Introduction to Medical Imaging and Analysis Softwares

Professor Debdoot Sheet
Department of Electrical Engineering
Indian Institute of Technology Kharagpur

Module 3
Lecture No 11
Decision Trees for Segmentation and Classification

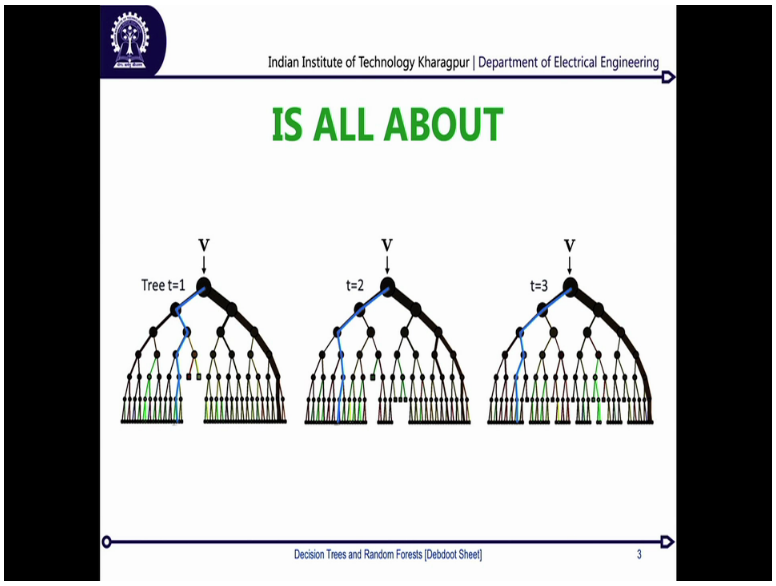
Welcome, so in this lecture we would be covering about one of the super learning techniques as I call them and that is on decision trees we would be calling back with next generation of learner called as random forest and this is a coupled lecture actually sought of you need to have it back to back you will need all concepts of decision trees when we migrate over to random forest as well.

(Refer Slide Time: 0:49)

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

NOT ABOUT WALKING IN A FOREST

Decision Trees and Random Forests [Debdoot Sheet] 2

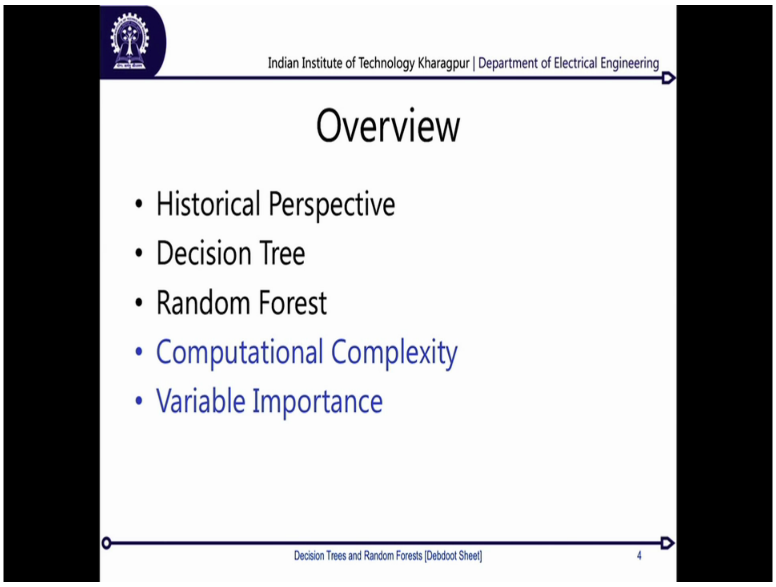

Indian Institute of Technology Kharagpur | Department of Electrical Engineering

IS ALL ABOUT

Decision Trees and Random Forests [Debdoot Sheet] 3

So, to start with it is obviously it is not about J walking in a forest as you see over here and it is actually all about these kinds of structures, what we are looking at. And each of them is actually called as a decision tree and if you collate all of them together it is called as a decision forest which is from the fact that lot of trees actually make up a forest. So without much of an ado, how this is organized is I would be speaking a bit about the historical prospective of what came first and how the whole field was involving into decision trees and from then eventually to random forests.

(Refer Slide Time: 1:16)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Overview

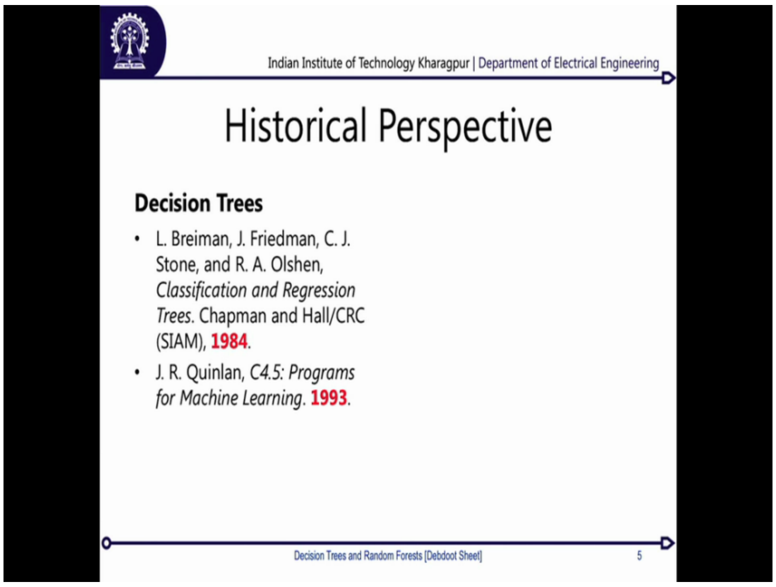
- Historical Perspective
- Decision Tree
- Random Forest
- Computational Complexity
- Variable Importance

Decision Trees and Random Forests [Debdoot Sheet] 4

And then I would be speaking about how you build up your own decision tree. So this is where we do the math which goes behind creating them which will be forming as the basis for us to move on next when we look into the coding stuff as to how can we actually get the problem solved with using these decision trees. So from there we move on to the random forest module and then we discuss about some other very engineering complexity perspectives and one of them is the computational complexity which everybody is obviously worried about whether you are say (()) (01:45) laptop can handle the same kind of a problem or you need a very specialized hardware in terms of the high performance computing or supercomputing to handle them.

And the other factor is some of you might have been aware of fact called as feature selection which is quite common in the community in order to reduce down the features set or sometimes you work on a stuff called as feature compression. So random forests and decision trees together both of them have an inherent problem of eliminating useless features from them and that is why we discuss about variable importance and how you can use that in order to speed up your whole algorithm development process.

(Refer Slide Time: 2:28)



Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Historical Perspective

Decision Trees

- L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Chapman and Hall/CRC (SIAM), **1984**.
- J. R. Quinlan, *C4.5: Programs for Machine Learning*. **1993**.

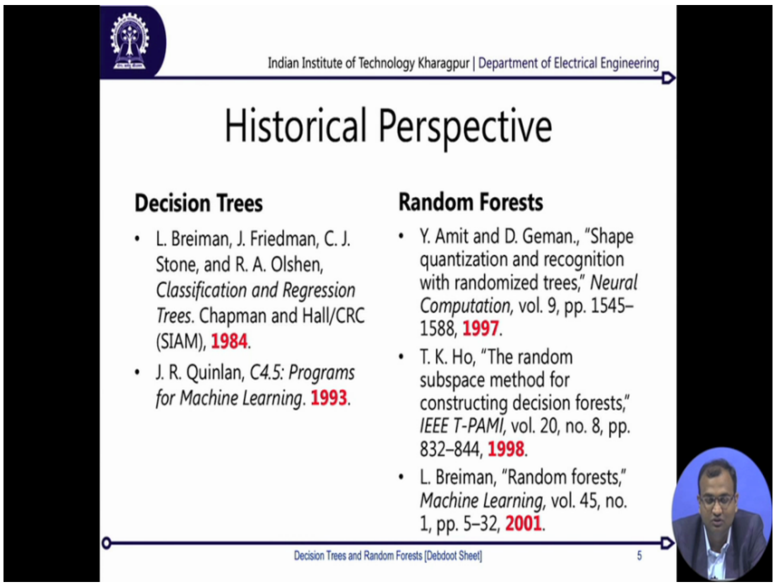
Decision Trees and Random Forests [Debdoot Sheet] 5

So coming down to the first part of it is on historical perspectives. So decision trees typically came up with these four famous people called as Breiman, Friedman, Olshen and Stone and they wrote down a very famous book called as classification and regression trees, which comes down also as acronym called as cart, you might have heard about this one long back.

So this was a work which was funded by the US neighbour research board somewhere in early 80s and the book came around 1984 and interestingly this whole machine learning paradigm got developed for solving medical problems. So today you have clinical decision support systems one of them is Oncosyn there is (()) (03:06) and these are ones which a lot of clinicians and in fact, there are certain websites where you can in fact, go on and put down your symptoms and how you are feeling in order to see whether you have a particular disease, all of them make use of rule based diagnosis which is based on what is called as classification and regression tree.

So this is how you can automate rules generation, the whole process of automated rule generation is what happens within them. So that is what came down in 1984 and from there (()) (03:36) goes experimenting on making them much better and we had programs called as Itredive dick atomizers and how they would be working out. So C4.5 is much stable release and the one which forms as basis of all stuff which we do today which came out in 1993.

(Refer Slide Time: 4:00)



Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Historical Perspective

Decision Trees

- L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Chapman and Hall/CRC (SIAM), 1984.
- J. R. Quinlan, *C4.5: Programs for Machine Learning*. 1993.

Random Forests

- Y. Amit and D. Geman., "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, pp. 1545–1588, 1997.
- T. K. Ho, "The random subspace method for constructing decision forests," *IEEE T-PAMI*, vol. 20, no. 8, pp. 832–844, 1998.
- L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

Decision Trees and Random Forests [Debdoot Sheet] 5

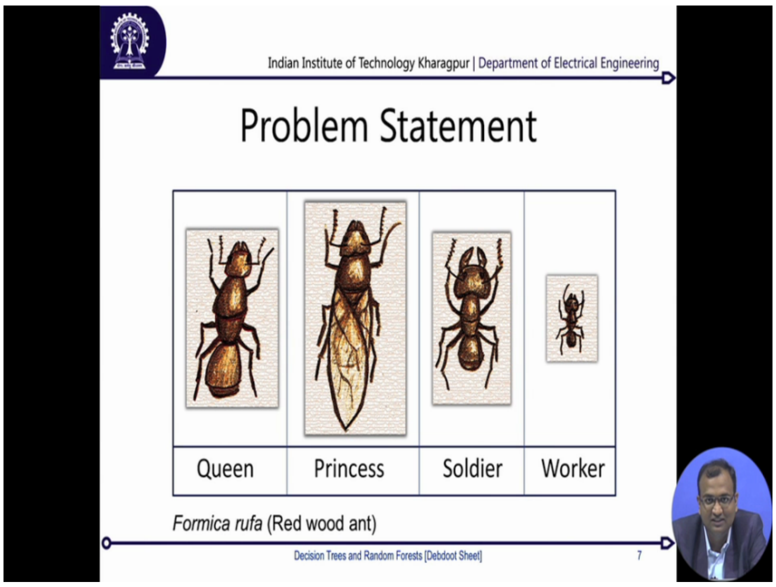
So looking from there from 1993 till 1997 it was an age which was more of govern by support vectors and you had an onset of adaptive learning techniques using boosting as well, but then around in 1997 came down this very simple idea from Amit and Geman and this was about can we create a bag of trees which is instead of using one single decision tree can we randomly create multiple number of decision tree.

So they observed one single thing was if you are sampling down in a very different way than you can actually have different kind of decision rules being created over there which will make independent trees. So if it is an independent tree and multiple of them then can we have a democratic voting which is gives the same importance to each of them, but you listen to everybody's decision over there.

And from there came down this concept about random forest. So it was quite good, but in 2001 was when Breiman came with this term the same old Breiman from who invented cart gave a very formal name called as random forest and same random forest which we are using today. So from then on this has actually taken the community with drive, you have lot of consumer grade devices including gaming platforms like Microsoft kinect for xbox which uses them for real time processing of your body movements when you are doing body driven and gesture driven gaming over there. So from that medical diagnosis it is in a heavy use. We will be studying few of these scenarios, but let us start with the basics first. Now a decision tree, obviously you see one particular name over there which is tree it does not look like the banyan tree or bamboo tree which is just outside but it is some sought of a tree.

Now to bring you to your very basic concepts about when you are doing programing and data structures in your early days, you had have sought something called as binary tree. And a lot of you might have also implemented a binary search tree. So we use a similar concept over here except that instead of searching it will now be looking at the left or right node and there would be taking a decision a yes or a no very much intuitive to how as humans we take decision we take when we are pleased with very complex scenario.

(Refer Slide Time: 6:14)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Problem Statement

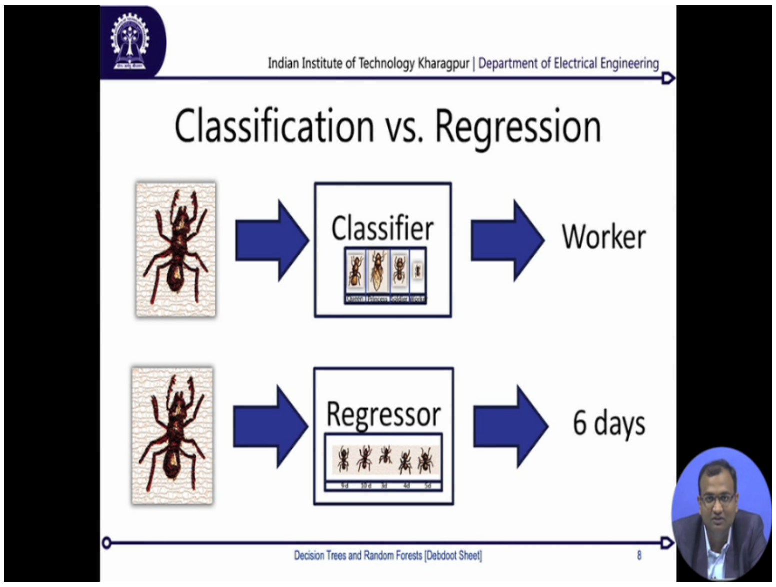
Queen
Princess
Soldier
Worker

Formica rufa (Red wood ant)

Decision Trees and Random Forests [Debdoot Sheet] 7

So let us start with a simple problem from taxonomy. Now you might have seen red wood ants, so they are those big thick red ants which are on trees and generally you would forbid that they do not bit you otherwise it is seriously painful because of the formic acid they contain. Now these ants are basically of four types so that is called as taxonomy where you are going to classify which type of ant they are. Now to a inexperienced eye if you just look at ants they are just ants, but then they can be workers, soldiers, soldiers are the most furious one which actually bite, princess you would rarely find them, queen it is even more rarity to find the queen. Now, if we start with this problem of these ants and let us take two different typical scenarios.

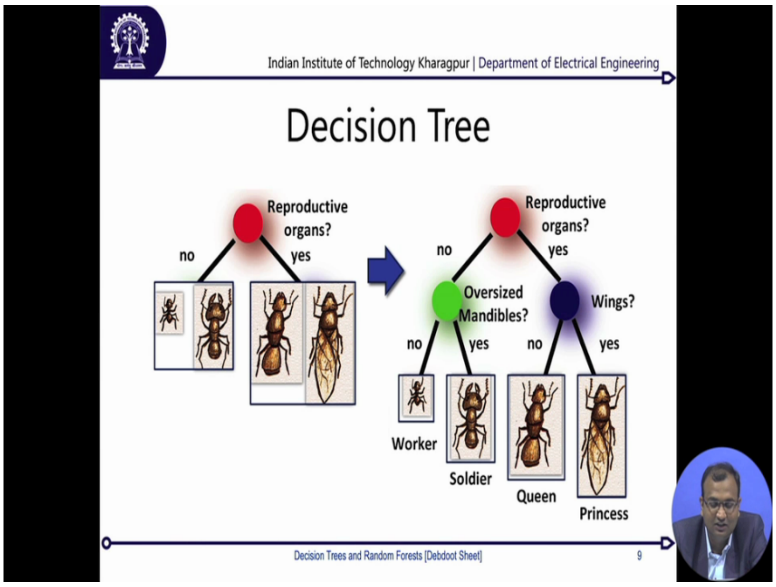
(Refer Slide Time: 7:05)



Indian Institute of Technology Kharagpur | Department of Electrical Engineering
Classification vs. Regression
Classifier
Worker
Regressor
6 days
Decision Trees and Random Forests [Debdoot Sheet]
8

One of them is classification scenario in which given an image of the ant I want to find out what kind of an ant it is whether it is a worker, it is a soldier, it is a princess, or it is the queen. The other problem is given the picture of a ant, I want to find out what is its age. So a worker which is smaller in size, so much aged worker might be similar in height and length to a much younger soldier or to a much younger princess. Now how do we do it, so this is about classification versus regression, regression is when you are trying to continuously find out the age which is a continuous variable not a categorical one and classification is when I want to find out which of these four classes it is coming up. That is a category classification problem and that is a classification.

(Refer Slide Time: 7:54)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering

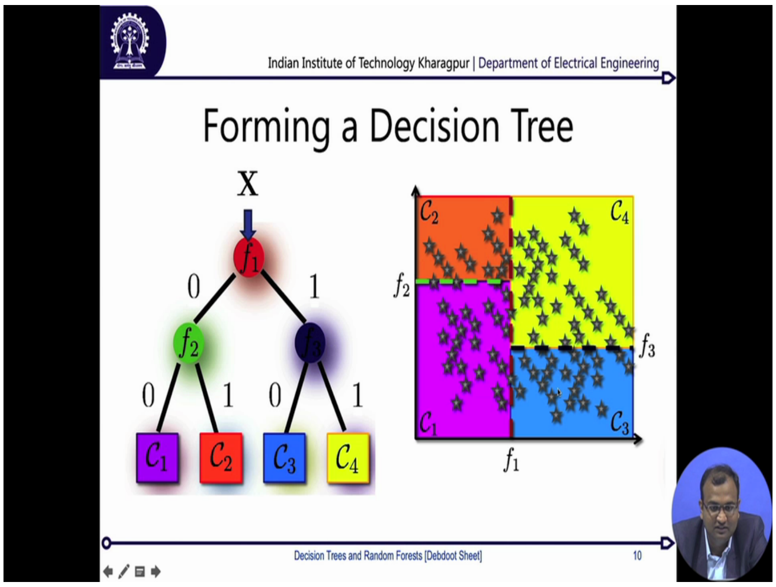
Decision Tree

Decision Trees and Random Forests [Debdoot Sheet] 9

So let us look into the first problem which is about can I find out whether it is a worker, or it is a soldier, it is a princess, or it is the queen, ok. So let us take an image and on that image of the ant let us see if on the image I can see down reproductive organs or no, so if I do not see reproductive organs, or I would see reproductive organs. So there would be two of these images on which I can see two of them on which I will not be able to see. This is the first level of decision which we take.

The second level is if there are no reproductive organs then let us look at the mandibles which are the tooth of the ant. So they might be oversized mandibles which are seriously big appear that would belong to the soldier, so we go down to this part. Or there might not be oversized mandibles and they would be workers, ok. Now if they have reproductive organs we ask a different question, we ask a question that does it have wings or not. If it has wings then it is a princess, if it does not have wings then it is a queen. So basically you see that this structure represent some sought of a tree and you are taking just yes or no decisions which are binary decisions at each level of the tree.

(Refer Slide Time: 9:07)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering
Forming a Decision Tree
X
f_1
f_2
f_3
0
1
C_1
C_2
C_3
C_4
f_1
f_2
f_3
Decision Trees and Random Forests [Debdoot Sheet]
10

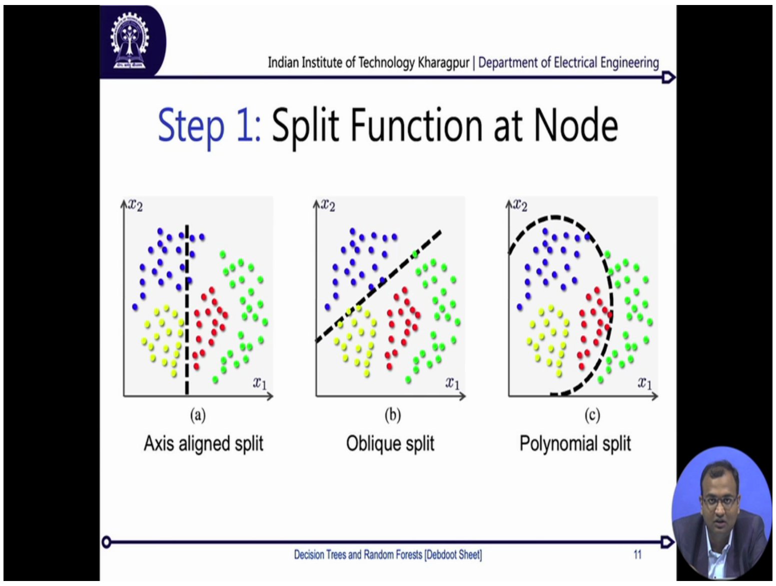
Now from there we come down to how we form a decision tree, ok. Here what you would be typically taking is you take down an image of the ant say that is our random variable X, ok. We put down that X on to the tree and the first node so it will look into all factors around X and ask one simple question. So X might have all of this information, whether it has oversized mandibles, whether it has reproductive organs, whether it has wings or not, and you will having certain yes or no or there may be soft variable inputs over there. Now, it asks this first question, now based on the first question it will either get a no or it will get a yes. Eventually on the no part there will be certain samples which will travel down on the no part, then I ask another question based on whatever information is present and I take a yes or no.

Now each of these circular points over here is called as decision nodes, this is where the decision is taken whether to go left or right. And each of them the square ones are called as the leaf nodes or this is where the decision stumps press. So if you come down to one of these you are at one of these classification classes, so this is how the whole process goes down.

Now in terms of mathematics we can put it something in very simple terms. So say that I had a feature space in which I had two features f1 and f2. And across this f1 and f2, I would be able to somehow draw certain boundaries over there. Now the first one is the red boundary which corresponds to this first decision over here, ok. Now based on this I can put it down into either left or right so just have a line equation I can just drop any point on this space project it on to the line equation I either get a positive sign or a negative sign, based on that I will be going either to the left or the right.

So based on that if I want the left I am at C1 and C2 which are these two, then I go down to the next decision which is at f2, at f2 I would be asking another question which is based on this second feature over here. And then I can classify them into C2 or C1. Similarly I do it for C3 and C4 and this is how I can split down my whole observation space into different kind of classes with different decisions.

(Refer Slide Time: 11:38)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Step 1: Split Function at Node

Decision Trees and Random Forests [Debdoot Sheet] 11

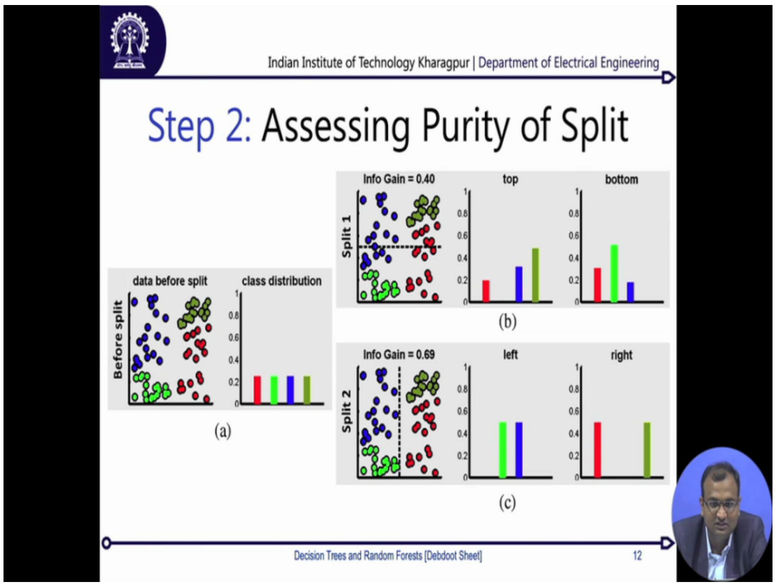
Now the question is, how I split them, how do I use what feature and how will I select which value of the feature to be. So this is an iterative process, which gets solved while creating this decision tree. So the first step is I need to find out what is sort of what we qualify as a split function. So is it basically a line equation, which is straight line so there can be these kind of straight lines which are align to an access. So this can either be a line to one of these orthogonal accesses, for a 2D case it would just be align to one of these accesses.

Say I have a 3D problem; I can have an axis aligned plane over there. So either it is align to the X-Y plane, or it is align to the X-Z plane, or it is align to the Y-Z plane. So there can be this kind of one’s which are align to each axis. There can obviously be oblique split as well so this can be a Y = MX + C kind of line equation, which can exist not necessarily along one of these axis but they can exist on a 2D space as well.

The other kind of thing can obviously also be a polynomial so you can define some sought of parabola, or a curve, or some sought of an ellipse function over there and this can also be used to segregate into different one. So you can see a typical example for each of them they have their own beauty and merits and demerits together. So if you look at this polynomial split you are very easily able to segregate this green class from the red, yellow and blue.

So even if I am using the straight line I can get the yellow and blue and red but then eventually on this side of it when I want to segregate the red from the green, I might have to use this kind of a polynomial split so that I get a perfect splitting condition coming up. Now, there are pros and cons and how we do it which we would eventually come down in the next subsequent steps.

(Refer Slide Time: 13:12)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Step 2: Assessing Purity of Split

Decision Trees and Random Forests [Debdoot Sheet] 12

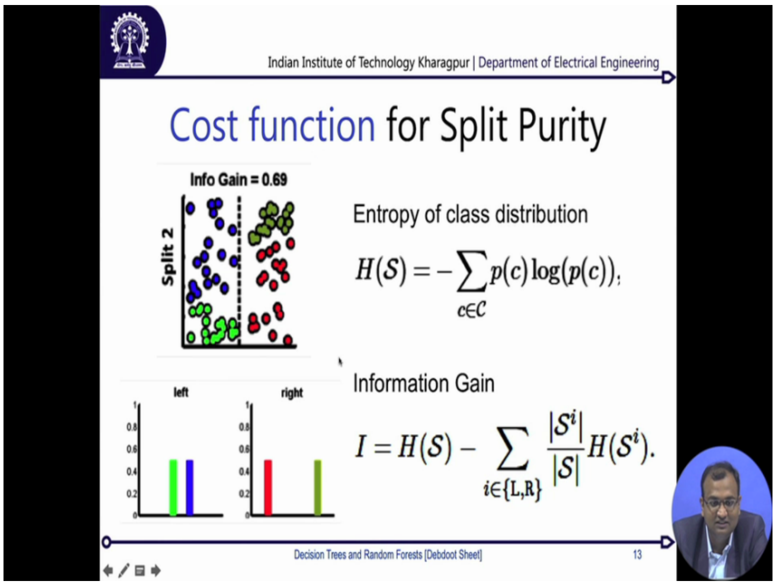
The next part is that if I am splitting the question comes is to how do I assist that I am splitting it properly and that is what is called as assessing the purity of a split. Now, in order to assist the purity of a split what we do is let us say that I have this data which is coming down at one of my nodes or at the first node. So I have different classes over there, I can just plot down my pdf of each of these classes, so you can count down and see that this pdf perfectly matches down to 0.25 because each of them has equal number of samples coming in.

Now, this is for each class, so the number of samples per class divided by total number of samples present in this space. Now I can take one of this split in this first condition which is a split which is horizontally aligned over here. Now if I do that I will be getting down a top and a bottom part. So on the top I can observe my histograms again, so I will be computing my pdf on this top over there, so there are some samples from red class all the samples from this yellow class and a majority of the samples of the blue class which come over there.

Now I look at the bottom part, over there I will be having samples from there will be no samples from the yellow class but all the other three classes are present over there. Now say I do not take this split but I take another split which is just this vertical, which is it can very accurately without making any errors segregate between two classes. So you club down all the blue and the green classes together and you are clubbing down all the yellow and the red classes together.

Now, typically if I ask you a question as to which one you would prefer taking, the intuitive answer would be, let us go split 2, because I am able to segregate between 2 different classes. So my first question on the ant problem which was just looking whether they have reproductive organs yes or no, I could club them into soldiers and workers and into princess and the queen. So it is a similar kind of problem which resolving over here as well.

(Refer Slide Time: 15:16)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Cost function for Split Purity

Entropy of class distribution

$$H(S) = -\sum_{c \in C} p(c)\log(p(c)),$$

Information Gain

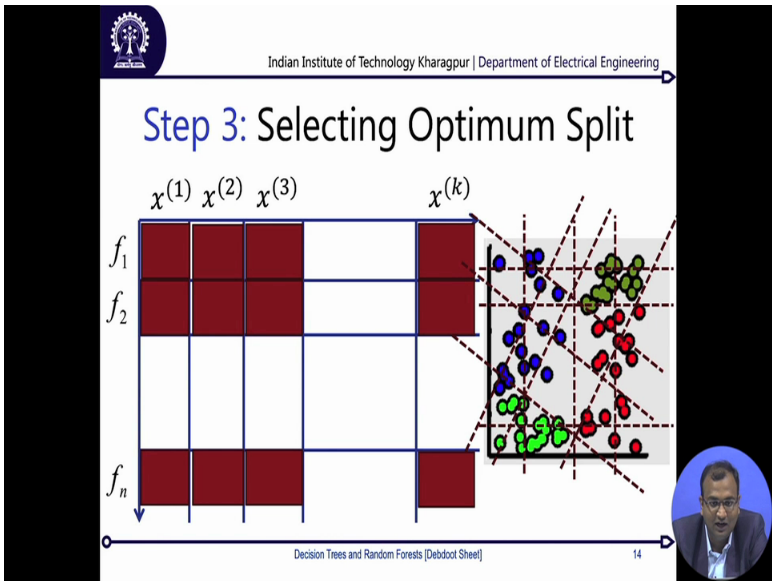
$$I = H(S) - \sum_{i \in \{L,R\}} \frac{|S^i|}{|S|} H(S^i).$$

Decision Trees and Random Forests [Debdoot Sheet] 13

The question comes that this is a mathematical framework I just have features and their values, so how will I get, how will I say that this is a perfect one which I want to do and that is where we use a concept called as cost function. And what this does is this assist is what is the cost associated with splitting something properly. So either the cost might go up or the cost might go down. So based on what sought of the cost function you are using now what we use is called as a information gain, to the concept comes is that if I have done this split I should be having very pure classes or my entropy should be going lower over there. So we use this particular information gain index over here, which is just a ratio matrix summation over the total information which is being gain.

And what you do is typically if you look at these classes left and right and so on the left side there will be just two classes on the right there will be again be just two classes and if you look at their probabilities there is somewhere around 0.5 each of them. Now this is where the information gain happens over there because you do not find all the other classes you just have two classes out of four classes which have a probability of 0.5.

(Refer Slide Time: 16:25)

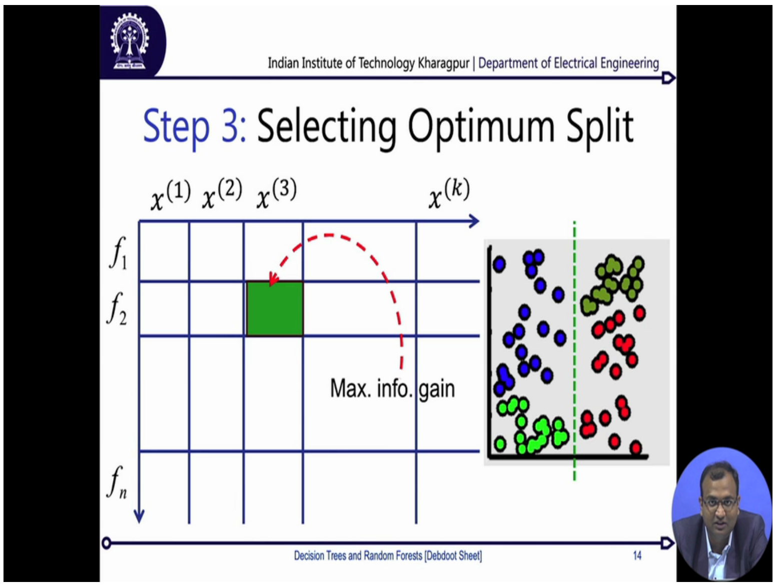

Indian Institute of Technology Kharagpur | Department of Electrical Engineering
Step 3: Selecting Optimum Split
$x^{(1)}$ $x^{(2)}$ $x^{(3)}$ $x^{(k)}$
f_1
f_2
f_n
Decision Trees and Random Forests [Debdoot Sheet]
14

So you obviously came down with a much better entropy over here and entropy which is much closer towards 0 then in the earlier case. Now from there, the question is that I have two problems to solve one what is the feature I am going to use the next question is which value of the feature I am going to use. Now this comes down as a joint optimization problem and that can be solved out very easily suing this simple cost function which we have.

So what we do is say that I have a k dimensional feature vector so from x(1) till x(k), ok. Now for each of these I can take certain number of split points called as f1 to fn. So my x(1) can range in the range of say - 100 to + 100, I can take n number of such say n is 50, 50 some random points over there, x(2) might range from 0 to 1000, I will take f1 to fn which are some random number of points only on x(2) which will be in a value from 0 to 1000.

My x(3) might range in the value of 0.5 - 0.5 to + 0.5, my x(k) might range in the value of 100 till 1 million and I would just be taking random number of n samples from there. So it is not necessary that the same values are taken for each vector, they will just be randomly picked up within the dynamic range. Now if we do that for each of this features and each of these values I will be able to design a split. So whether the feature value is greater than or less than that and based on that split condition I will be able to compute my information gain. Now this gives me a 2D matrix over here, so for the first feature and for the first value I get an entry, for the second value of the first feature I get another entry. For the nth value of the first feature I get an entry, for the second feature I take the first value and get an entry for second value of the second feature I get an entry. And similarly this keeps on repeating and I would be able to fill up my whole matrix over here. So you see that I am drawing down different splits over there and appropriately I am getting down an information gain which I fill within the matrix. Now out of this whole matrix there would be one point where I would be getting the maximum value of information gain coming down.

(Refer Slide Time: 18:27)



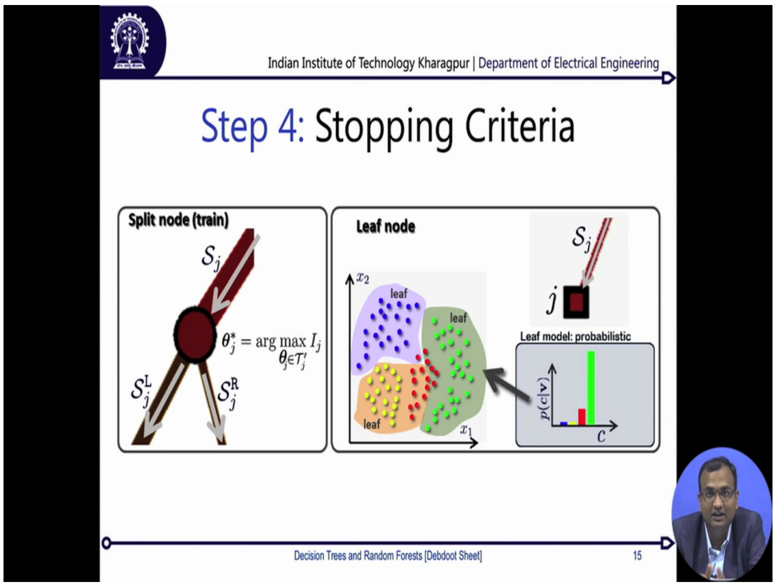
Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Step 3: Selecting Optimum Split

Decision Trees and Random Forests [Debdoot Sheet] 14

And this maximum value of information gain will be the point which gives me an optimal split. So typically this is where it is just a maximization finding within a 2D matrix and a very simple problem. So you are solving both the problems together so what value and which feature to use together to split in one single shot over here.

(Refer Slide Time: 18:52)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Step 4: Stopping Criteria

Decision Trees and Random Forests [Debdoot Sheet] 15

Now from there we go on to the stopping criteria, which is I cannot keep on splitting forever long. So at some point of time I might end up that there is only one sample at one node. Now I do not know what to split over there, so obviously it is going to be that particular class. Now if we keep on doing that infinitely then we would actually land up into getting down the number of nodes = into worst case scenario the number of nodes will be equal to the total number of samples which are present over there.

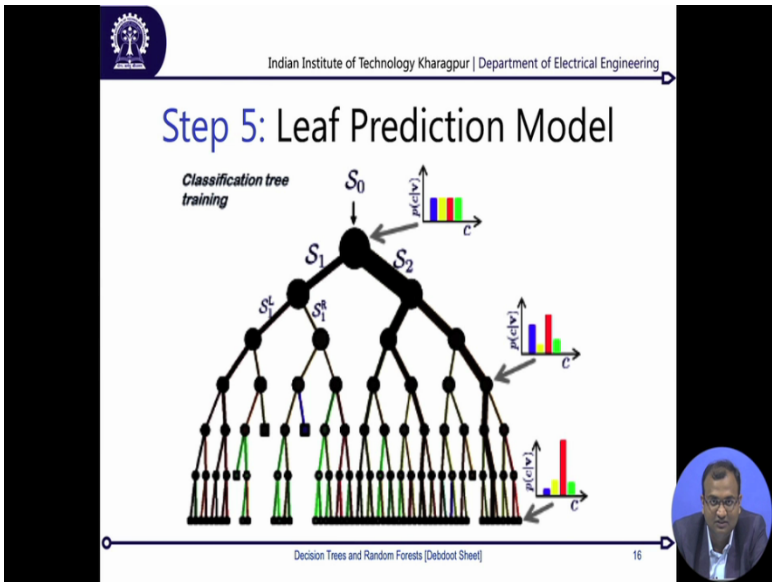
So say in your training you have 1 million samples which is not so hard to imagine as such because a 1 megapixel image would actually have 1 million points over there. So if you build a 1 megapixel image without any stopping criteria then you would actually be ending up getting the in the worst case scenario 1 million such leaf nodes which are associated with the decision.

Now instead of doing that what we do is, we set a criteria that if the number of samples which come down to a node false below a certain number, I just stop over there or otherwise if the information gain over there is not substantial or say the entropy is very close to 0. Say the entropy somewhere in the order of 10 power - 3, then I would be stopping it down, which means that there are some other classes present in those samples as well but they are negligible as compared to the majority of the class.

So this can be two different criteria which we can use. So before you start splitting down a node since you will always have to compute the entropy before split so once you have that computed you can use that as the stopping criteria to decide whether I want to use any splitting further or not. So this is how we can stop down, there are obviously many other ways so sometimes trees are created so this depends on what kind of packages and implementations you are using.

There are certain implementations which would offer you that you can grow trees up to a fix depth only, so beyond that depth you will never be able to grow your tree any further. So there are multiple facts which play around over there. Now from there still step 4, we could use our training samples and create the whole tree. Now the question comes as if I want to predict out of this tree which is on the testing side where I do not have any more class labels given down to me then what I would do.

(Refer Slide Time: 21:04)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering

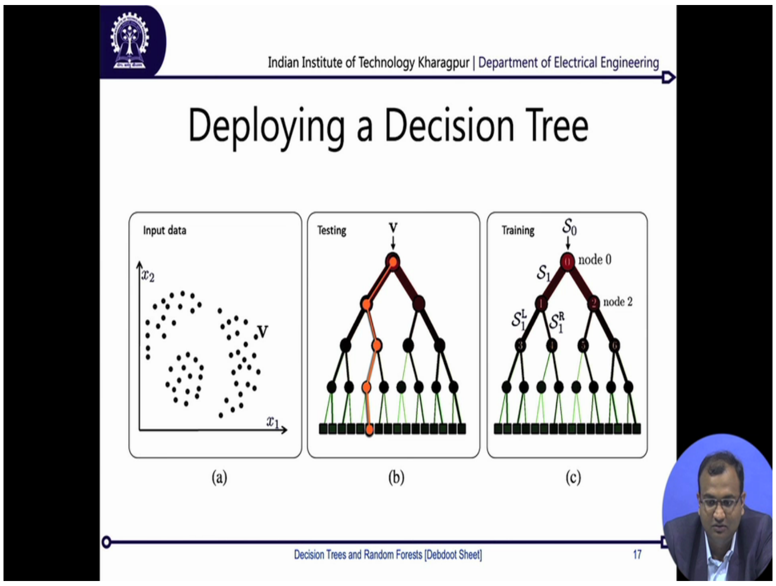
Step 5: Leaf Prediction Model

Decision Trees and Random Forests [Debdoot Sheet] 16

So I put down on my first mother node which is the 0th level node the sample. Now, it will use one of the features over there which is on my split condition and then do a yes no split based on that it will be traversing. So if we look typically over here, this is how it would travel. So for this particular example it goes down to this one, then till here, then here, here, here, here, so this collided nodes and as you see it going down over here, this is just the relative probability distribution across each of these nodes.

So what it shows is that this is the probability of samples which flow down across these ones. And this was computed during the time of training, because here I do not have a label associated as such, so I can never compute this one so during the time of training I can always compute this and (())(21:47). And you see that this is where it was not at all pure, there is a high probabilistic chance so basically any class has a probability of getting classified as a one fourth or 0.25.

(Refer Slide Time: 22:16)


Indian Institute of Technology Kharagpur | Department of Electrical Engineering

Deploying a Decision Tree

Decision Trees and Random Forests [Debdoot Sheet] 17

This is where it has comes down to a very high probability of belonging to this red class. So this is how the whole purity comes down in a decision tree by just doing yes no, yes no kind of questions. Now from there is the question of how do you deploy a decision tree. Now when we are speaking in terms of deploying a decision tree, the fun comes is that you have an input data space and none of them is labeled right, now what do I do.

So during testing what I would do is during training obviously I got down my samples over here. During testing what I do is I pick down so I can either randomly take down from this space from I want to test or I can take in order because in any case I will have to test down over all the samples present over there. So I take one sample I pass it through over here, it does this (()) (22:48) so if you remember your binary decision tree programming so at each node you can obviously take a left or right traverse decision over there.

So either a left traverse or a right traverse and then based on that you end up going to another node and on that node you again do a left traverse or a right traverse. So if it is a left traverse you go down to the pointer which points to the left traverse will do the next node over there and eventually and the leaf node you will have a null pointed to any further and there would be a class probability and this probability is actually during training purpose whatever was the posterior probability of each sample coming down.

So the posterior probability is the number of samples for each class which landed up on that particular node divided by the total number of samples which ended upon that node. So if it is a true classification say for the red class then red class will have a posterior of 1, other will all have a 0. If it is somewhat so if there might be impure classifications as well that where red class might be 0.98 and all the other 3 classes will be 0.01, there might be 0.005 and 0.005 respectively.

So you would be getting down this posterior probabilities over here as per your based (()) (23:58). And during training what happens is you actually create down the whole tree and the whole traffic passes down the whole thing. So this is the basic difference in how you would be doing it, so eventually in the subsequent ones we do about random forest and I discuss about certain properties of random forest. We would also be covering one simple code (()) (24:18) example where we show how we are going to train this whole model. So with that I would be concluding on decision trees and we wait for the next one on (()) (24:28) thank you.