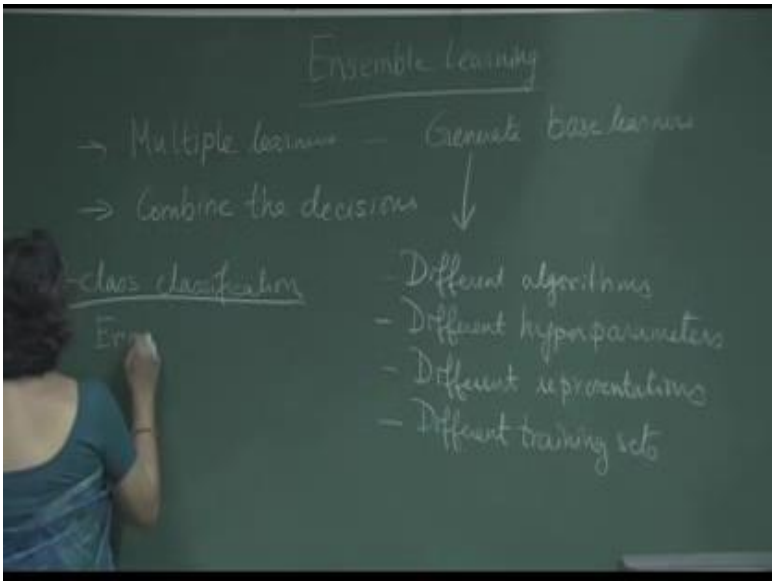
Introduction to Machine Learning
Prof. Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module - 8
Lecture - 35
Introduction to Ensembles

Good morning, today we will start the 8th module, where we will talk about ensemble learning. In ensemble learning, we look at multiple classifiers and combining the output of the multiple classifiers in order to get better prediction or classification accuracy. And under certain conditions, where the classifier outputs are independent of each other and make errors in an independent manner, it is possible that by combining the outputs of several classifiers. We get a resulting classifier, which is better than any of the constituent classifiers.

So, today we will give a brief introduction to what is ensemble learning, and what are the some of the different ways in which the different learners can be formed.

(Refer Slide Time: 01:21)



So, the topic of today is ensemble classifier. More specifically, let us talk about ensemble classification. So, we have multiple learners, or multiple classifiers. And given a test example, this multiple learners will give different outputs; the outputs may be all same, all different, or you know some of them will be same; some of them will be different.

So, this multiple learners will give different decisions, and we want to have a way of combining the decisions output by the multiple learners. So, for this, we have to generate a group of base learners. And we will see how we get to generate a group of learners; it can be done in multiple ways, and we will discuss about them. And these different learners have to be different; and there are many ways in which the learners can be different, they may be using different algorithms. So, you could think of there is a learner which uses a decision tree, or learner which uses a neural network, a learner which uses a support vector machine and so on. So, we can fed the training example to different learning algorithms and combine the output.

But more usually, we may use the same algorithm, but use it such that the learners are different. How, so the different learners may use different parameters or different hyper parameters. So, if you take the same learning algorithm, but give different values to the parameters, you get different models as output - the models will be different. They may have different representations; the learners may have different representations or different modalities or may use different training sets.

Earlier, we had talked about variance with respect to a learning algorithm. We said that learning algorithms may have high variance. So, when they get, they are applied on different subsets of the data, if the algorithm has high variance, the model output will be different. So, we can try to give you know learning algorithms which are weak and have high variance can be fed different training set, and as a result each time we will get different models which can be combined.

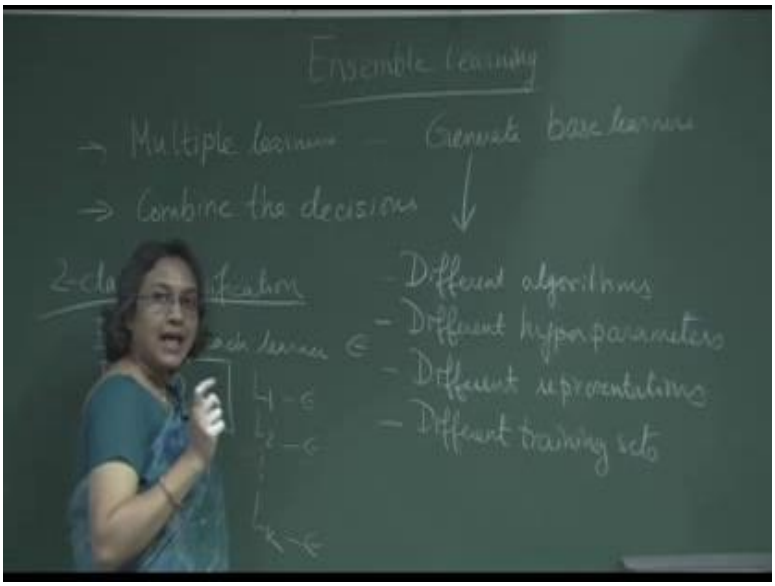
How do we set different hypo parameters? For example, in neural network, we have seen that there are certain parameters that we can set; we can decide the topology of the network that is the number of hidden layers, number of nodes in each hidden layer, the values of the weights and bias and so on. So, these are the different parameters. By giving different values of these parameters, we can get different; you know within neural network we can get different models. Then we have looked at decision trees; in decision tree, we try to come up with the strategy for deciding this attribute on which to split at a particular node.

For split, we can use different heuristic functions such as entropy, gini index etcetera, but we could modify the decision tree algorithm, so that we use different functions, so that

the different learners do not split on the same attributes and force the learners to form different decision trees. These are some of the ways, in which you can get a group of learners.

Now these ensembles you know often it is found that no learner is absolutely better than the other learner, there is no one good learner for many problems. But if we have multiple learners, and they may not be good, in fact they may be weak, but they may make independent errors. So, all the learners which are different, they may work well in different parts of the data or the instant space; and by combining these weak learners, it is possible to get a very strong learner.

(Refer Slide Time: 06:46)

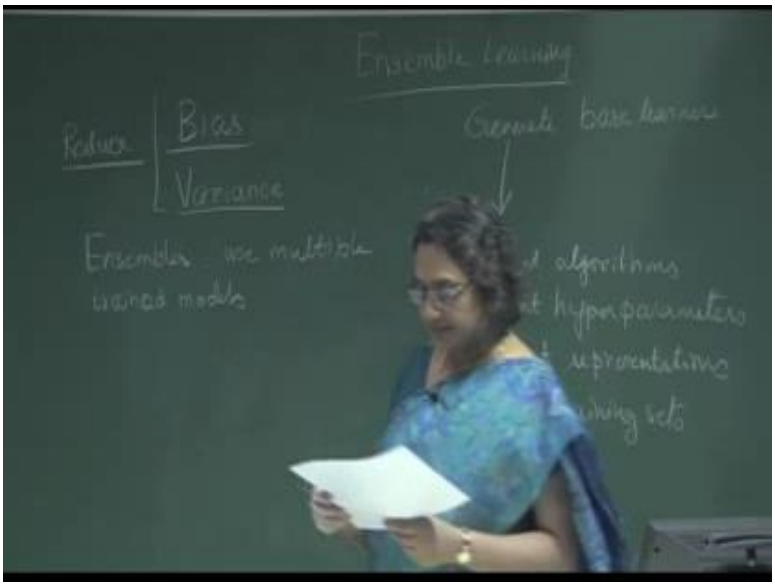


For example, let us take the 2-class classification problem. In 2-class classification, we have two classes - positive and negative. Suppose, we take some learners and each learner has an error. So, the error of each learner is epsilon, but epsilon has to be greater than 0.5, why, epsilon equal to 0.5 is a random model which does not give any information that is an example that you give it has 50 percent chance of being positive 50 percent chance of being negative. So, error equal to 0.5 does not give you any informative learner. So, for any learner, error has to be greater than 0.5.

Now, we want to see that in a ensembles we want the different constituent learners, each will have an accuracy that accuracy may not be very high, but it must be greater than 0.5. But, we will show that it is not required, if other conditions are met, it is not required that

the accuracy has to be much higher than 0.5; it has to be higher than 0.5, but it need not be very high. But, if these different base learners satisfy certain independence conditions, then it is possible to combine weak learners, each having error greater than 0.5 in order to get a very strong learner. So, this error has to be such that you know all we have different base learning algorithms L 1, L 2, L k and each has error epsilon, but the errors made must be independent.

(Refer Slide Time: 09:06)



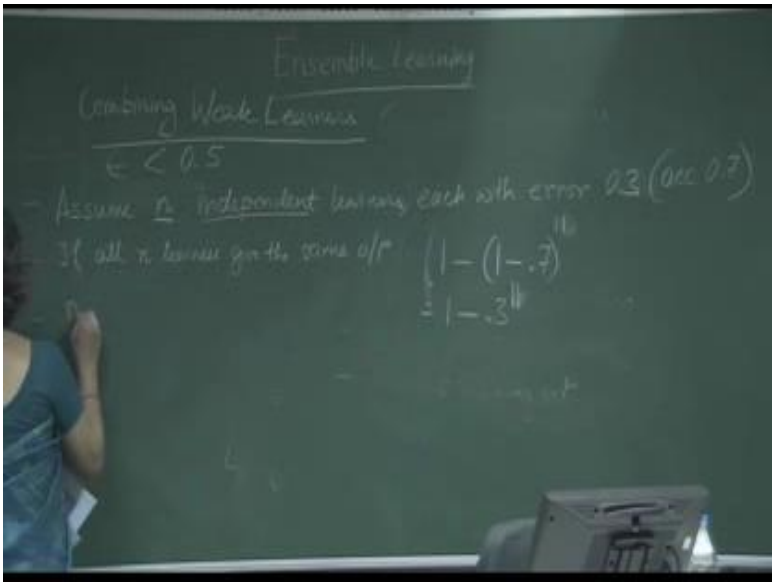
Now, we will just take a make a digression, and talk about bias and variance. We have already discussed bias and variance earlier. What is bias, bias is a measure of how flexible the model is. So, if the model is very flexible or very powerful then the bias is low, for example, a decision tree is quite flexible, bias is low; a neural network is quite flexible, bias is low; but a linear classifier is not so flexible and bias is high.

Secondly, we talked about variance. Variance is high when if you give different subsets of data as training set, the models output are very different, then we say variance is high. Usually, that hypothesis for which bias is low they have high variance that is more flexible, the more powerful the representation is then variance will be high, bias will be low. Now our objective is to reduce both bias and variance. So, we want low bias and low variance. And we will see that one of the ways we can achieve both low bias and low variance is by ensembling.

So, we want low bias error and low variance error. So, we have by using ensembles, which use multiple trained models. And let us assume that these models have low bias and high variance. By combining the output, we can get low variance, while maintaining low bias. In fact, even if the bias is high to start with, by combining we get a classifier, which has low bias, because the individual hypothesis may have high bias.

But when we combine them we get a new hypothesis, which may be out of the hypothesis class, but we get a new potential hypothesis which has low bias. So, by using ensemble, it is possible to get low bias and low variance. And if we achieve low bias, low variance, what does it give us we will get less over fitting; we will not have to worry about stopping criteria when to stop etcetera, if we can achieve low bias and low variance. So, the essence of ensembles by which it achieves this is by combining different base learners.

(Refer Slide Time: 12:31)



So these learners may be weak; so combining weak learners. A learner is weak if its error is high, but error cannot be higher than 0.5; error must be less than 0.5. But, if error is not very low, we can say it is a weak learner. So, in order to combine weak learners we need certain conditions. So, the main condition that we require is independence. So, let us say you assume n independent learners. So, we assume n independent learners; and let us say each has accuracy of 70 percent or each with error 0.3 or that is accuracy 0.7. So, we have n independent learners each with 0.3 error or 0.7 accuracy. Now you apply the

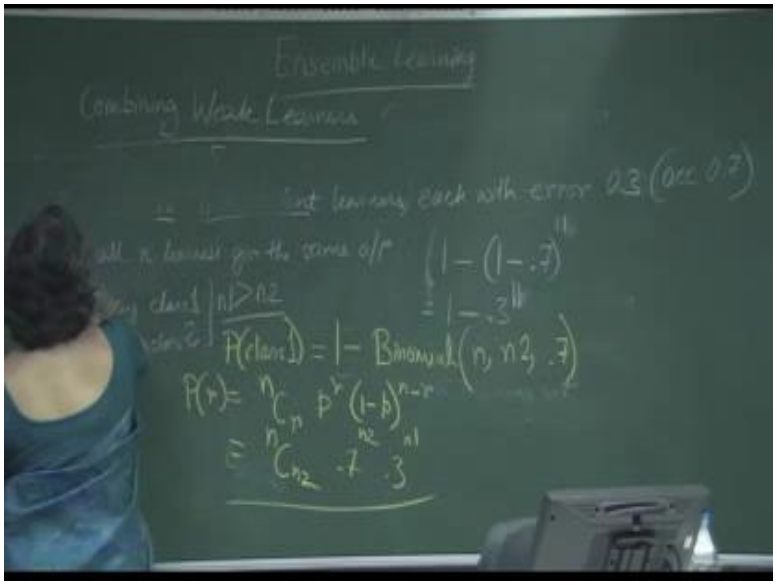
learners on some test data. So, it can be that all the learners give the same output, if all the learners get the same output that is a possibility.

Now, if all n learners give the same output on a test example, then what is your confidence on the test example? Because the learner has accuracy 0.7, your confidence if you use one learner your confidence is 0.7. But if you have ten learners, all of them saying positive then what is your confidence on the resulting ensemble. So, you could say that your confidence is 1 minus 1 minus 0.7 to the power 10. So, if all ten learners, you use ten learners, all of them give the same output, your confidence on the learners is 1 minus 1 minus 0.7 to the power 10, which is 1 minus 0.3 to the power 10, so you can find out the value of this expression, but it will be quite high. It will be very close to 1, when you do this.

So, you are getting higher confidence, if all the learners agree on the output. In practice, we cannot always expect that all the learners would agree. If the learners have accuracy 0.7, it is unlikely that all of them will agree on the output, so some of them will say class 1, some of them will say class 2. And we will our strategy will be taking the majority class. So, it is unlikely that all the learners will give the learners are independent; it is unlikely that all of them will agree, but some of them will agree and we will get a majority. In fact, if you want to get a majority for a 2-class problem instead of using 10, we can use a odd number, we can use 11, so that we have always we have a majority on a 2-class problem.

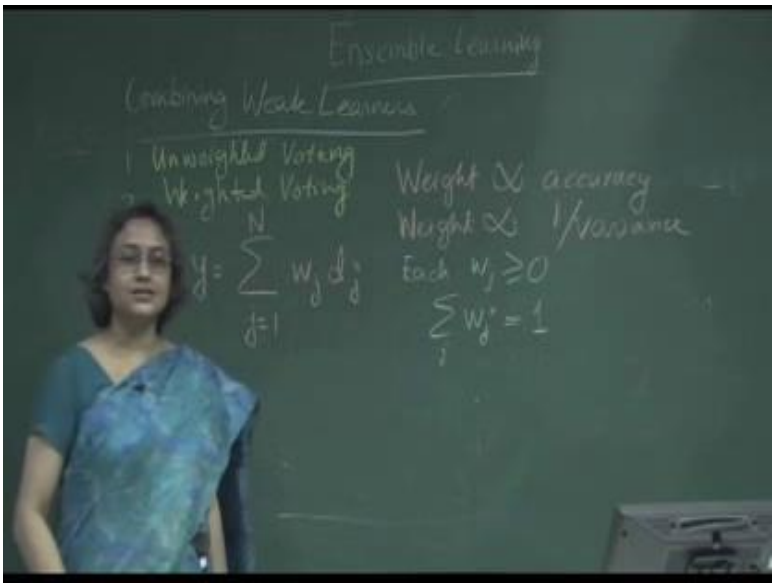
Now, in the general case, if out of n learners, n 1 say class 1, and n 2 say class 2. And without loss of generality, let us assume that n 1 is greater than n 2, so that is n 1 is majority, and therefore class 1 is the opinion about of the majority of the learners. Then the probability of the output really being class 1, in such case, the probability of class 1 is given by this expression, so 1 minus binomial distribution of n the minimum of n 1 and n 2. In this case, we have assumed n 1 greater than n 2, so minimum of n, n 2 and the accuracy which is 0.7.

(Refer Slide Time: 17:20)



And, the binomial distribution is given by probability of r equal to n factorial, so it is actually n C r p to the power r 1 minus p to the power n minus r. So, if you substitute here this here, we get n C n 2 p is 0.7, 0.7 to the power n 2 and 0.3 to the power n 1. So, this is the probability of this class. So, this is given by the binomial distribution and we see that combining these weak classifiers gives us a classifier with higher accuracy.

(Refer Slide Time: 19:32)



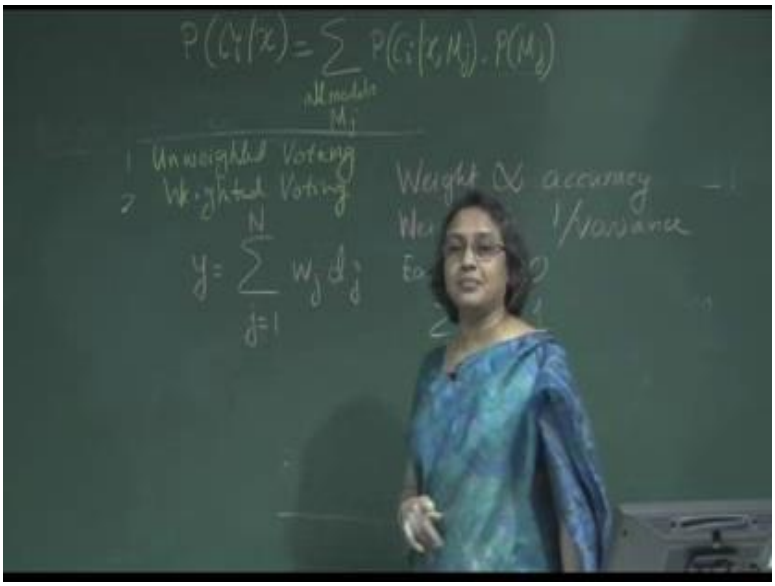
Now, next we will look at how learners are combined the mechanism of combination. So, combination in the simplest case the combination can be done by voting. For example, just now we looked at majority combination or majority voting. So, the voting can be equal weight voting we can give each learner equal weight. So, we have a equal

weight or unweighted voting or we can have weighted voting that is we give different weights to the different classifiers. We can give different weights to the different classifiers.

And how can we give the weights to the classifiers? So the weight can be proportional to the accuracy of the classifiers right or weight can be proportional to one by variance; lower the variance of a classifier we can give it high weight so there are different ways in which we can weigh the classifiers. Then we can give them weight based on their probability of being correct which is related to accuracy and this is what is done in Bayesian classifiers.

We have talked about Bayesian classifiers briefly. So, we can say that the output y is taken to be if there are N learners, we say w y is equal to sigma j equal to 1 to N W j d j where d j is the output of the jth classifier and W j is the weight of the jth classifier. So, they must satisfy certain condition, so each w j must be greater than equal to zero and sigma W j over j equal to one to n must be equal to one so these are the conditions that the different values of W j can satisfy. Now, we have talked about Bayesian learning earlier in the class.

(Refer Slide Time: 22:03)



And we have seen that in Bayesian classifier we take probability let me write it at the top. In Bayesian learning, we take probability of a C i given x, C i is class i given a particular training example x to be equal to sigma over all possible models. In Bayesian

learning, we use all hypothesis, we combine all hypothesis or all models; so summation over all models probability of getting class C i given x and m j times the prior probability of m j, so this is what is given by Bayesian learning.

In Bayesian learning, what we are doing is that we are finding the probability of a class in terms of the prior probability of the model, and the probability of class i given the learning test set and the model. So, we are combining the outputs of the different models, we are combining it and the weights are the prior probability of that models, so this is one way of ensembling, but this is not a vary.

We have talked about earlier, and we will talk again that this is often not practical because the size of the hypothesis space is so huge, we cannot take get so many hypothesis, and combine them, and there are other difficulties. For example, many learning algorithms, many of them that we have seen, they output a class and they do not output a probability. For example, in decision tree, we can modify the algorithm to output a probability which is the fraction of examples at a particular leaf node. But many other algorithms in the conventional form, they output a class and not a probability, so we cannot directly applying them, we have to modify the algorithm or we have to reinterpret the output of the algorithm in terms of the probability.

Secondly, it is not always possible to get an estimate of the prior probability of a model, but the most important reason, why this Bayesian combination is not very practical is because given a hypothesis space typically the number of hypothesis is so large; it is not possible to enumerate all of them and combine them. So but this is what the base thing do.

(Refer Slide Time: 25:25)

So, from this, what we get is, so in the base classifier, we can write base classifier as the output y equal to we can rewrite it as that class c j included in the set of classes C for which summation over all hypothesis in the hypothesis space capital H, so this expression is maximum.

And what is this expression? It is probability of c j given h i times probability T given h i, T is the training data or we can which is the sample that we have. So, probability c j given h i times probability of training data given h i times the prior probability of h i. So, when you apply the base model to classification, this is the expression that you get. Where y is the predicted class, C is the set of all possible class, and c j is a particular class - c 1 c 2 etcetera, capital H is the hypothesis space, h i is one element of the hypothesis space, T is the training data.

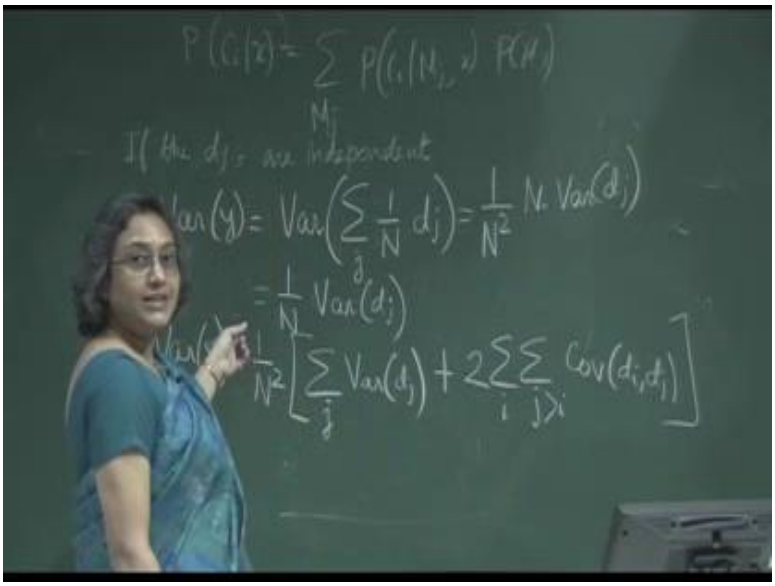
So, the base optimal classifier will represent a hypothesis and this hypothesis is not necessarily in the hypothesis space H, when you combine this hypothesis you get a new hypothesis. If you combine the output of the different single layer neural network, you get a hypothesis, which may not be a single layer neural network, so the hypothesis that you can combine a set of conjunctive Boolean formula to get a hypothesis, which is not necessarily a conjunctive Boolean formula.

So, the ensemble gives you a hypothesis which may be outside the hypothesis space H, but such hypothesis when you use base optimal classifier the output hypothesis that you get is the optimum hypothesis in the ensemble space. But as I said the base classifier

cannot be practically implemented because of various factors, hypothesis space may be too large; it may be difficult to get probabilities or prior probabilities of the hypothesis space.

There is a model, which is sometimes used, which is called Bayesian model average or BMA. In Bayesian model averaging, it is all possible models in the model space are awaited in the probability of being correct as we do in this thing; and it is a optimal hypothesis, but we cannot use it practically, so we can use certain variations by using some sampling Monte Carlo sampling etcetera, but we will not talk about it. Instead, let us talk briefly about why are ensembles successful.

(Refer Slide Time: 29:16)



So, from the Bayesian perspective, we have seen that I just removed this. We have seen that probability C i given x is equal to sigma probability of C i given overall models M j C i given M j and x times probability of M j, and we know that this gives us the optimal hypothesis from the Bayesian perspective.

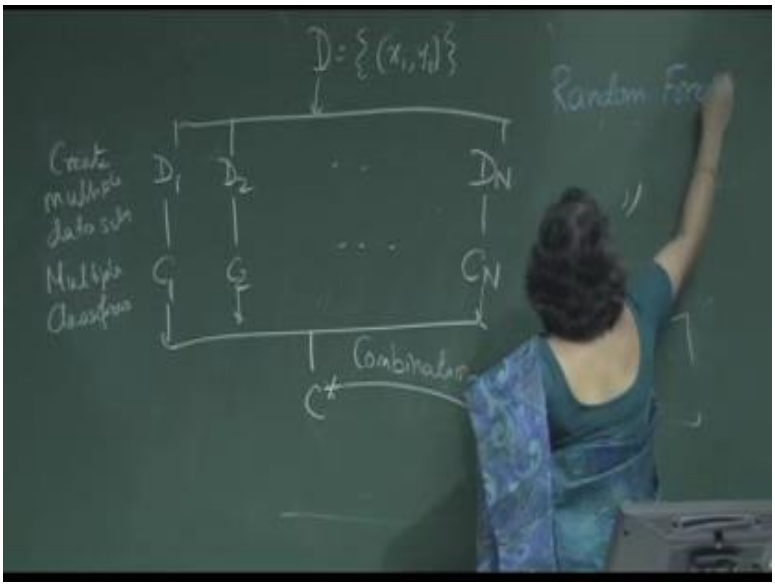
From other perspectives, if the outputs by the different learners are independent, so their output may not be accurate, but if the outputs are independent then the variance gets reduced. And it can be shown that the variance of the output, if you are using N number of learners, the variance of the resulting ensemble is variance of summation over j 1 by N - N is the number of ensembles 1 by N d j. This is the resulting variance which can be shown to be 1 by N square times N into variance of each of the individual classifiers,

which is equal to 1 by N variance of d j. So, if the outputs by the N learners are independent, the variance gets reduced by a factor of N, which is a very strong result.

Now, even if the learners are not completely independent, but they are not completely correlated, we can have a modification of this expression. So, in that case, we can tell if they are dependent, but error increases with positive correlation, we can tell that the variance of y is still 1 by N square variance of this thing, which can be shown to be equal to 1 by N square summation over all models j variance d j. So, this is the same as the previous factor plus there is another term which is 2, we take all the pairs of the learners, so 2 summation between the pairs of learners, co-variance of d i and d j. So, if the learners are not completely correlated, we will see that even in this case, there is a reduction in variance.

Now, what is the main challenge of developing ensemble algorithms? The main challenge in developing ensemble algorithms is not to get individually good learners, which we have been doing in the rest of the class, but to make sure that these learners are independent. And how can this be achieved? Let me rub this board, so there are different ways in which we can achieve independence and what I told in the beginning of this class, you have a training data.

(Refer Slide Time: 33:00)



You have a training data D, which is given to you; D is x i, y i so this is the training data that is given to you. Now one of the things that you could do, you put to the n learners

you will feed the training data, so D 1, D 2 and D n. Now it is possible that two different learners you feed different training data which has sample from D, so the data fetched to the different learners may be same or they may be different. And in the next class, we will look at different ways of sampling to make this training set different. This is step one to create multiple data sets.

Next, what you can do is you can feed this to the different learners, so multiple learners or multiple classifiers. So, you can feed this to C 1, C 2, and C N. Now these classifiers can be the same classifier or can be different classifiers or same classifier with different parameterizations. So, you can make either the D is different or the C is different or both different.

And as a result, you will get output from all the learners, and you will combine the output by using a voting mechanism, and get the resulting ensemble classified. And the combination can be done by we have just talked about weighted voting, unweighted voting, it is also possible to use other types of combination like taking the median, taking the product, taking the minimum, taking the maximum, so some type of voting or combination is done.

So, in the next class, we will look at bagging and boosting by which we can force these data sets to be different. We have already talked about different learners and we have also talked about parameterizations of the learners. If you wish, you can go through one particular ensemble algorithm which is called random forests. Random forests are an ensemble defined on decision trees, where you force your ensembling model to come up with different decision trees. How do you force them, you put some randomization in the attribute selection phase, where different attributes will be selected in a random manner when you go to split a node. And these different learners act on different subsets of the data.

So, random forests is basically an ensemble of decision trees which do different samples by bagging which is what we will talk about in the next class. And, use the decision tree algorithm with different with a randomized approach to force them to produce different decision tree every time the algorithm is run. And based on this, we have an ensemble classifier which is called random forest. And in many examples, many test cases, random

forest has been found to do much better than the basic decision tree; and it is a powerful learning algorithm.

With this, we close today. We will take up bagging and boosting in our next class.

Thank you.