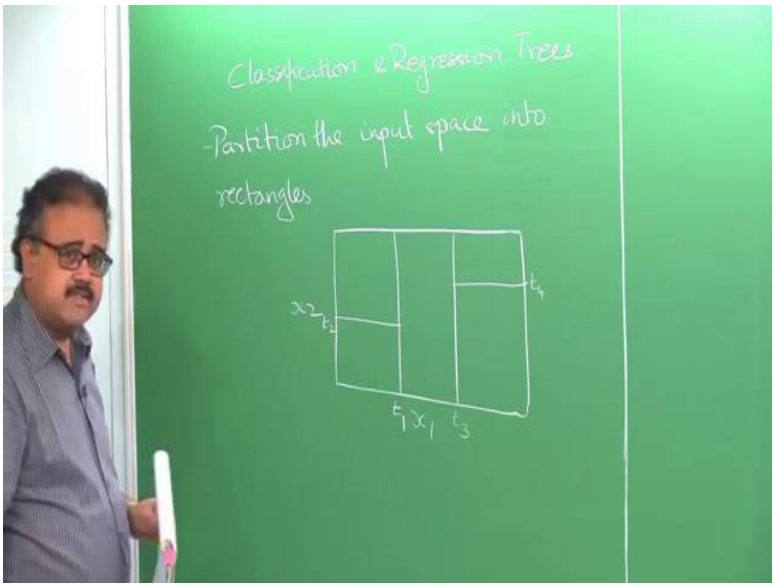
Introduction to Data Analytics
Prof. Nandan Sudarsanam
and Prof. B. Ravindran
Department of Management Studies
and Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 05
Lecture – 25
Classification and Regression Tree

Hi and welcome to this module on Classification and Regression Trees. So, today we will look at a very simple, but powerful idea for building a both classifiers and regressors.

(Refer Slide Time: 00:13)

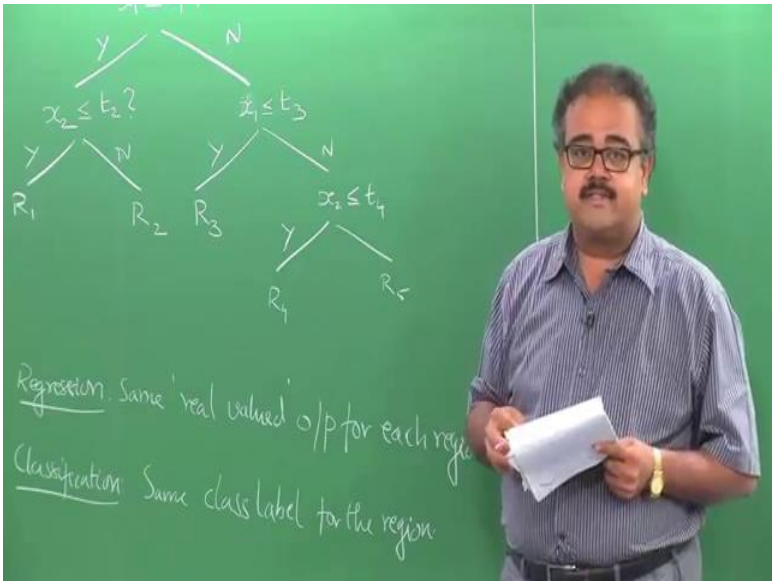


The basic idea is that, you are going to partition the input space into rectangles. So, let us imagine that you have a two dimensional input space x_1 and x_2. So, you are going to try and partition this into rectangles by drawing axis parallel lines. So, why are we drawing access parallel lines here? Because, these lines can be with specified very easily by just comparing against one of those dimensions of the input data. So, for example, to draw this line all I need to specify is the intercept at the x_2 axis.

So, likewise to draw this line I need to specify the intercept of the x_1 axis. So, one way of thinking about these kinds of partitioning of the input space for using axis parallel lines is to think of making a series of decisions as to, which side of a specific line is your data point line. So, you can think of this as following, we will call this say t_1, this point is t_2,

this point is t_3, this is t_4. These are the intercepts along the respective x_1 and x_2 axis.

(Refer Slide Time: 02:52)



Then, one way of representing this is to think of this as a series of tests or decisions that you are making, so I can start of by asking the question, is $x_1 \leq t_1$. So, that is essentially saying here is a line that represents $x_1 = t_1$ that is your data point lie to the left of the line or to the right of the line. So, if it lies to the left of the line, so this will be an x, then I ask the second question, which is essentially is $x_2 \leq t_2$.

So, $x_2 = t_2$ is this line and I am asking the question if the data point is above this line or if the data point is below this line. So, if it is below the line, so then I get a yes for this question as well and I will denote this by R_1, so this region is R_1. So, since this represents both x_1 being less than t_1 and x_2 being less than t_2, it is essentially bounded in this region. So, likewise if x, if the point is actually greater than t_2, so in this case this will evaluate to no and say I get to a region, which is called R_2.

So, what would this region be if you can think about it? This is essentially the region, where $x_1 > t_1$, but $x_1 < t_3$. So, I am going to call this region R_3, so here is x_1 is greater than t_1, but $< t_3$, so that would be R_3 and if $x_1 > t_3$, so in this case I am again splitting in to two regions. So, essentially, so I am testing on x_2 and t_4. So, what is that you notice about this tree that we have drawn here?

So, every point I am asking you a binary question, is this yes or no. So, essentially it is a binary tree, at every point you divide into two branches. And once I have divided into one of these branches, once I go down one of these paths I am, from here on I am only

concerned about data points that has satisfy the first question that I asked. So, from this point on in the tree I am only worried about data points that are to the left side of the line here.

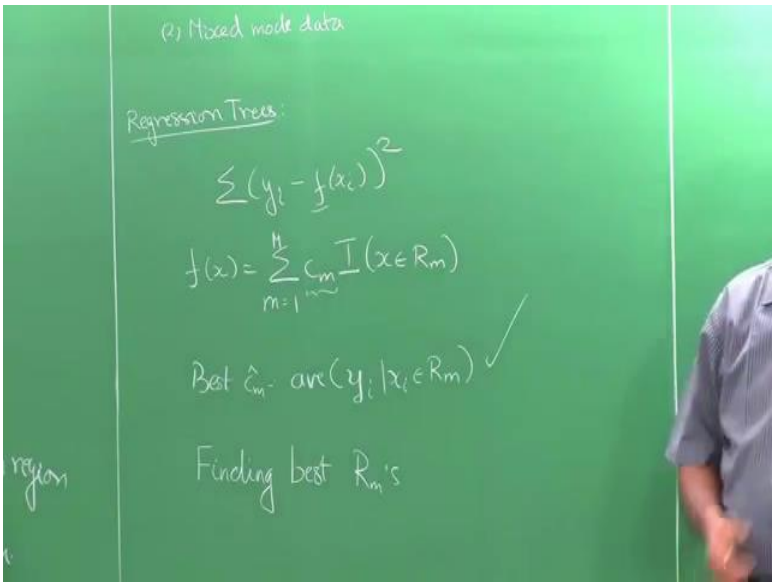
At this point in the tree I am worried about data points that are to the right of the line here. So, that is a couple of distinguishing features of the decision tree that I am making binary decisions and I am also looking at some kind of a divide and conquer approach great. Now, what we have done is that we have a representation that allows us to split the input space into different regions.

So, depending upon the kind of problem that we are solving whether it is a classification problem or whether it is a regression problem, so we would like to fit a single value to each of this regions. So, if it is a regression problem, so regression problem we will be outputting a single value for the entire region. So, if your data point falls in R_1 regardless of, where in R_1 it is falling, it is so the data point could fall here, it could fall here, it could fall here, it could fall here regardless of where in R_1 the data point lands up, I am going to predict the same output.

So, it would be the same real valued output for each region, so in the case of classification, what you expect it to be, it will be the same class label for the region. So, regardless of where in R_1 the data point falls I will always output the same class label for the classification problem. So, now, so we have two questions that we have to answer in the case of decision trees. So, the first one is how do we form the regions and the second question is, having formed the regions, how do I decide what is the output that I am going to produce for that region.

So, we will look at each of these problems in turn, but the first thing if I wanted to mention before I go on to look at, how we solve them is that decision tree is a fantastic, because they are the most interpretable of all of the classifiers that we are going to look at, even more so than linear regression at some point. Because, if we think of the way we constructed the decision tree, it seems like a very natural way to map it to how humans think about making decision.

(Refer Slide Time: 08:56)



So, that way the interpretability, so the interpretability of decision trees are very high and in fact, that makes it one of the classifiers or regressors of choice in a very wide varieties of problems. And the second advantage of decision trees is that they can work well with mixed mode data. So, here the example I gave you assume that x_1 and x_2 are actual numbers and you could pick arbitrary comparison points x_1, x_2 need not be numbers they could be a categorical variable like color or it could be age, but represented as young old and middle age.

It did not necessarily be a number on which, you have to run this kind of test I could compare whether the color is red or not red or I could look at whether the person is young or middle aged verses the person is old. So, I could have any kind of binary test among categorical attributes and then, I can still construct the decision tree. So, the first advantages one of interpretability the second one the, which you can hand mixed mode data.

So, now, let us step back and let us look at regression trees specifically, so what we know about regression. So, regression the goal of regression is essentially to minimize some squared error. So, this is one of the goals of regression is to minimize some squared error will stick with that of course you can build regressors for whatever objective that you want optimize, So I want to fit a function f says that I minimize this some squared error.

Let us, suppose that I have a tree that has split by input space into m regions, which I denote by R_1 to R_m, so I have m regions in the input space. And then, for each of these

regions, so I am going to output a specific value, which I denote by C_m. So, C_m is the value that have output for any data point that lies in region R_m and I here is an indicator function that denotes whether the data point lies in region R_m or R_0.

So, essentially, what this summation tells us is that if the data point input data point x that is come to us is going to lies in one of these regions 1 to m, 1 to capital M. I do not know, which region it is going to lies in, but this indicator function will tell me, which region it lies in. So, this summation will be non zero for only one term essentially the region in which, the data point lies in. And therefore, the output will be the C value corresponding to the region in which, the data point lies in.

Suppose x lies in R_2, then the output will be C_2, suppose I am given this tree already this region split has been decided for me, then we know what is the best value that we have to output for C_m, so what would be the best value you have to output for C_m. Essentially I go through my training data I pick out all those x is, which lies in the m^{th} region pickout all the x is that lie in the m^{th} region look at the corresponding y and take the average of those and that will be the value that I output if the data point lies in the region R_m.

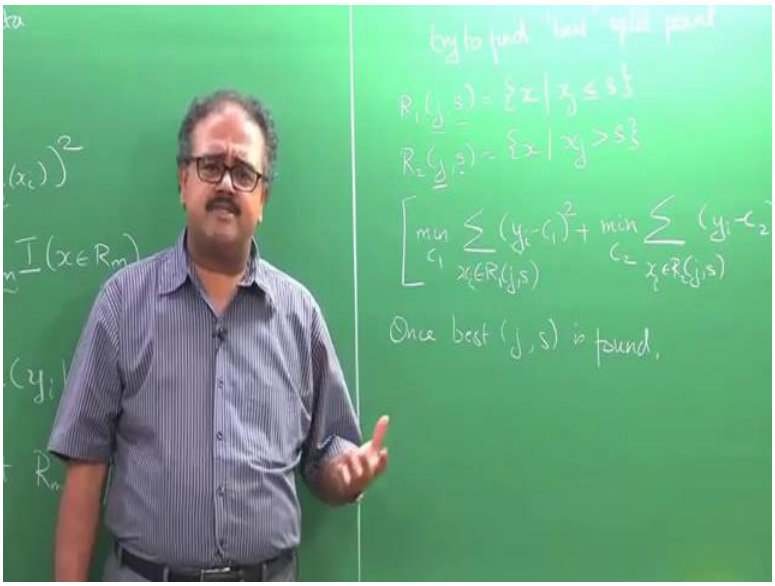
So, why is this a reasonable choice well, so one way to think about it is when I am trying to minimize the error in a specific region when I am trying to minimize the error in a specific region, let us say R_4 I do not have to worry about any of the training point that lie outside of R_4, because the value I have to predict for them is completely independent of right all these other data point. So, I only have to worry about the data point that lies within R_4 when I am trying to make a fit for the value there in output in R_4.

And among all the data points with lie in R_4 the best prediction that I can make is; obviously, the average in terms of minimizing the some squared error. So, if I have a different criteria let us say among to minimize the median I mean I want to minimize absolute deviation, then I probably have to predict the median not the average. So, this part is fine we know we are solved one of the two problems. So, what were the two problems one is given the region split, what is the output that have to predict for each region.

So, that we know how to do that at least in the case of regression, now comes the harder question, how are you going to find the regions, how are you going to find the best R_m's. In fact, finding the best R_m's finding the best region split is actually a combinatorial problem when it is actually infeasible and it is going to take a very, very long time to find

the exactly the right set of regions. So, quite often what people do is they adopt a greedy approach to finding this regions.

(Refer Slide Time: 15:59)



So, what is that greedy approach to, so you basically start of by considering basically start of by considering a split variable, so what is the split variable. So, in the case of the example tree here, so the split variable at this level is x_1 and the split variable at here is x_2 the split variable here was again x_1. So, essentially you consider some split variable and then, try to find try to find the best split point.

So, what is the split point again, so in the tree that you saw earlier, so in the first level the split point was t_1 likewise, so at each level we have to find out what is the appropriate split point is. So, let us take us a simple example, so I am going to define going to define two sub regions R_1 and R_2. So, R_1 is that part of the space, where the variable x the j^{th} coordinate of the variable x is lesser than or equal to some chosen value s. Likewise R_2 is that sub region, where the j^{th} coordinate of the variable x is greater than some chosen value s.

So, now, what we are really trying to do is trying to find j and s; such that we can solve for the best possible split just to give an intuition here. So, in this case think of the original data, so I have chosen a split point, which is t_1. So, the s in my choice is t_1 and this part is wherever $x < x_1$ was less than t_1 and this part of the space was wherever $x_1 > t_1$. So, in our new terminology this will correspond to and this will correspond to, so the one here is, because we are looking at x_1 the and the t_1, because that is the split point of

you are considering.

So, now, we have to find j and s both; such that this expression is minimize, what are this expression. So, we know this. So, C_1, is the prediction I am going to make if the data point lies in the sub region R_1, so y i, so these are all data points that lie in the sub region R_1, so prediction I make for this C_1. So, this is essentially the squared error for all the data points at lie in R_1 and this is like wise this squared error for all the data points at lie in R_2.

So, we already saw that, so we can basically find the C_1 that minimizes this error likewise find the C_2 that minimizes that error. Now, my problem is to find j and s such that this entire expression is minimized some on a little daunting in the beginning, but if you think about it is not that hard, why because we are operating with the finite training set, like the data that is given to us at the beginning from, which we are going to build this tree is going to be finite.

So, what does it tell us for every x_j that I can choose as my splitting variable there are only finitely many points at which, I have to consider a split. So, the essentially tells me for every j I choose there are only a finitely many s that I have to try, why is that because that can only be a finitely many different values that the variable, x_j can take in your training data. So, essentially what we do is that at every level in your decision tree you basically look at this expression for all possible split points for every possible splitting variable that you have in your data and then, decide on which is the best possible splitting point.

Once the best j and s is found, so what you do next you essentially go hide and split the data into two parts one corresponding to the R_1 of the best j and s and other corresponding to R_2 of best j and s. And now, you repeat this process in both R_1 and R_2. So, now, if you think about it a problem is become much simpler, because the ranges that you have to the number of data point that you are looking at this much lesser and, so likewise you just keep going until you come to a point where you are happy to stop.

So, this essentially I will stop here as in this module and the next module look at till when we will grow the tree and how are you going to handle the classification.