NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

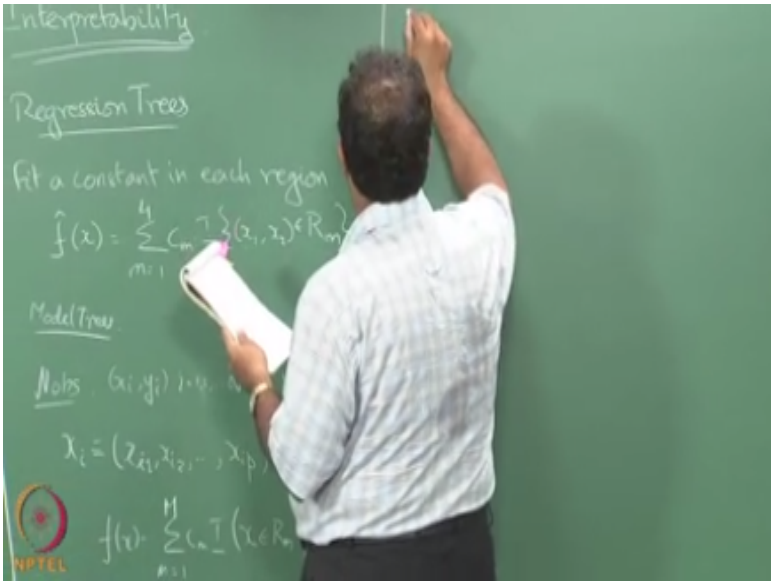
Lecture 40

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

Regression Trees

So as with the linear methods we will first start by looking at regression right, so we will see how to use decision trees for doing regression. So far I have just told you how to do the partitioning right.

(Refer Slide Time: 00:31)



Let us look at regression tress, so I split the region into four right, so I will first see if the data point that comes to me right, whether it lies in region 1 or region 2 or region 3 or region 4and for each of those regions I am going to have a some constant that I will output right, so if you think about it the function that I will output from here right, so we will have one value in this region right one value in this region right, another value in this region another while in this region so it will be like a piece wise constant thing right.

So people understand that right, you are not going to test my 3D drawing skills right, so you can see that there will be one output for any point in this region one output for any point in this region, one output for any point in this region, one output for any point in this sense and in some sense it is similar to KNNs, because you are assuming that there is a piece wise constant assumption about the function that we are trying to model, right.

By the second parametric or nonparametric, parametric okay, what are the parameters, so is it parametric resonance is depends on N becomes larger what happens, so we can apply one of those ideas here instead of fitting a piecewise constant per region you can fit a linear function on that region right, I have done the splitting right, so I am going to have some training data right, so some of the training data points will be here, some will fall here some will fall here and some will fall here I can take all the points in R1 and fit a plane to that.

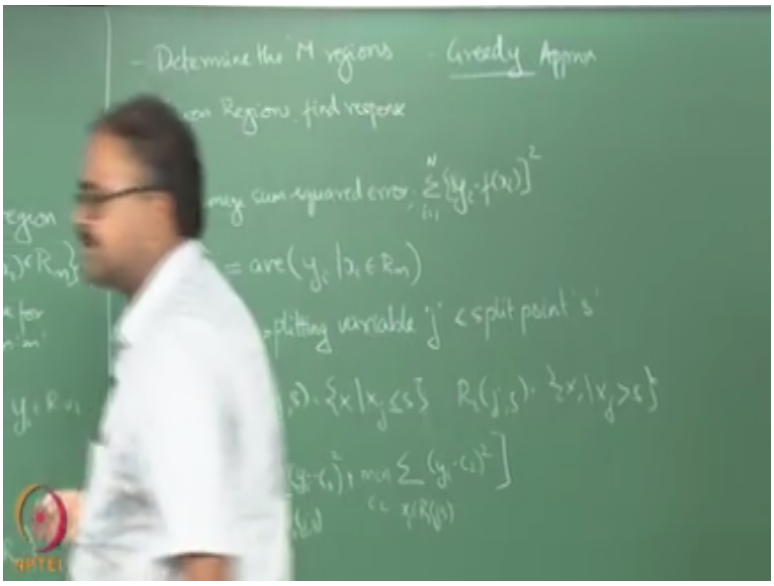
And I can take all the points in R2 and fit a plane to that likewise for R3 and R 4will that be better or worse than fitting a constant? Better. no actually it does not depend actually it is always better right, it is always better in the worst case you will fit a constant I mean if constant is going to be really better you will fit a constant because you are minimizing squared error and anyway end up doing that right.

So what is the problem with that little bit more work here that do more work and there is variance the significant variance we will come to the variance bit yeah, but the significant variance but such things are called model tree sometimes, model trees in fact you can do more complex stuff itself it does not have to be linear, a linear is easy right, I can do any kind of regression I want on this right, I can fit I can use a neural network if I want and to learn a curve only on the data points that lie in R1 right.

Not a good idea usually because I have already divided my entire training data into at least 1/4 if not smaller right, I mean some of the regions could have much smaller data some could have more right, but still I am cutting down on my training set that is available so it is going to be harder to train okay, so I am going to erase this stuff and try and generalize this or let me do it again. So I have n observations as you all know right, so this is our usual setting so where the input comes from Rp and the output comes from okay.

So the function that we are trying to learn is Cm and Rm right so sorry, the set of parameters that we have to estimate our Cm and Rm correct, so we need to know where are we splitting, how are we splitting the region and having found the regions right, what is the constant that we will fit within that region okay.

(Refer Slide Time: 07:07)



So there are the two questions determine the M regions and given the regions find the response right, but one thing to note is unlike KNN or anything the M is not given to you right, the M is something that you discover from the data and that is why I said it is nonparametric right, you can actually have more regions if the data requires it you can have lesser regions M is not given to a priori right, but sometimes as a regularizer you can decide to fix M as well you can say I do not want a tree that is more than four levels d as a complex d measure but again that is derived from the problem definition the model itself is not parametric.

So let us look at the second question first, because it is easier we actually have a proper answer to the second question right. Right, so this is essentially what we are trying to minimize right, and so we can try to do this region wise right, because the output that I produce for one region does not depend on the output I produce for another region so I can do this minimization region wise right, so I can pick yeah, every point would have it is own box kind of that. Okay, yes. Could likely yeah, so we will come to that will again address this question later, right.

Yes, so assuming that there is some amount of so you will not okay, let us step back into this you are going to get training data at best what you would do is you would have regions set within a region only 1.6 right, so that is not saying that I am going to fit it tightly around that point right, so all of our for that might be only one point but still there will be some kind of regional segmentation that is happening on the input data right.

Second I have I am going to introduce some kind of regularization that prevents me from doing that okay, so that will not be recap that is exactly what he was asking so you could end up with that that is what I am saying and we have to find some way of regularizing it so that you do not do that right. Right so if you going to minimize this region wise but anyway right now I am talking about given a region right so that is easy so we will not be over fitting things so given the region right.

So I am going to find out what should be the output of the mth region right, Cm is the output of the mth region right, that is what we assumed, what should it be give me a simpler I am fitting a constant I am not fitting a straight line here, average of all the points okay, like average of all the points which lie in the region and take the yi is corresponding to the points lying in the region and take the average okay, that is the best response that is that is easy okay that is done.

What is the harder part, finding the regions right, in fact it can be shown that finding the best possible Rm set of RM right is actually NP-complete right, and is NP-complete now in the again NP exactly NP-complete right, so you can show that finding the best possible Rm is very, very hard right, so we have to come up with some kind of approximation so essentially we use a greedy approximation. No, they just tell you what XII I told you I told you what the training data is right, yeah this is all the training data is you get xi, yi your job is to find the regions and find the region I find the response.

Yeah, yeah find the best region you can given it a region you can tell me what the performance is right, but then finding what the best such segmentation is actually hard you have to search through the combinatorial really many segmentations. So the way we do it is following right, yeah, okay, now for a given M so I want to find the smallest M such that I get that performance smallest region, smallest region yeah, the smallest region said for which I get the performance that is really ideally what I am looking for right, smallest M sorry, given an M finding the region yeah, in general that is also hard but I want to find it for the smallest M as well.

Ideally you want to find it for the smallest, ideally like there is some data if you are right, like you would have to either specify the M and then you find the best or you see that when the best and find the smallest M for which you can. Ideally it should be find the best and then find the smallest M for which you can do the best right, but we end up doing compromise on that as well so what will what we will do is you just making me go do this all out of order by asking leading questions.

But what we are going to end up doing is we are going to say okay, here is a greedy algorithm right, greedy algorithm find the best that the greedy algorithm can do okay. Now find a smaller tree okay, that will achieve close to the greedy algorithms best performance again I will have to make a compromiser I cannot say that give me the smallest tree that will give me the same performance as the greedy algorithm.

Because if that is exists one in fact greedy algorithm would have found it right, along the way as it was growing right, and therefore we have to say that okay give me a smaller tree right, that is close in performance to what I get with the greedy algorithm right. See remember we already made an approximation by assuming we are doing recursive partitioning right, so you cannot get the best possible performance okay, that we have given up right by choosing a tree representation right.

So this is a lot of approximation that is why I said right, I mean there is no good understanding of how decision trees eventually work if you ask me two specific questions like okay, how good an approximation will this greedy algorithm converge to right, suppose my performance the best possible performance on this the Bayesian optimal error on this dataset is say 93% performance I can see 7% is optimal error okay, how close to optimal error will a decision tree algorithm get to no answer right.

While you can answer some of those questions for things like logistic regression and consider some splitting variable j so what I mean by splitting variable the splitting variable is the question that I asked here, okay this variable here in the question so x1 right or x2, so in this case of splitting variable is x1 here the splitting variable is x2 okay, and what is the split point it is the number on the other side right so 0.6, so in this case the splitting variable was x1 and the split point was 0.6 okay.

Consider some splitting variable j and a split point s, okay so I am splitting my input data into two parts one where the jth variable is less than s, less than or equal to s the second part where the jth variable is greater than s okay, let us still to get to two parts so what we really want our j and s such that okay, I am seeking j and s is that if I fit the best value for the points that lie in R1and if I fit the best value for the points that lie in R2 that is what the inner minimum minimization is right the sum of this is minimized, so sum of the squared errors over the two regions is minimized right ,so the j and s actually influence which data point goes to R1which data point goes to R2 right.

So once I decide which points go to R1 and R2 I have a fixed optimization problem that I solve right, which one we already solved there. Yeah, I am just talking with the full data set I am at the root right then we can worry about the recursive splitting part right. So make sense for people so far yes, I want to find j and s such that this happens right, how do you solve this minimization problem.

So we can do this. No, this is not classification right, this is actually a regression I am solving right, so all these data points in our one I am going to output one value for all the data points in R2 I am going to output another value so you can think of saying there came grouping all R1 into one I am going to output one value and grouping R2 into one I am going to output one value find the right grouping such that I can output a value such that overall error is minimized okay.

So the first thing know it can be slightly better than that right, or worse I do not know I will tell you when depends a little better or not okay, the first thing you have to note here is I am going to do this for each and every j that I have okay, find the s and then I am going to pick the best j right I'm going to do this in turn for j=1, j=2, j=3 from 1 to P I am going to do this, and then find the best s and that will give me a value for the objective function, right.

And I can use this to compare which j is better that I do not have to do this jointly okay, so that is the first thing you have to notice okay, so given a j right how will you find the right s, so once you have fixed a j you can think of it as just a just a line right, I have to find that s at some point so that I can split everything to one side to R1 everything to other side R2, exactly so what are the steps I should choose for s, so he was talking about recursive doubling that is one way of doing it if you have no other clue right.

And then you have to come back and then you have to search through thee so people know about recursive doubling I start off by looking at 2 then 4 then 8 and I keep doing that at some point some sign will change right so I mean I will be fitting it one way then I will actually my error will start increasing again, so I stop and then now I will have a window of some power of 2 right, so I have to look between 8 and 16 and then I will do a search through that that is one way of doing it.

But there is a slightly better way we can do it any guesses something better than that imagine you are trying to do this not imagine so remember that you are trying to do this from data, I give you a training data set. Exactly, so order the training data along ascending order in that coordinate right, and then just keep hopping on that. Suppose I have data here, here, here and here somewhere there right.

So now if you think about x2, let us say x2 is my, or x1 is my splitting criterion so there are only five different values of x1that actually occur in my training data right, so that is x1 equal to this, x1 equal to this, x1 equal to this, this and this right.

(Refer Slide Time: 24:16)

It does not matter if I consider any other values for x1 because that is one of these five values will give me the same split, right suppose I consider this a splitting point does not matter I could have as well consider that as my splitting point you see that right. So I do not have to consider it

have to go smoothly along x1 I can just use any one of the data points that has come to me already right, so that is the easy way of searching for a splitting point in x1essentially what we will do.

Right, we just start it one of the reasons I already used either less than or equal to here it is not easy, so how much work do we have to do for finding one splitting point nxn logn or because the sorting part is it okay, you do not have to sort here yeah, you do not want to sort here that is what yeah okay, you can get away you can just go through whatever already good yeah, it does not have to sort here. No, if you sort I mean the computation becomes a lot easier but for computing the complexity you do not have to sort, right you can leave it as it is you can it is NP N for each feature and you have P features right, so the amount of work you have to do is NP.

But if you sort then life is a little easier but you do not have to when you are writing the code you will know what I mean, but yeah an NP is the amount of work you have to do for one feature one level right, great so now what we do I have found the optimal j and optimal s have a the optimal and all our assumptions right, because I am actually doing an exhaustive search over j and s right, I am not doing any approximation here I am doing an exhaustive search so given the assumption that we are going to do something greedy and we are going to split on one variable at a time so we are finding the best possible variables great.

Now what do I do, I actually create the two sets R1 and R2 at having found the best possible splitting point and the splitting variable and the splitting value I find the two regions R1 and R2 and then I go into R1 I do the whole thing again, assuming that R1 is my entire data set likewise I go into R2 and do the whole thing again assuming R2 is my entire data set right. Thus, it makes sense to consider the j again the j that you split on say some j* right does it make sense to consider j* again yes, what does not make sense is to consider j* along with the same s.

In fact does not make sense to consider j* along with any s greater or lesser depending on which side you are right, so you can progressively you keep pruning your search based on, it but you do not have to worry about it because it is automatically taken care of because you are only looking at the values that are present in your data point, I am not written those things down right, you want me to write down the whole process all of you remember it right, j* is the one that gives me the minimum in this that I am considering each feature in turn right.

So j* is the feature that I finally choose to split on and s* is the value that I finally choose to split that j* on okay. there is another question somewhere okay, good okay great, so how far do we go so this is a question that we all had in our mind from the beginning right, if I do not put any restrictions on it I will keep going until I have one data point per region right, great yeah, so that is a really other people actually notice it could very well be that the number of features ends right.
How can you, yeah j can be repeated, but if j cannot be repeated can end up with something like this there is only one data point per leaf per region, where that is no more than one data point per region. Now you could do that but answer my question, we are allowing features to be repeated right, can you still end up with a point where you cannot grow the tree anymore but you have more than one data point per region.

If you keep getting your betting point as your border if your region you can traverse any further. Not, they are the same data point we can repeat it I never said your exercise have to be unique right, now it is not so it sounds like a trivial thing but no it is actually important I mean you should think about it right, so this is very important in cases where the yi's are different the xi's are same and the yi's are different there is no way you will get 100% correct maybe if you are assuming that it is a deterministic process truly underlying process is deterministic and it is corrupted by noises, but what if it is a stochastic process truly a stochastic process is generating all of these things for you right. Yeah, sure you can you should there is no question of you allowing it is life happens to you.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in