

**NPTEL**

**NPTEL ONLINE CERTIFICATION COURSE**

**Introduction to Machine Learning**

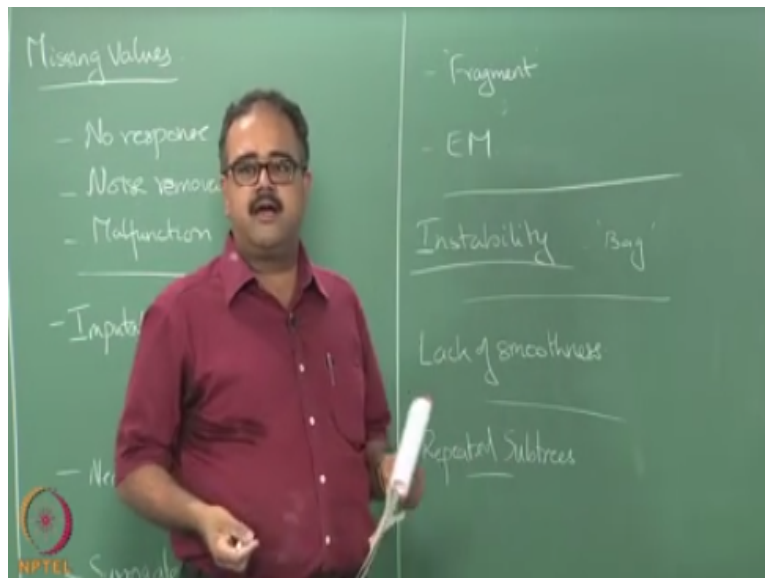
**Lecture 46**

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

**Decisions Trees – Instability,  
Smoothness, Repeated Subtrees**

So the next thing would not talk about is instability.

(Refer Slide Time: 00:19)



So this decision trees are pretty unstable so what we mean by not stable all of you know what we may not, stable small changes in the training data will could cause potentially large changes in the decision tree, so what could happen things that you split at the root might go somewhere down because of some changes in the data right and if the data you start out with small where is the sample size is small then the variation is going to be very high right. So I said any way of getting around it some of it some of some regularization helps to some extent the solve the pruning and stuff helps to some extent.

But still not a lot right because the variance is really high these things is really unstable but still trees are very useful, so what we will do is we look at a very specific technique so that that is what we are going to do one of the ways of doing it yeah so there is a very specific technique called bagging right, so that not the next not the next four classes from now okay I will do bagging in more detail right but the basic idea is that to minimize variance is not just with trees you could do it with any unstable classifier.

So what you do is instead of training it on the data that is given to you train on slightly different versions of the data right maybe you can just take say 70% of the data randomly choose 70% of the data and then train a tree randomly choose another seventy percent train another tree keep doing this and then somehow combine the class labels predicted by all the copies of the trees.

That you have trained so this allows you to have slightly more stable classifier what is the problem with this yes anything else yes anyone else and I think I heard somebody say that you lose the biggest advantage of decision trees, which is simple comprehensibility so instead of having 13 or 15 trees or 100 trees now again you have the problem of keeping your job and your manager ask to explain what happened right so now I have 100 trees and somehow they make this magical prediction and you do not know.

So that is a problem so smoothness in your prediction is something that you are looking for a condition trees are not going to give, you that right there will always be this jagged jumping around especially regression trees right, so when you are talking about making some predictions if you are using a piecewise constant fit right for every region you are going to have some amount of jumping around so if you are looking for a smooth function for doing your prediction this is not going to work so you have to do some kind of post-processing after you build the tree in order to smooth the predictions right.

And there is nothing you can do this nature of the beast so the trees are so much convenient in other ways but smoothness is a problem, so if you are looking for a smooth fit right for your prediction that is not going to happen and problem of having repeated sub trees, so what do I mean by that multi way splits it does not have to be multi waste receiver in binary splits you can get into this problem think of  $x$  all right how will  $x$  or look like right I will split on  $x_1$  if  $x_1$  is 0 I will go down the left branch  $x_1$  is 1 I will go down the right branch and then what do I do in the left branch.

I will test on the test on  $x_2$  if  $x_2$  is 0 I will go down one branch  $x_2$  is one I will go down the other branch and likewise I will test on  $x_2$  on the other side with 0 I will go down one branch I will go is 1 I will go to the otherwise so we can think of it these two sub trees are kind of similar right, so it could very well be that I split on one attribute but everything underneath it could be similar the tree structure is very similar but I cannot collapse it because I end up with different conclusions right.

So if  $x_1$  was 0 and  $x_2$  are 0 I would be outputting 0 right but  $x_2$  is 1 and  $x_1$  was 0 I would be outputting one so the outcomes are different so I cannot really club the two trees but then the test that I do are exactly the same right, so this kind of trees are prone to having this kind of repeated sub trees right so you could have the same test set of tests that are actually implemented in many different points in the tree so it just makes it three more complex.

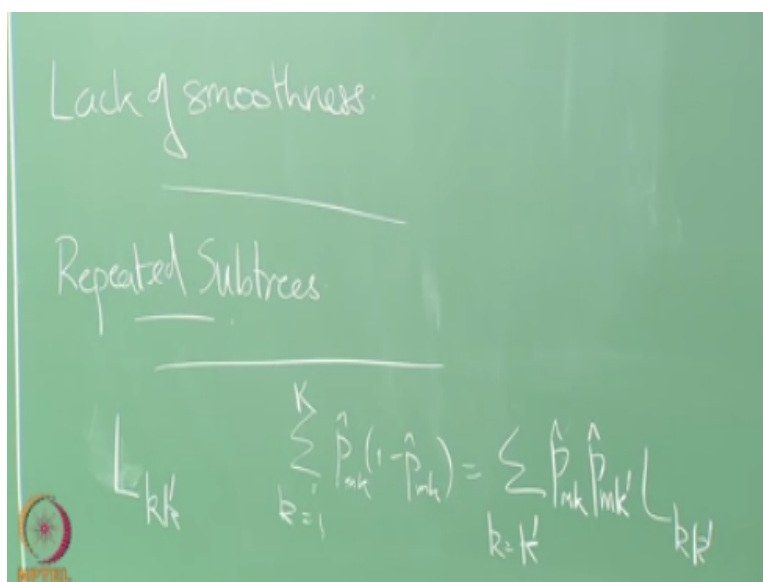
(Refer Slide Time: 05:51)

Lack of smoothness.

---

Repeated Subtrees

---


$$\sum_{k=1}^K \hat{p}_{nk} (1 - \hat{p}_{nk}) = \sum_{k=1}^K \hat{p}_{nk} \hat{p}_{nk} \quad L_{kk}$$


But there might be other ways of reordering things, so that you get with the simply the  $x$  or is a bad case right so if you reorder  $x$  are you still get with this we still end up with the same kind of repeated structure but there might be other cases where you might have just done the splitting in the normal way but end up with too much repeated structures but if you had we flip the ordering of some variables even though it is not the best variable to pick at some point but you might end up with a more compact tree but finding that is finding that is very hard finding that ordering is very hard as I so you have to just live with it just pointing out some of the caveats.

So far we have assumed that we are dealing with the 0 1 loss function so what is the 0 1 loss function for classification right yeah as good as a mile I mean I do not care there is no ordering in my class labels if I miss if I do not predict it correctly I penalize you with one if I predicted correctly it 0 but there might be cases where some miss classifications are more acceptable for you than others right so what do you do in such cases they so you are going to have some kind of a loss value right.

So I am going to have some kind of some LKK` which is essentially the prowl loss that I will suffer by classifying the data into  $k`$  when it is actually class  $k$  correct so I am going to have this so how do I accommodate that in the decision tree setup, how do I account how to accommodate that in the spm setup optimal hyperplanes by the way if I am missing one thing we never actually talked about how you use svms for multiple classes I will ask me the question what is the margin maximum margin.

What is a margin mean and you have multiple classes that is a topic for another day I will come back to that but yeah, so as she is not immediately clear right so how we do that so suppose you have neural networks like we all know about neural networks right, all of you are familiar with back proc by now I suppose right so how will you accommodate this kind of thing in back prop I think of ways of doing it, so it turns out there is no easy way of doing any of these but what you can do is at least in this shin trees.

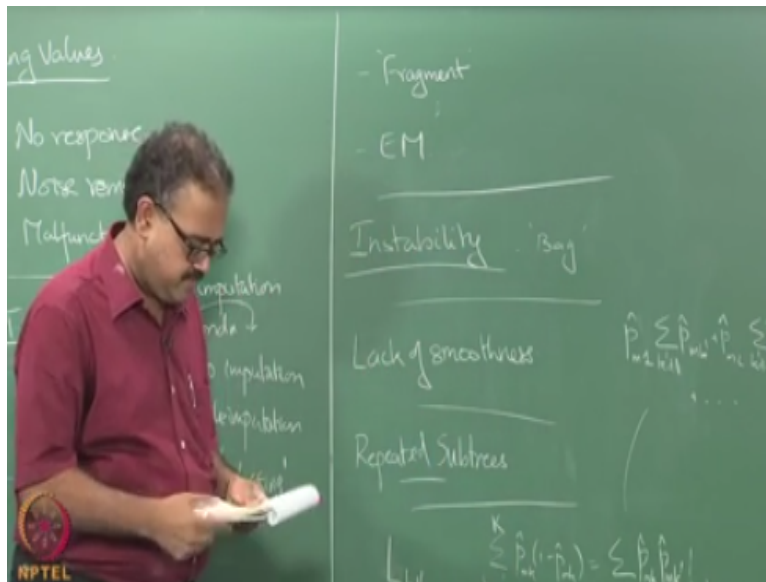
Try to incorporate this when you are computing your guinea index or your what information gain whatever right, so whenever you are looking when you are doing that so you can figure out okay what is the probability that I will miss classify this so what is the guinea index expression that we have right, so this is what we had right so this is essentially this was the probability that a data

point in region  $m$  will be in class  $K$  right times the probability that data point in region  $M$  will not be in class  $K$  right.

So as another way I can write this which is essentially right so probability that the point is  $k + k$  and probability that is  $K$  so essentially from here to here what I need to do is take out all the terms where with  $p$  hat  $m_k$  and some out some of our the remaining so which will be  $1 - p$  - hat  $m_k$  right, so that is essentially what I did so for each  $k$  I take  $P$  hat  $m_k$  out from here and some over the remaining things and I will get this expression okay so this is some way of saying let okay the original probability is  $k$  okay.

And the estimated probability is  $K$  this is the someway of looking at it so here what I can do is I can add my LKK prime, so you have to actually work this out for all of the measures that you are going to work with so if you are going to have a neural network mean squared error criterion you are minimizing or cross the deviance at your cross entropy or minimizing whatever is error function you are minimizing lot to figure out what is the appropriate way to use this class information this class specific loss information okay.

(Refer Slide Time: 11:40)



There was a not equal to the right yeah, okay without this yes they are equal without this year equal so essentially what is what I am doing here is so for every  $k$  right I am writing 11 term like this so he this I can simplify like this right, so that is  $p$  hat  $m_1 \sum k$  not equal to  $1 - p$  hat  $m_k + P$  hat  $m_2 \sum k$  not equal to  $2p$  hat  $m_k$  right like that I can do that so I will get case terms and this

summation is essentially  $1 - \hat{p} m_1$  and this summation is one minute  $\hat{p} m$  to write like that so that is essentially what I get here right so like that you have to work it out for everything so if we have a different loss function okay.

**IIT Madras Production**

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

[www.nptel.ac.in](http://www.nptel.ac.in)

Copyrights Reserved