NPTEL

NPTEL ONLINE CERTIFICATION COURSE

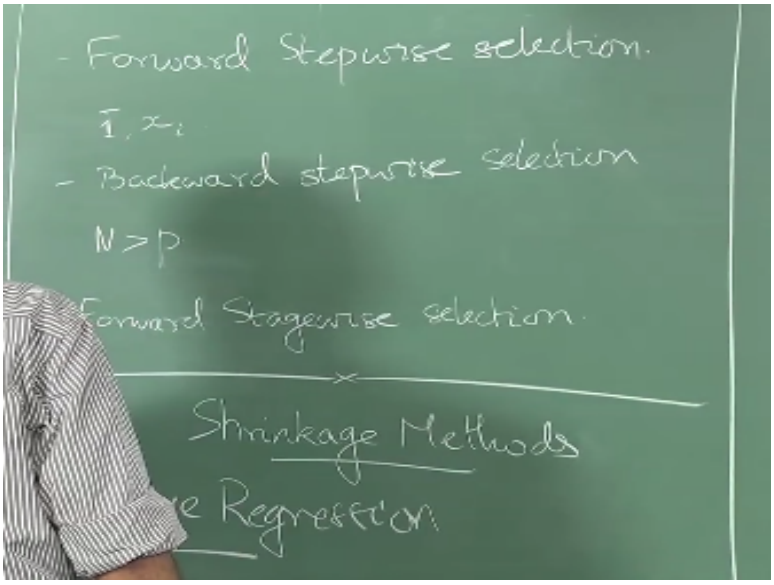
Introduction to Machine Learning

Lecture 15

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

Subset Selection 2

(Refer Slide Time: 00:21)



Right, so it is called forward stage weight selection where at each stage you do the following okay. Let me rephrase it, on the first stage you do the following, so you pick the variable that is most correlated with the output like you pick the variable that is most correlated with the output and then you regress the output on that variable find the residual. Now what you do is pick the variable that is most correlated with the residual okay, regress the residual on that variable okay.

Now add it to your predictor okay, so what is your predictor, you already had one variable right then you had a coefficient for that variable which you got by their first regression. Now you have a second variable and they have a coefficient for that variable which you got my regressing the

residual on this variable they essentially what you are trying to do is okay the first variable make some prediction okay.

The second variable is going to try to predict what the error is right, so essentially now I will be adding the error to the prediction of the first variable. Did that make sense? Right. So the first variable let us say that is the true output that I want right, so the first variable will make a prediction saying that okay this is the actual fitted value right, and this is the residual. What I am trying to do with the second variable is actually to predict this gap right.

So when I add the second variable with this coefficient to the first variable, so what I am essentially doing is okay the first variable will give this as the output, the second variable make some other prediction let us say that much so I will add the two, so the new output will be that right. Now I still have a residual left right, so then I will pick a third variable which is maximally correlated with this residual.

And now I add the output of all the three okay and then I get my new predictor okay, does it make sense? So at every stage I find the residual whatever has not been predicted correctly by the previous case stages right, whatever is the residual error after the previous case stages and try to predict that using the new variable and essentially I find the direction which is most correlated with this prediction and then I try to give you that okay, make sense? Right this is called forward stage wise selection right.

So what is the advantage of stage by selection? Come on I asked a question I believe can you think of any advantage of this sorry, neither was I randomly picking a variable in the previous methods right, I was picking greedily that was not random. No even in the previous case I only pick variables that gave me better fits right. In fact I will tell you that it will probably converge faster in forward step wise selection rather than forward stage wise okay.

But there is another significant advantage here if you just thought about the process of fitting the coefficients at every stage I do a univariate regression right, at every stage I am just regressing the residual on one variable right every stage it is a univariate regression right. In forward stepwise selection so every stage I will add a new variable, but then I have to do a multivariate regression, I have to do the regression all over again, I am not able to reuse the coefficients from the previous step right.

So when I add a new variable I basically now I have k+1 variables and into a new regression with k+1 variables, but in this case what is happening at every step stage I just have to do a linear regression I keep all the work that I have done so far intact. So in fact since we are doing this only one at a time right I am, so I am not even though I might have K variables in the system right.

But the coefficients I have for the K variables might not be the same K coefficients I would have gotten, if it started with this K variables and did a linear regression on it okay. So the coefficients could be different right, if I take those K variables and do linear regression I will get a better fit rather than doing this stage wise fit. But we prefer to the stage wise, because it saves us a lot of computation okay, makes sense.

Eventually everything will catch up and we will get the same kind of prediction at the end of it, but you might end up adding a little bit more variables in this work, in this approach, but that is fine right. Another question okay, so the next class of methods we will look at are called shrinkage methods, but the idea is to shrink some of the parameters to zero, it shrink them towards zero right.

So in the subset selection here essentially if you think about what we are doing all this variables or all the variables that we did not select you have setting the coefficients to zero right. But instead of doing an arbitrary greedy search or stage by selection and so on so forth, in shrinkage methods what we do is we come up with a proper optimization formulation right which allows us to shrink the unnecessary coordinates okay.

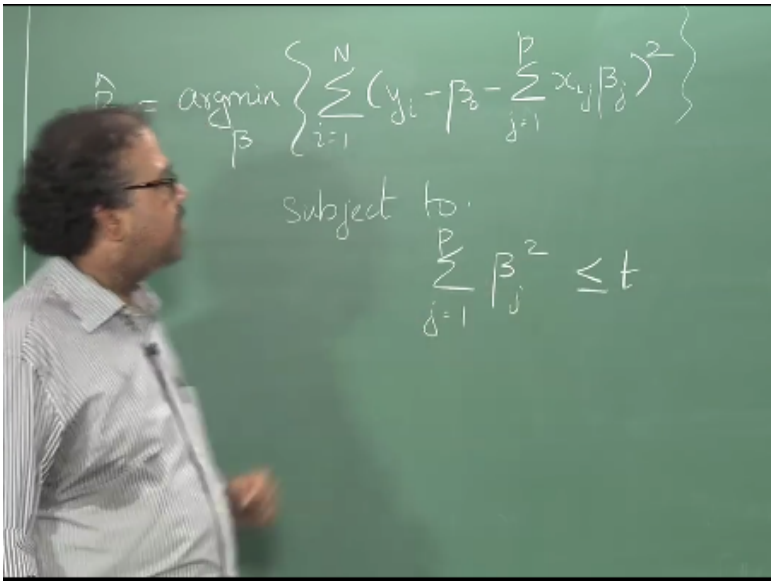
Ideally you would like to shrink them all the way to zero, but there are problems in doing that, but we will try to keep them as small as possible you can do some post-processing and then get rid of really small coordinates and things like that. But we really like to shrink these coordinates right. So this is fine from the prediction accuracy point of view right from the interpretability point of view it still leaves a little bit to be desired, because you might have a lot of coefficients with I mean a lot of variables with very small coefficients back in the system.

So it is still a little bit of a thing, but mathematically this is a much sounder method than any of these things we have been talking about. And of course this is the soundest, but also impossible right. So the first thing we look at it is called ridge regression the whole idea behind mean all of

this shrinkage methods is that you are going to have your usual objective function which is what the sum squared error that you are going to try and minimize the sum squared error.

In addition you are going to impose a penalty on the size of the coefficients right. So you want to reduce the error, but not at the cost of making some coefficient very large right. You do not try and shrink the coefficients as much as possible, so what will happen is your optimization procedure will try to find solutions which have as smaller coefficient as possible and give you the similar kind of minimization in this squared error objective okay.

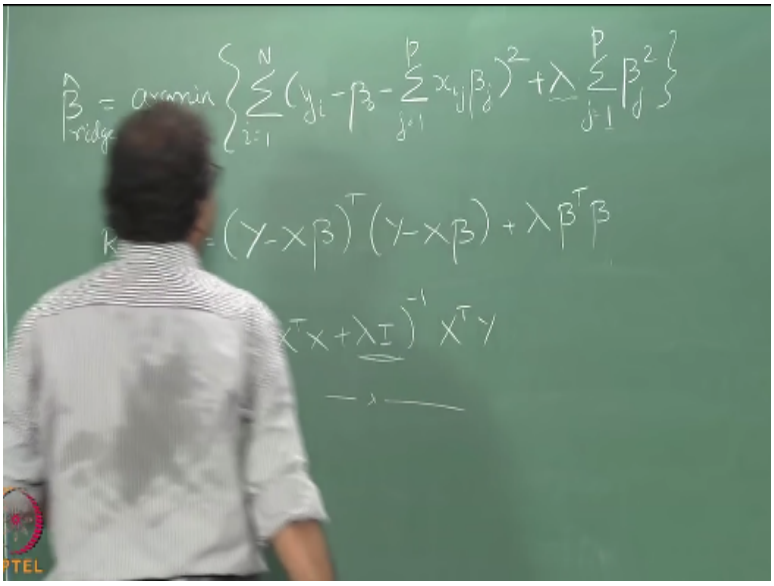
(Refer Slide Time: 08:52)



It is okay I will waste that much of the board I will write things here. So what is your normal objective function right. So that is a normal objective function for finding their β, and so your β had essentially this. So now what I am saying is, let us not do this, but let us do this with the constraint right. So what is a constraint okay fairly straight forward I have added a squared norm constraint right.

So I am making, I am just saying that okay, so this is essentially we think about it is L2 norm for my this thing, so I am taking the root I am just leaving it as a square does not matter right. So it is like an L2 norm constraint for my data.

(Refer Slide Time: 11:16)



So I can make this into an unconstrained problem right, because λ has to be greater than zero why do I want the βs to be small okay good question actually. So what we wanted to do was to make sure that you are reducing the variance of your model right, so that is essentially what we are trying to do now, all the subsets selection was we set the coefficient to zero, we said you have lot fewer parameters to estimate right.

So now what I am doing if I am by imposing the size constraint on the parameters right, the size constraints on the variables I am actually reducing the range over which these variables can actually move around okay. So if you think about it if I have moderately correlated input variables are correlated or anti-correlated input variables, so let us say I have two variables which x1 and x2 which are correlated okay.

Now I can have a large β1 and a large negative β2 okay that essentially will cancel out each other in terms of the predictions I am making, because x1 and x2 are themselves correlated right. So I can actually make my β1 very large and my β2 is largely negative right, so that it will just cancel out the actual effects of the two variables right. So it essentially becomes a difference of β1, β2 that actually matters right, not necessarily the difference in magnitude of β1 β2 that matters not actually not the actual values.

So in which case so I can basically have a large class of models which will give me the same exact output right. So this makes my problem much harder to control and then that increases the difficulty of the estimation problem right. But now we are saying that no, no I cannot allow these

things to become very large, then I am restricting the class of models I am going to be looking at okay.

So that is the reason why the stating size of β helps yeah I did not explain this completely last name so thanks for asking the question right. So we just have to make sure that our λ are positive we know that little so Lagrange multiplies have to be positive and so on so forth. So now I can go ahead and minimize this right. So a couple of things which I want to point out now, so one thing is if you notice the penalty here, so what do you notice about this.

I am not including β0 right see the sum runs from 1 to P it is not running from 0 to P also note that I actually explicitly wrote out β0 here I did not squish it into the P+1 thing, because I am going to be treating β0 specially here mainly, because if I penalize β0 then what will happen is if I move my data up right, so let us say this is my X and Y axis and I have this is the data that I had right.

So now I have to fit that line through this right, it is a univariate regression problem Y is my response and X is my input I have to fit a line right. But now the same data points okay, if I shift them up right, so shifting up the data points is hard, so I will just shift the origin okay. If I shift the origin what will happen if I penalize β0 no, no. So if you penalize β0 it will try to keep this intercept small right, penalizing β0 will try to keep the intercept small.

So earlier when I had that right if you look at the fit it will pass very close to the origin the intercept will be close to 0 right. Now when I shifted this it is going to try and make the intercept small in stuff there is line just shifting the slope of the line will change right. It is the same data it has just shifted up a little bit right, so the slope of the line will change, so it will try to go through somewhere here.

So essentially earlier when the line would I mean like this right, now the line will become like that because I am penalizing β0 right. So we do not want that to happen so just simple shifts in the data should not change the fit right. So we do not penalize β0 right. Does it make sense? And anyway we know what β0 should be do, you know what β0 should be right, it should be the average of the outputs anyway.

So one way which we can actually get rid of β0 from this optimization problem is to say that we will center the inputs right. So we will subtract the average from the Y's and likewise we will

subtract the averages from all the X's okay. So we will center the input, so we will make all the X variables centered on zero right. So we will subtract the mean from all the X's, we will subtract the mean from the Y's okay.

So this will give me a centered input okay, and then I will just do regression on this centered input well there will be no β0 okay. So from now on when I write X it is a nxp matrix where the input has been centered okay. So that way I do not have to worry about the, so essentially what I have done here is I have taken my data from there okay, and shifted it so that the fit whatever is the fit I am going to get will pass through the origin right.

So that is essentially what I have done I have taken the original data translated it, so that whatever fit will pass through the origin okay. And I will go back and add the β0 later to get any original fit does that make sense okay good. So matrix form I write it like this, so you can minimize this take the derivative and set it to zero solve for it you will get this. So here, so both my x and y are centered.

So I subtracted the mean from Y, I subtracted the mean from the columns of x so they are all centered here okay. So just remember that and so once I get this centered values I can solve for it, this gives me the β hat ridge for 1 to P right in the β0 I estimate as Ybar and that gives me the full solution okay, is it fine. So one thing which I forgot to point out earlier you remember I had this variable T here, there was upper bound on the, so I said subject to the constraint that it should not be larger than T, the T has vanished yet, but you can show that this λ and the T are related right.

So it does not matter, so for every choice of T you have a choice of λ okay, but typically what happens is you choose your appropriate lambda and then you work with it, you do not worry about the T formulation okay. Any questions on this, so this tells you why this is called ridge regression, because what they have done here is you essentially added a ridge to your data matrix you take the X^TX okay.

And then you add a diagonal λ which is like adding a ridge of size λ to the diagonal elements of X^TX okay. So that is why it is called ridge regression. So why are you doing this and can you see one advantage of doing this λ I think here, sorry, this whole thing becomes invertible right. So as

well as I add the λi I am sure that this is non-singular. And even if X^TX was originally singular and adding λi makes it nonsingular and it is invertible.

In fact this was the original motivation for ridge regression right, back in the I forget, in the 50s when people came up with ridge regression the original motivation was X^TX could be badly conditioned okay, even if it is non singular we talked about this in the last class right. It could be that some variables are so highly correlated. So even if the matrix is invertible numerically you will get into problems right I told you that the residual might be so small right.

So when you try to fit the coefficients you will get into problems. So numerically the inversion might be a problem right, even if the matrix is non-singular, but by adding this λi term to it you make sure that it is invertible and by controlling the size of the λ you can make sure that numerically also the problem is well behaved right. So that is the idea behind with original motivation for ridge regression was essentially to make the problem first of all solvable right.

But then it then people went back and understood rigid regression in terms of shrinkage or variance direction. And since it makes it convenient for us to talk about a whole class of problems, shrinkage problems right we motivate the motivated rigid regression from the view point of shrinkage as opposed to this inversion problem right. Any questions, so I am going to encourage you to read the discussion that follows rigid regression in the book right.

It requires you to work out some things along with the book you just cannot just sit there and passively read it okay, but then it draws a lot more connections from ridge regression to a variety of other statistical properties about the data which will be useful to know and I am going to make you read, I mean so go read it I mean ask you questions on it later. So go and read the discussion okay. So the next thing.

IIT Madras Production

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

www.nptel.ac.in