega j in the

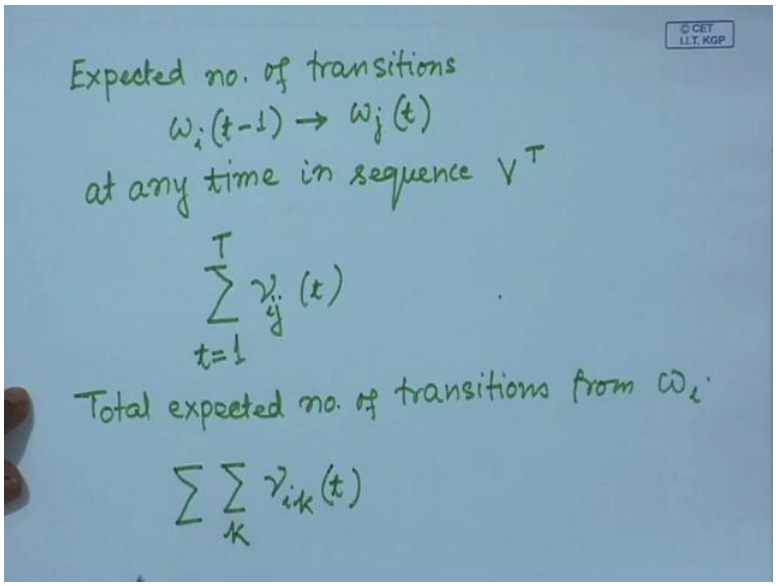
time instant t and you find that this P V T given theta, this is the probability that the model theta has generated this sequence V T following any path. That is quite obvious because when we have defined V T, definition of V T was something like V T; P V T given theta, the definition of P V T given theta was something like this.

So, it is summed over all possible paths omega t, omega r is one of the possible sequences of length capital T. I am summing over all such possible sequence of hidden states. So, this P V T given theta is actually the probability that the model theta has generated this V T following any path. This includes only that path where a transition from omega i to omega j from time step t i minus 1 to time step t is involved.

So, this is an estimate of the probability of transition in the t th step from omega i to omega j t at time step t. Now, here again you find that this omega i this a ij and b jk, they are still not exactly known. So, how this algorithm is going to work? So, the usual practice is initially you choose the transition probability is a ij and b jk are arbitrarily at random.

So, I choose some random values, some arbitrary values for the transition probabilities a ij and the transition probability b jk. Using that and using training sequence V T, you try to estimate what is omega i j t. Now, you can use this estimated i j t to refine or to improve the values of a ij or the values of b jk. So, how that improvement can be done? You find that the expected number of transitions from omega i t minus 1 to omega j t at any time in the sequence is given by.

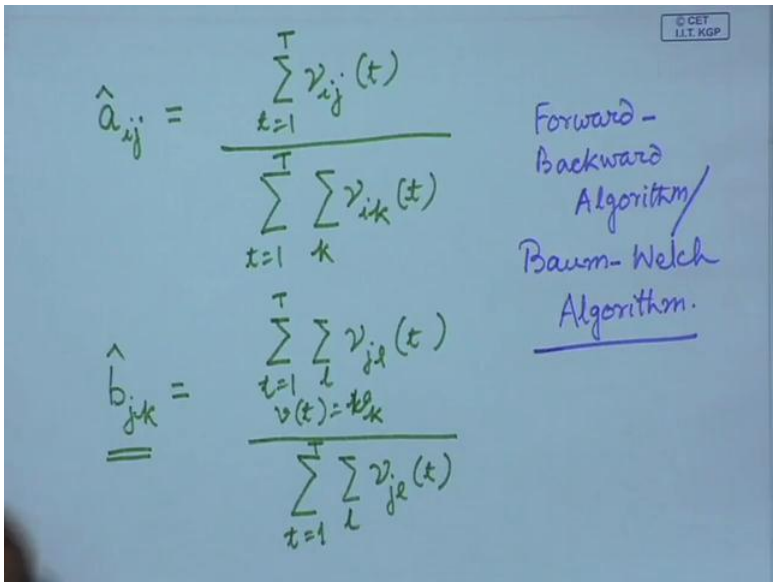
(Refer Slide Time: 14:56)



So, I will write that expected number of transitions omega i in time step t minus 1 to omega j at time step t at any time in the sequence, in sequence V T, it will be simply sum of omega i j t where this summation has to be taken over t equal to 1 to capital T. So, in this inter sequence v t wherever whenever I have a transition from state omega i to state omega j, I have to sum all of them.

So, that is what is giving me the expected number of transitions from omega i to omega j. The total expected number of transitions from state omega i, so against this, I have total expected number of transitions from state omega i that will be given by sum of gamma i k t or I have to take summation over k over the index k and this whole this thing has to be summed again over t equal to 1 to capital T. So, you find that I have two quantities. I have the total expected number of transitions from state omega i to state omega j given this sequence V T and I also have the total number of expected transitions from state omega i to any state. So, once I have these two quantities, then I can have the transition probability a ij.

(Refer Slide Time: 17:40)



So, I can have a refined transition probability a ij. I will put it as hat, which will be sum of gamma i j t, t is equal to 1 to capital T divided by sum of gamma i k t and the summation will be taken over all k, this whole thing summed over t is equal to 1 to capital T. So, this is our refined value of transition probability a ij and in the same manner, I can have a refined value of the transition probability b jk. So, I will have b jk hat, which will be given by sum of gamma j l t, take the summation over l and this whole thing is summed t is equal to 1 to capital T. I have to consider only those cases, but the emitted visible symbol is v k.

So, this has to be computed wherever V T is equal to v k because I am interested in the transition probability b jk and this divided by sum of gamma j l t summation over all l. I have to take summation over t is equal to 1 to capital T and this denominator is irrespective of any symbol that is emitted by the state from the state omega j. So, the numerator is only considering those transitions wherever the emitted symbol is symbol v k and denominator is irrespective of whether the emitted symbol is v k or not. So, this ratio gives me a refined estimate of b jk. So, the procedure I have is initially you choose the arbitrary values of a ij and b jk. Using those arbitrary values of a ij and b jk, you estimate what will be your alpha i t and what will be beta j t.

So, using this arbitrary estimate, the initial estimate of a ij and b jk, you have an estimate of alpha j t and an estimate of beta i t. Using this estimate of alpha j t and beta i t what

you go for, you go for an estimation of the transition probability from omega i to omega j and that transition probability is this.

So, you find that I have alpha i t minus 1 or j t minus 1, whatever subscript I use that does not matter. Similarly, I have an estimate of beta j t; I have an arbitrary estimate, initial estimate of a ij and b jk. So, I can estimate again using this a ij and b jk, I estimate what is P V T given theta and from here, I can estimate what is gamma i j t.

Using this gamma i j t, I go for refinement of the transition probabilities a ij and b jk. In the second iteration, I use this refined values of a ij and b jk again to estimate what is alpha j t minus 1, what is our beta i t minus beta i t. I go for estimate of what is gamma i j. Again, using that refined value of gamma i j, I go for further d refinement of a ij and b jk.

This entire step starting from refinement of gamma i j, refinement of b jk and refine, refinement of a ij, this entire process repeats a number of times unless I reach to a situation when the difference in a ij or the change in a ij and the change in b jk, they are within the tolerance limit. That is when I assume that my algorithm has conversed.

So, when I reach that state, that a ij and that b jk can be used for evaluation of my model theta for any sequence V T. That is how hidden Markov model is trained and because this process is using both forward estimation and the backward estimation, the algorithm is called forward backward algorithm. Also, this is also known as Baum-Welch algorithm.

So, let us try to recapitulate what we have done during this entire course on pattern recognition and applications. Initially, we said that whenever we want to classify or we want to recognize a pattern, for the pattern, we have to have some dissipaters or some features and when we take a large number of features put in a particular order that is what is called as feature factor.

So, when a pattern is represented by feature vector, suppose the feature vector is of dimension d, then this entire pattern is represented by a point or by a vector in a d dimensional space. The advantage that I have by representing a pattern by a feature vector is as the pattern is represented by a point in my d dimensional space, if I have two

patterns p 1 and p 2, the pattern p 1 will be represented by a point in d dimensional space and pattern p 2 will also be represented by a point in my d dimensional space.

Now, what I can do is I can find out the distance between these two points. If I find that the distance is very large, immediately I can infer that these two patterns p 1 and p 2, they are not similar, they are widely different. Whereas the distance between the corresponding points is in my d dimensional space is small, ideally if the distance is 0, that is both the points are marked the same points in my d dimensional space, then immediately I can infer that the patterns p 1 and p 2, they are same.

If the distance is small, I can infer they may not be same, they may not be identical, but they are very much similar because their corresponding feature vectors are similar. They are very close to each other. So, this is the advantage I get whenever I represent a pattern by feature vector and when I say feature vector having d number of components, each of these d number of components capture certain aspects of the pattern. These aspects or these properties of the patterns can be estimated in the time domain or in the spatial domain; in the time domain if it is the time domain signal, in the spatial domain if it is the special domain signal.

So, I say it is time domain say for example a speech signal; a speech signal varies with time right or the signal coming out of a microphone that is also a signal, which varies with time. So, whenever I have a time domain signal, I can extract the features in time domain itself. I say spatial domain signal say for example in case of an image; the image is defined as auto dimensional space. I can have combined spatial domain and time domain signal like in case of a video sequence. In a video sequence, as you know that the video sequence is nothing but a number of films which are played one after another at certain intervals. So, usually when we have commercial videos, they are played at an interval of say twenty five frames per second, at a rate of twenty five frames per second and or at the rate of thirty frames per second.

So, every individual frame is a spatial domain signal and when you take the number of frames, which are played one after another, we move into temporal domain. So, I can have signals, which will have both spatial property as well as temporal property. So, when I talk about features, I can extract the features in spatial domain, I can also extract

the features in the temporal domain. So, whichever way I extract the feature or I can also extract the features in the transformed domain.

So, this signal can be transformed to the transform domain using the various types of transformations. I can use Fourier transformations, I can use discrete cosine transformation, I can use velvet transformations I can use gavotte transformations, and many such transformations. So, by using these transformations, what we do is we simply transform the signal into the transformed domain and the features can be extracted in transformed domain also.

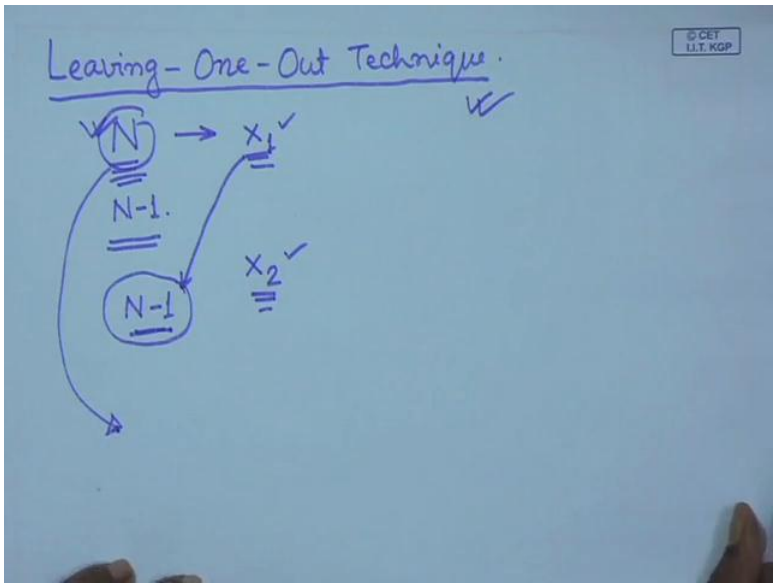
So, I can have spatial domain signal, I can have spatial domain features, I can have time domain features, I can also have transformed domain features. Each of these features as I say captures certain aspect or certain property of the pattern or certain property of the signal, when these features are put in a particular order, what I get is a feature vector and the advantage of converting a pattern or a single or a signal to a feature vector is as I said that it can be represented by a point in my feature space or in my d dimensional feature space. We have talked about various types of classifiers starting with our statistical classifier like bays' classifier; we have talked about the parametric classifier where the probability density function is having a parametric form.

We also have talked about the non parametric classifier when we still estimate the probability density function at a particular instance of the feature vectors, but we do not assume that the probability density function has any parametric form. We have also talked about other types of non parametric classifiers like nearest neighbor classifier, minimum distance classifier, and all these different sorts of classifiers. Then among the other types of classifiers, we have talked about using the discriminating functions for different classes. We have also talked about what are the decision boundaries with different classes.

So, using that, we can classify the patterns or we can recognize the patterns. We have also talked about the neural networks for pattern classification or pattern recognition. We have talked about hyper box classifier. We have talked about the fussy min max neural network for pattern classification. So, these are different types of pattern recognition techniques or pattern classification techniques that we have done throughout, we have discussed throughout this course.

Now, the question is as we have so many different types of classifiers or as we have so many types of recognizers, how to find out or how to estimate the performance of different types of classifiers?

(Refer Slide Time: 31:47)



Before that, there is another problem that how we design a robust classifier, so for the design of classifier, there is one of the techniques we will talk about is what is called leaving one out. So, what is this leaving one out technique? It is quite intuitive that when we train the classifier using the supervised technique that is when you design the classifier using a large number of feature vectors for which the class belongingness is known, then it is quite intuitive that if I increase the number of such samples used for the training of the classifier, the classifier will be robust.

Similarly, for testing, if I use a large number of samples for testing the classifier, the tested result will be more accurate. But, practically, it is difficult to obtain such large number of training samples or such large number of test samples because only after proper training and only after getting the test results, I can put the classifier into the actual job. So, the leaving one out technique is basically a virtual technique by which a limited number of samples, training samples or test samples can be posed as a large number of samples. So, suppose you had been given capital N number of training samples.

So, what this leaving one out technique does is you set aside one of the training sample for testing purpose. So, if I set aside one of the training samples, I am left with N minus 1 number of training samples. You design your classifier using N minus 1 number of training samples and the sample x, which you have left aside, you use this sample x for testing the classifier, which has been designed using N minus 1 number of training samples.

So, using suppose this is x 1, which I had left aside and using all the samples other than x 1, I train my classifier and use x 1 to test the classifier. Next, I set aside another sample say x 2 and use this x 1 for training. If I set aside this training sample x 2, I am again left with N minus 1 number of samples for training, but this N minus 1 number of samples include this x 1 and excludes x 2. So, using this N minus 1 number of samples, again you design the classifier and using x 2, you test the classifier.

So, if every time I set aside one of the samples out of total number of samples N, and every time I design the classifier using the remaining N minus 1 number of samples and the sample which was set aside, I use that for testing purposes. So, effectively what I am doing is I am designing N number of classifiers. When this x 1 or this x 2 is used for testing the classifier, which has been designed using the other N minus 1 number of samples, sometimes this test will result positive.

That means, the sample may be correctly classified, sometimes the test result may be negative that is the sample may not be correctly classified. So, what I can do is I can estimate a classification error rate that is out of so many test samples, how many times I have got misclassification of the sample? So, I have a classification error rate.

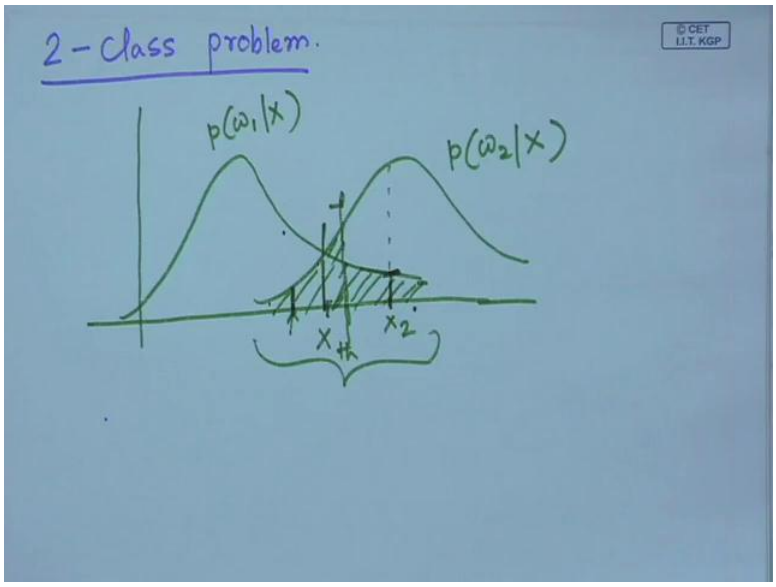
Then for designing your final classifier, you use all the N number of samples to design your final classifier and when all these N number of samples are used to design the final classifier, I can say that the error rate of the classifier will be at least less than the error rate that has obtained when I had designed N minus, N number of classifiers leaving aside one, so because I am using more number of samples to design my classifier. So, when I design the classifier, I know that what is the expected error rate? This is the method of designing the classifiers, which is called leaving one out technique.

Now, this can be slightly relaxed in the sense instead of every time leaving one sample, I can say set aside a 10 percent of the samples and I design the classifier using the

remaining 90 percent of the samples. This classifier will be tested using the 10 percent samples, which was set aside. So, every time I set aside 10 percent, using the remaining 90 percent, I design the classifier and test this classifier with this 10 percent samples.

So, here you find as I am setting aside 10 percent of the samples, I will be designing 10 different classifiers. For every classifier, I find out what is the error rate and then when I finally, design the classifier using all the N number of samples, all the samples, I know that the maximum error rate of this classifier will be the error rate of the individual sample that I have got. So, this is one way of designing technique. Now, suppose I have designed my classifier. Then how do I present the performance of this classifier or how do I estimate that what is the sensitivity of this classifier? Now, when I say sensitivity, typically I am talking about a 2 class problem.

(Refer Slide Time: 38:43)

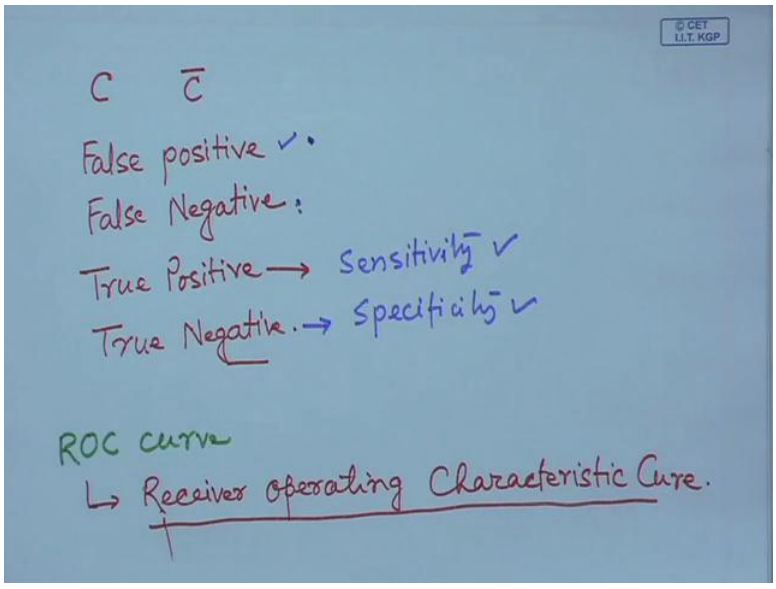


So, to discuss about this 2 class problem, I will take our old methods, which we have used earlier that I have set these two probabilities densities. So, this is p of omega 1 given x and this is p of omega 2 given x and for a minimum error rate classifier, we know that our threshold, our decision bound was somewhere over here. So, this is x threshold. So, in case of minimum error rate classifier, we had assumed that if the value of x which the vector x is to the left of x threshold, then the feature vector is classified to class omega 1. If it is to the right of x threshold, then the feature vector is classified to class omega 2.

So, this is my decision boundary and while doing this because there is a finite probability, that x over here may belong to x 2. So, I have an error, which is given by this amount. Similarly, when I decide feature vector say x 2 falls on this side because p of omega 2 given x 2 is more than p of omega 1 given x 2, I decide that this feature vector belongs to class omega 1, but there is a finite probability given by this that the feature vector may also belong to class omega 1. I have decided that it belongs to class omega 2, but there is a finite probability that the feature vector may belong to class omega 1.

So, the probability of error over here is this much, the probability error over here is this much and the total probability is nothing but the area under these two curves. Now, if I shift my decision boundary to say this side, then for one of the classes, the probability of error will increase; for the other class, the probability of error will decrease. So, if I shift my decision boundary towards class omega 2, then the probability of error for class omega 2 will increase; probability of error, the total error for class omega 1 will decrease. So, for all such decision processes, I have one type of probability error, one type of error rate which increases and the other type of probability which decreases with the shifting of my decision boundary.

(Refer Slide Time: 41:58)



So, a particular case of this supposes we consider a 2 class problem where I am interested in samples belonging to class c and the samples, which do not belong to class c. So, I represent them as c and c complement. So, this represents the samples belonging

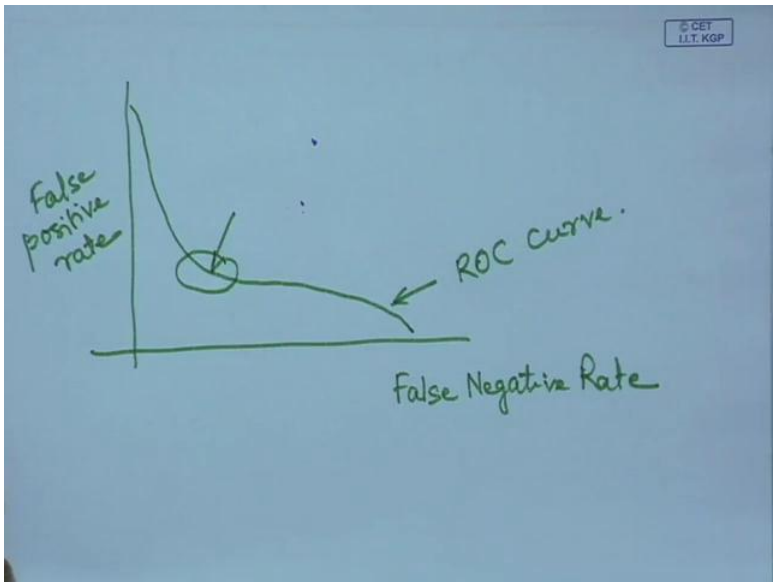
to class c and this represent the samples which do not belong to class c. Now, because of or decision making process or because of the classification process, I may have a situation that a sample, which actually belongs to class c complement is classified as belonging to class c. So, this is a kind of situation which is expressed as false positive. I can also have a situation that the sample actually belongs to class c, but it has been classified to be belonging to class c complement. So, from c, you have put it to c complement. That is what the classifier has done and such cases are known as false negative.

There are correct classification cases when a sample actually belongs to class c and it has been classified to belong to class c, which is called true positive at the case when a sample actually belongs to class c complement and it has been classified to belong to class c complement and that is the case, which is known as true negative. So, I have true positive case and I have true negative case. So, these are the different cases that I can have. This true positive or rate of true positive is also called the sensitivity of the classifier and the true negative this is what is called specificity of the classifier.

So, I have these two terms, one is sensitivity, other one is specificity. As we said that by varying the position of the decision boundary, I will increase one error rate. At the same time, the other type of error rate will decrease that is the number of false positives, the number of false negatives, in one case the false positive can go on increasing, the false positive negative go on decreasing. In the other side, the false negative may decrease, but false positive false negative may increase, but false positive will decrease.

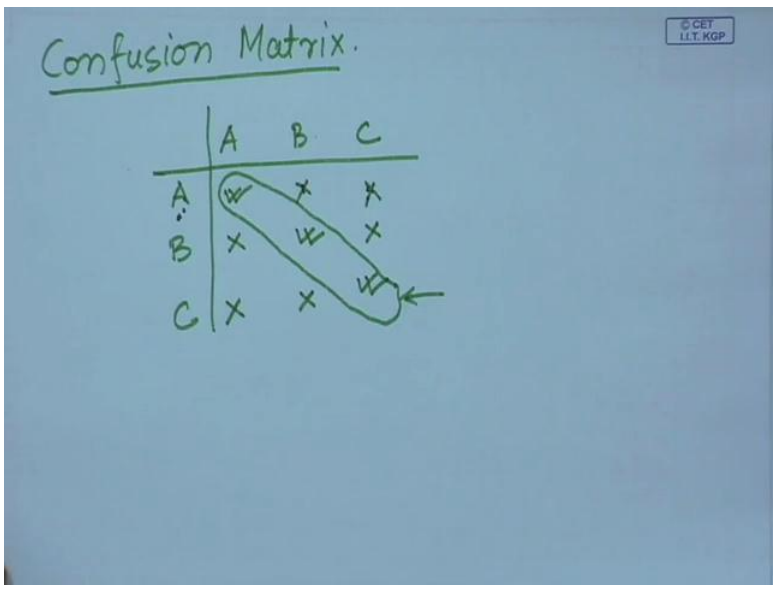
So, naturally the position of the boundary leads to a trade off and the situation of based trade off is actually described by a curve, which is called a characteristic curve or which is also known as ROC curve. This ROC curve, this ROC actually represents receiver operating characteristic curve. How do you plot this ROC curve? ROC curve is you plot one of the false cases against the other that is I can plot false negative versus false positive.

(Refer Slide Time: 46:23)



So, a curve something like this false negative rate versus false positive rate for different positions of my decision boundary. This curve may be say something like this and this is called an ROC curve or characteristic curve. The advantage of this ROC curve is it simply says that how sensitive your classifier is against shifting of the decision boundary. And from this ROC curve, I can choose the best trade off, may be something somewhere over here I can say that total number of false positive and false negative that will be minimum. So, whatever for whichever position of the boundary I am at this position, that position in can use as my decision boundary position.

(Refer Slide Time: 47:56)



The other way of representing the performance of a classifier is by means of what you call a confusion matrix. The confusion matrix says that suppose I have got three different classes, class A, class B, class C and the confusion matrix is plotted in this manner A B C. So, it simply says that how many samples belonging to class A has been classified as class B, how many samples belonging to class A has been classified as class C. So, these are actually wrong classifications or false classification. The sample actually belongs to class A, but it has been classified as class B or class C. So, these are false classes and it also says that how many samples belonging to C has actually been classified as class A.

So, these are the false classifications. This is the true classification. Similarly, how many samples belonging to class B has been falsely classified to class A, how many samples have been correctly classified as class B, how many samples have been falsely classified as class C. Similarly, for class C, how many samples belonging to class C has been classified to A, falsely classified to B, how many samples have been correctly classified as class C. So, you find that the diagonal of this confusion matrix, this actually gives you that how good your classifier is because this is what represents the number of correct classifications. So, these are the defined techniques. There are many other techniques, which can be used for to measure the performance of the classifier.

So, in this course, we have talked about different classifier techniques, classification techniques or recognition techniques. We have also talked about different types of clustering, where the similar patterns of the similar feature vectors are put into the same group. We have talked about the classification or recognition of temporal patterns and towards the end, we have concluded our course on pattern recognition, and applications using the design techniques like leaving one out and the different types of performance measures of the classifiers. So, with this, we come to end of this course. I hope you have all enjoyed this course and you got benefitted by this course.

Thank you.