NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

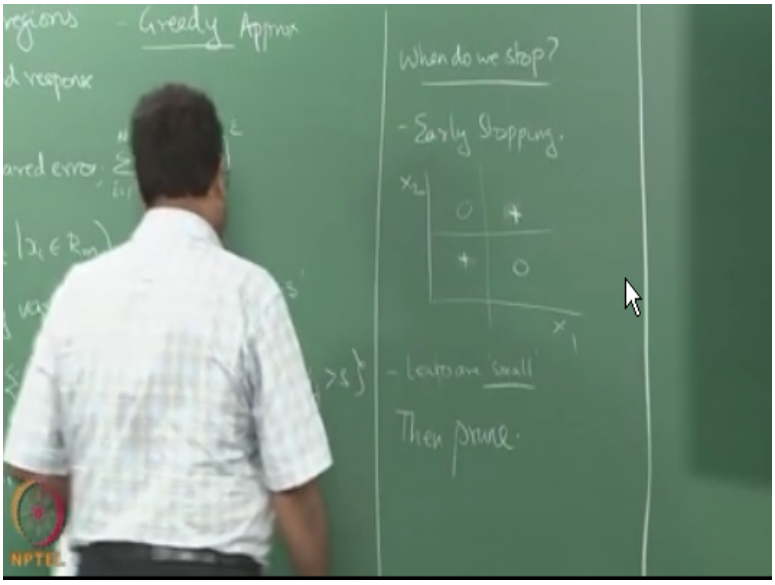
Lecture 41

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

Stopping Criteria and pruning

Right anyway so the question is when do we stop.

(Refer Slide Time: 00:27)



So that is one technique called early stopping okay, where you say that. Hey, I am considering all of these regions right all of the split points but the amount of improvement I get in my error is very small and therefore I stop. I come to some point so I let us say I have like several regions here now so I consider R_1 okay where I consider all possible ways where I can pick an X_1 all possible ways where I can pick gets to and try to split and the error does not change by much.

I can stop I do not have to keep going into smaller and smaller regions even if there are many data points here I can stop right is it a good idea so let us go back to XOR right so this is not a

classification problem we are talking about regression so the excess have an output of + 5 this 0 is output of -5 right and you are trying to fit this. I can try to look at splitting it on x_1 so I have x_1 let me speak I have x_1 and x_2 right so if I were to try to split on x_1 right what will I do right.

And the only thing I can do is this right so if i split anywhere else I will be just keeping a all the data in one side right and all the other mean other one will be empty right this the only meaningful breaking point right and what will be the prediction error here whatever is the half of it i will be predicting the average of this right so it is 0 and 5 then I will be fighting 2.5 so it is essentially 2.5^2x 2 right.

What would have been the prediction error if I had kept the entire region as one. Same thing the average--average sum square will be the same right so if I split on x_1 I am not getting any improvement so let me not split on X_1 okay what about X_2 same we split on X_2 I will not get any improvement let me not split to X_2 that essentially means that I will just give you the average output for all the four data points.

But if I split on X well and then split on X_2 okay now I can do really well right so early stopping is usually a bad idea because we will miss out such these kinds of interaction effects if we stop to early. Good point yeah! So in this case I mean one of them is a trivial split right this case is easy but in general yeah so you will have to take a call yeah! Case like your institutions day careful of them at one time but food is easy see yeah.

So just think about think about this optimization problem right if I m going to split on two at the same time this optimization problem becomes harder that instead of minimizing over J and S I allowed to minimize over j_1, j_2 S_1, S_2 it becomes harder and harder sure you could think of other ways of optimizing it right so but the most common way of doing it this main reason people do not do two variables at a time is the interaction effect I made.

So if I start looking at two variable Saturday then I can start thinking a whole bunch of other things so why do I look at j greater and alone right I could think of other combinations hey J_1/x xj/xk right X JXK I mean so then it just starts exploding so they said okay fine we will just do it this way and make sure that we grow a large tree very large tree so that we actually capture the interaction effects.

In fact you stop when the when the leaves are small I think of a tree right so the leaf of a tree is a region right so the region small I put it in quotes is not the extent of the region it is the number of data points in the region right so--so small would be like two or three or five or something of some really small number depending on how large your data set was you keep growing your tree. So if you view some of the standard tools right.

There will be an inbuilt parameter right which says how small the leaf should be right and you might have to go and prettily with it if you are going to use a decision tree function from either VICAR or MATLAB or something right they have an inbuilt parameter that says how small is the leaf and they stop right so for VICAR it is 2 now you might want to change it to five or something I am not sure what is the limit in MATLAB but you might want to change that.

So that is that is a parameter that you have to fix right and it matters, it actually matters. like he pointed out so if you if you set it too small then you might miss things like this right and then what you do? You build a very, very big tree right this is what I was telling you. You try to use your greedy algorithm and get the best possible tree that you can write so a tree with very very small leaves this kind of the best tree that you can build right right.

Once you get there now you are going to ask the question okay what is the smaller tree that I can get that performs almost as well as the big tree that I have right so there are two ways of doing it the first one she called reduced error pruning case is rather simple so I each leaf then the smallest the largest leaf is smaller than the threshold effect okay so every leaf should be less than that size so basically I mean I can stop each branch independently.

So whenever a leaf reaches size 2 I do not split it anymore so I keep doing but other branches can continue growing so the tree does not have to be of uniform hydrate at some part some sub tree might be shorten some sub tree might be longer right if you remember that picture here so I kept drawing lines in only one region so that means that path alone would have been a much deeper sub tree and others would have been much shallower that is fine.

so reduced pruning is something very simple and so I have a training day training set I built the tree fully on the training set and then I have a validation set may we talked about validation set long time back I have a validation set now what I do is I start greedily or not greedily it is very

safely pruning away my internal nodes right so what I can do when I erase the only tree I had onboard here right.

So what I do is I have this prediction that I am making right I will replace right an internal node with a leaf it does sorry exactly I am just joining the region is together now I see the performance of this with respect to the validation set is I had the original performance on the whole tree right now I look at the performance with respect to the validation set right it could go down right it could go up depending on how the validation set is right.

When it because the tree was constructed only on the training set when you do the pruning the error might actually go up I mean the error might go down sorry right for on the validation set if the error improves our state is the same I will keep this right but if the error mug becomes much worse right I will put it back try otherwise so I as I use the Yi's for making a prediction right and then I keep doing this in turn.

Yeah that could cost could cause more variation agree provided mean usually when you have a large enough validation set so you can actually trust it right and then you try this again right and then if does not work keep going yeah so it is like to have this region but in stuff that I just treat this as one reason now once I you collapse the region I again do average on this whole region and you start at the output right.

See once I collapse this question is each one of this could have been outputting a different value right what will you do with the combined node right so I will take all the data points in the combined node take the average output and we use that as the new output for this right I could take the average of these two but why is that not a good idea the number of data points could be different right so it is it not be truly the average of the outputs right.

So if there are having the same number of data points then I can take the average of these outputs and use it otherwise they should okay so I keep doing this suppose I was able to prune right and now I have pruned this and I have ruined this as well then I can go back and try to prune that also right I can replace this whole thing with this and see how the performance is on the validation set right.

No reduced pruning is only on one thing you see the problem we do cross validation is I will end up with five different trees after the pruning now the question is how do I combine the 5trees

right yeah so exactly so see that is this a very same pruning works it is only one validation say does not use cross-validation right so in that for that reason it is not that popular anymore I am just introducing reduce air pruning because is easy way to think about pruning right.

But like issue is pointing out first of all the variance will be very high depending on what you pick for the validation set right you will end up with a very different tree right so just like when already decision suffer from very high variance and the reduced pruning will actually make the variance worse but this is conceptually easy way of thinking about pruning and if I introduce a more complex pruning method right.

Then a little harder right yeah sorry as long as you are improving sure I will come to that I have a whole class planned on all those model selection methods right since he knew about cross validation he asked me the question I answered but I will come back to that right a whole --whole lecture planned on the model selection okay so cross validation is something guys should never forget.

Once you learn the other kind of pruning which we are all familiar with is called cost complex tree pruning right where you have your error function right and you also have your share in the name also have a cost for the complexity okay like you had here β^2 in your ridge regression and things like that right and norm β. so you already know about this kind of cost complexity measures right so we looked at that in ridge regression we looked at that in lasso and things like that.

And here what we essentially do is we grow the full tree right and then what we do is for every possible non terminal node that you can collapse right you collapse that non terminal node so it should mean that the entire sub tree underneath it you consider as a single region and replace it with the average prediction for the single region like that you can collapse each of them on terminals and create many many different trees right.

So each of this is a sub tree of the original tree right so what do you do is once you created such a collapse tree you look at the average prediction error of the tree it essentially look at the prediction error for each data point divided by the number of data points you get the average prediction error and add a complexity term right the prediction error plus some size of the tree right

So what is the complexity that we are really if so T is at three and α as a parameter will come to that so what is the complexity measure you think is good for a tree number of leaves right number of leaves number of regions you are split into so that is a measure that we use so when I say size of a tree it is the number of regions that the tree splits the input space into so α is a parameter that controls how small a tree I want.

Large α means small trees small α means large trees so now I essentially find my T okay which is a sub tree of the original tree right so it is not any arbitrary T tree okay I have original T tree that I have grew that I grew with this procedure right with this procedure I grow a tree and then I stop when the leaves are small and then what I do is I try and collapse each of the internal nodes of the tree and you can do this in a slightly better fashion right.

You can try to collapse from the lowest level on up and then stop at some point and things like that but you should remember that it could very well be that may be collapsing one sub tree alone might not give you much of an improvement right but if I collapse everything above it right it might give me an improvement why so maybe collapsing this alone does not give me an improvement but collapsing here when might give me an improvement.

No see the point is the error reduction might be small right but then I might not have gotten rid of enough nodes and if I get rid of this whole thing that I get rid of a lot of regions the complexity of my tree comes down significantly so even if I am making a slightly higher prediction error I might be willing to accept that because I have reduced the size by such a significant amount right so that is one of the reasons you consider all possible things.

May be pruning lower down might not simplify the tree enough for you to accept error reduction but if you go higher up the tree you might actually get the same error reduction right I mean whatever by reduction error worsening right but you must have you met a reduced entry by a much larger amount therefore you are willing to accept that right so it is actually no a great idea to just go bottom up all right so --so the small things to remember right.

So that's something that you pick since the magic word has been introduced so you pick α who ask the α question okay since the magic word has been introduced you pick α by cross-validation I will tell you what cross-validation everybody okay so all of you understand what validation is

right so cross-validation essentially is kind of a multiple rounds of validation and instead of just using a single validation set.

You in fact try to use all parts of your data as validation in a very systematic fashion okay we will talk about this more detail later but just to give you a rough idea and so this is clear so what we are doing here yeah no, look at all possible collapsing right so basically what i mean by collapsing remove an internal node the entire sub tree structure underneath it whatever regions it was covering you consider that as a single region and you replace it with that.

So I can choose any internal node to collapse full sub tree and yeah concept is an expensive process so i said if it is expensive you can come up with other mechanisms of ordering it right but the best way to do it sorry nope nope decision trees there is nothing that is optimal I mean right I mean everything is hard right so any questions on any other questions on cost complextive pruning.

Likely yes but we do not know until you actually fit it you wouldn't know for example in the XOR case what we call it over fitting or not so you would not know right until until you fit the data you do not know whether you are or fitting or not so you have to grow the whole tree and if you are over fitting then when you prune you will actually end up removing it I mean the error will obviously on the training data the error will obviously be lower when you over fit right.

And that is why you need the complexity criteria right so when a prune if I am do not lose too much in terms of accuracy then I am happy to so any other question so so essentially what you do with the α as you pick a good choice of α right and then try to do the pruning on five different validation sets then pick another choice of α right on the same five validation sets you pick another different α right on the same five validation sets.

And then pick an α that gives you the best it depends on how you are normalizing the prediction error and as well as what is the expected size of the tree you are going to see right if the prediction error lies between 0 and 1and the tree sizes are order of, order of 10,000 right you would really want your α range to be small right where the tree is also of the order of say5 or 10 nodes then the α is could be larger.

IIT Madras Production

Funded by

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

www.nptel.ac.in