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

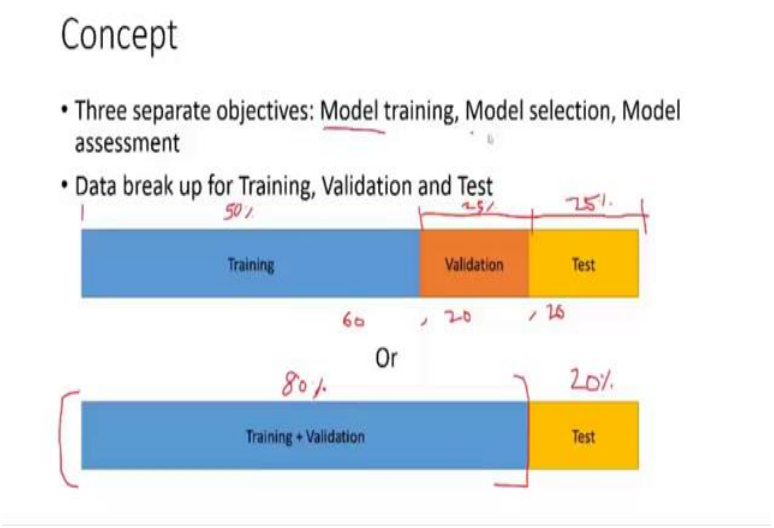
Module – 05
Lecture - 28
Model Assessment and Selection

Hello and welcome to our lecture today on Model Assessment and Selection. So, this lecture is in many ways similar both in style and content to our lecture on bias variance dichotomy. It is similar in style and that, we are not going to be teaching you a new techniques, but we are teaching you a concept in machine learning that is really important. And so like the bias variance dichotomy, it is really applicable to almost every technique that you could be using in machine learning and should not just say every technique in machine learning more narrowly in every technique in supervised learning.

So, your regression, your neural networks, your cards, your decision trees, your SVM's you know, so whatever techniques you are learning right now and whatever you have learnt this is a very, very important concept. It is also very similar in content, because in the bias variance dichotomy lecture you really learnt about how it is really important to understand that you can have highly simplistic models, which would have high degree of bias and low variance and therefore, not really good.

And at the same time you can have highly complex models, which tend to over fit the data and therefore, have a lot of variance, but very low bias. And therefore, you are being introduces the problem there, saying that well you it is not a good idea to go to either extreme. In today's lecture we will be answering coming up with solutions for that problem and not just that problem, for many other related problems, but somewhere the solution is in looking more carefully towards how do you access a particular model and therefore, how do you go about selecting between multiple models.

(Refer Slide Time: 02:02)



So, jumping into the subject in terms of when you have data and you are trying to relate that, use the data towards models, there is one part which is actually using the data to train the model. There is another part, which is you could use some of the data to not just train the model, but to choose between models or to fine tune the model.

So, using the data to train the model if model training, model selection the word selection is use kind of loosely here could mean selecting between multiple models or it could mean, you know it is a meaning you could use that select between completely different models or at the same time you could use that you can fix the model and you could just be like a parameters that define the model and you could go about figuring out what value to set it to by using data in the model selection. And finally, you have the concept of model assessment, which is once you fixed everything out, if you want to get an idea of how good this model is.

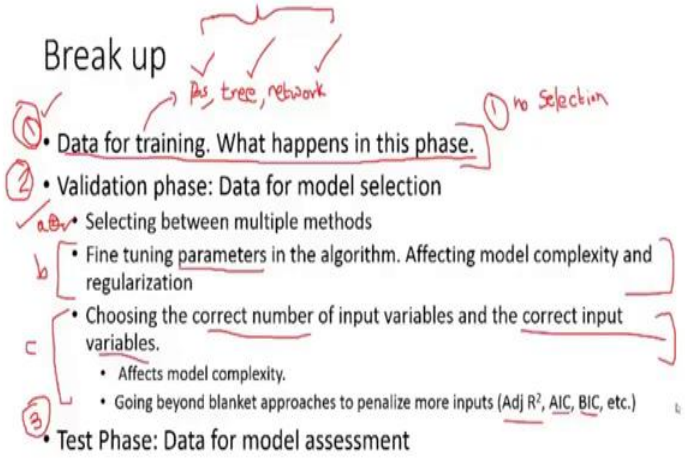
So, that you know that when you take this model to the field, this is the performance you would expect that has to do with model assessment and corresponding to these three goals or objectives, you could break up your data and this is something that you might not always have the luxury to do. But, you can if you have a data rich situation we have enough data, then you could potentially break up the data and training, validation and test, where the training goes towards model training, the validation goes towards model selection and fine tuning and the test goes towards model assessment.

And there is no formula exactly as to what this break up should be or what is enough data, that is a hard question to answer, it varies from case to case, but at the surface you know typically if you have enough data you tend to break it up as either 50 %, 25 % and finally, 25 %. But, you could also have you know 60, 20, 20 and again, even with this you cannot say one is right, one is wrong or something is perfectly right or wrong I am just giving you some values that are typically seen.

You can also see in the picture that I have, the representation I have below that sometimes you might want to do training and validation together and we will talk about that more specifically. But, if you, this sometimes comes up when you do not have as much data, where you do not have the luxury of just taking one 25 percent and calling it validation. So, somewhere you know you just use the same data to perform training and validation and how you will do that is something that we are going to talk about in good detail in this lecture.

But, you still try to keep some percentage for model assessment, then the final say on how good the model will be if you take it out to the field, if the model want to completely look at new data and try to predict how good will it work. So, what we are going to do next is, we are going to take each of these phases and talk in detail about what really we can do for each of these objectives.

(Refer Slide Time: 05:52)



The first phase, which is model training or is one, where you take the data and you train. What we mean by training out here is really that you fixed everything, you fixed all your steps, you chosen a particular model and you chosen all the parameters associated with this model. So, at this phase you are not doing any model selection, in phase one we are not doing any model selection, no selection, you also not trying to decide on what values that some of the modeling parameter should be, what I mean by that is let us see you are doing a ridge regression, in ridge regression you have a complexity parameter.

If you go back to our lecture on ridge regression, you are doing this optimization where it is look like the ordinary least square. But, you have another term, this is the term associated with regularization, where you penalize really large coefficients. So, for that you need to set a value, you need to set a particular λ, you already decided that, in this phase when you are going for data for training, you fix the model and you fixed all the parameters associated with the model.

If you for instance doing neural network training, you are not using this data to decide how many intermediate nodes should be there, that is not what is happening. Once you finalized on the exact model and its details, you are just using this data to figure out what the model should be, meaning take the case of linier regression, you fixed everything, you fixed what the input variables are, you fixed what data you are going to process with respect to a simple linear regression or a multiple regression, there are no other parameters to fine tune.

So, it is you just plug in the data and you get the βs, so here you might just be getting the βs. So, that you have created the model, you are essentially creating the model in this phase, if you are doing decision trees then let say you need to have fixed on the algorithm for the decision trees need to have fixed, what the input variables are, need to have fixed other parameters that you could probably play around within decision trees and we are going to talk about those parameter soon. But, once everything is fixed, you just plug in the data and how to get or is the tree structure or network, in the case of neural network.

So, this is essentially the last leg, once you fixed everything you are not doing any model assessment, you are not doing any model selection, you already selected your model, you already set the parameters for your model. You just plugging the data into this finalized

version and getting the actual model out of it. So, that is what you are doing in this data for training. What happens in the validation phase? In the validation phase, you are essentially using this data. So, this is separate data, you use some data for training.

Now, think about this, you using separate data and you using this data to make some decisions. You could be using this data to select between multiple methods. So, you might say, hey I have this data I could do a linear regression on it, I could do a regression tree on it or a random forest or something. I could do a neural network, how do I know which to do, why not to do all of them and see, which one does better and how do you do, see which one does better.

In the validation phase what you will do is, you will use the same training data that is you will use the same training data that you saw, we discussed in once and you will fit these three completely different approaches. You will create the data's for the regression, you will create a tree for the regression tree, you will create a network for the neural network. Now, you go apply these three to the validation data, what I mean by that is that you go take the input data in the validation and apply these three methods and you get some predicted outputs.

Each of these methods are going to give you some completely different, I mean might not be completely different, but they are going to give you a different outputs, different predictions. Compare these predictions to the actual output that you have in the validation data, any. When we say data whether it is for training, validation or test, it means you would have input output pairs. So, if you have five input variables you will have values of each of those five input variables, one data point is nothing but, the vector of all values of the five input variables and the one output variables, so whatever value that is.

So, in some sense when you have these input, output pairs what you can do in the validation phase is, you would have created the models from the training phase and what you are interested is in comparing multiple different models. So, you might have completely different methods and then you apply the input of the validation phase to get predictions from these methods and compare these predictions to the actual output and see how well they did.

And you can do that by both in a regression context or in a classification context, meaning that if you predicted 13 and it is 13.5 I can say are you miss by 0.5, if you predicted that is going to be a class like I am predicting that this is 70 % chance it is male and then it winds up being female you can say your you can use different kind of a metric to measure that performance like misclassification index or entropy or something.

But, essentially the idea in the validation phase is to choose is make some decisions, we spoke about how you can use that to choose between methods, but more often what we see is that it is huge extensively to find tune parameters for a given algorithm. So, let say I go about and I am decided right at the stage. So, we have already talked about let us call this a, b and c, so we already talk about two a into b I am really talking about here is let say I have fixed I have decided that I am going to use a decision tree somebody use the cart algorithm.

Now, the cart algorithm might have a lot of parameters that I need to set one potential parameter could be then minimum number of leaves in the terminal nodes what; that means, is the minimum number of data points in a terminal node of the tree. So, in a cart algorithm you can keep on splitting the tree into branches and branches still the terminal nodes just have a one data point, the idea is where do you stop and one way of choosing where to stop just to have a limit you put in the limit saying do not create new branches, if they are going to result in the final nodes having less than let us say 5000 data point or we calls them 5000 leaves.

But, what should that be, should that be 5000, should that be 500, should that be 50,000 what we can do is you can run all of them, you can create three completely different trees, one with each version and again validate that use the validation data set to see which does better and for pretty much any method, any complex enough method you will find that the method itself will have some parameters that you need to finalize. So, in the case I guess you guys have finished you might not have started neural networks here, but you must have finish. So, we finish ridge regression.

So, again let us think about ridge regression, in ridge regression we know that what ridge regression it does this process of regularization by introducing a new parameter called λ, λ is what forces is kind λ is kind of like the way in which penalty is applied towards that optimization problem and constraints the size of the βs. So, if we have a very high λ that

is like a very harsh penalty towards large sized data's or coefficient that linear equation. So, a ridge regression goes ahead and applies penalty.

Now, if that λ which is parameter is very low it is close to 0 then there is no penalty, the ridge regression becomes like the ordinary least square regression, it is actually identical if that λ is exactly set to zero. But, then the question comes about what should I said λ to. I understand the concept that I might want to apply some amount of penalty to really large βs and that was what we discuss and the course on regularization saying you know I do not if I have multicollinearity in my data I might have two data's that are off to opposing magnitudes that are becoming really large and so on and so forth.

But, what should I said the λ to be in the ridge regression, in the process of ridge regression I need to said the parameter λ to some value on what do I said it too is there right answer to it. The answer is you might not know that apriori what to said λ to, but you can use this validation phase, you can said λ to 0.25 set λ to 0.5 set λ to 0.75 and create three different ridge regression equations, three different predictors.

Now, take the input data from the validation set enough and apply that and all three predictors, all these three predictors to come up with some predictions, three different sets of predictions of the validation data. Compare the predictions of each of these predicators, compare these three predicts sets of predictions to the actual output data of the validation data set and you might and you can then compare these three and say oh it is look like when I said penalty parameter λ in ridge regression to 0.5 I do better than when I said that parameter to .25.

So, again just you can go to the bank I know you finished a sub set of techniques and you are still going to learned for instance neural networks and some other methods. But, what you should take you know commit memory and some sense is that almost any machine learning technique that you adopt you will have to choose some parameters and this validation can help you choose the parameters. Another example I can think of is for instance, the neighbors what should be a value of K be?

Am I suppose be a let me 5 nearest neighbors, 6 nearest neighbors, 10 nearest neighbors, 1 nearest neighbor what should in K nearest neighbor algorithm that we discuss in our class, where we compare regression to K nearest neighbors. What should the value of K be? and you can use the validation to figure that out. The last use case and sense of the

validation phase is one that can also be used to choose the correct number of input variables and to also say which ones those input variables should be such choosing the correct number and also choosing those which...

So, I can say I need five input variables are not six, but which five they should be. So, in that sense and this kind of really links up with best some sets ridge regression where you are trying a whole combination of different inputs. Again the idea could be that I can have a fixed model, I could say I am going to use ridge regression or like I am going to use regression or I am going to use trees with this parameter, but which inputs should I will be using, should I use all my input variables, should I use input variable 4, 7 and 9 you know. So, that decision also sometimes like we spoke earlier about regression here R^2 square will just keep on getting higher and higher as you keep on adding more inputs that does not mean you getting a better model.

So, choosing those inputs says you can choose a model with inputs a, b, c, d another one where you use inputs d, e, f, g and you can really compare the performance of these two different models on the validation data set. Now, there are some techniques already there that we have discussed which helps you choose what the input should be and we spoken about that using metrics that are more complex and R^2 like for instance I adjusted R^2 which you know goes ahead and you know penalizes more complexity and there are a whole bunch of a blanket of approaches to penalize more important.

So, we adjusted R^2 there something called the AIC information criteria AIC and then this Bayes Information Criteria (BIC). So, there are whole bunch of metrics that will just go way and say blanket I am going to penalize you for adding more input variables. But, we don't need to do that by that might be computationally convenient and it is you could if you have the luxury of creating this validation data set.

A simple thing that you could do is you can just try these different combinations of inputs for the chosen method of doing the training and so on and apply them and see which approach does better. So, you could just untimely wind up using prediction error instead of say something like a adjusted R^2, you can use the prediction error. Because, you are not just blindly looking at how good the model fix the data that you train the model on you now taking the data that the model has never seen when I trained and you

using that to decide which what your input variables should be or how many input variables you should have.

So, the important thing is the validation itself and this whole process of model selection is not just one thing, you could use validation really it is choose between multiple methods, you can use at to find tune the parameters of the algorithm and you can also kind use that and some way to choose what your variables should be. Finally, you have the test phase, now we are at number 3, the test phase and we have three and now we are talking about three the test phase.

The test phase, the purpose is different it is not to find tune the model, it is not to help you come up with the best model, you could come up with the good model, you could come up with the horrible model, the test phases I do not care, my job the job of the test phase is fairly simple. The job of the test phase is to give an accurate idea of what the performance of this algorithm is going to be and you can turn around and say why do not I just use you know something like my prediction error that I got in the validation phase.

And the answer is, you cannot because you have or hell I mean you can go even one step for instance say why cannot I just look at the predicted the error, the residue will some of the squares and some sense. The error that is comes from the training, you cannot do that for the simple reason that you are explicitly find tuning the model, find tuning the co-efficient, find tuning the parameters irrespective and this I am loosely using these term, because it applies to any approach you take, but there is decision trees whether it is discriminant analysis, whether it is support vector machines, whether it is neural network, whatever approach you take, there is a whole bunch of optimization going on these approaches you have to sitting there and trying to make this model make sense out of this data that process itself is going to make you do very good on this data.

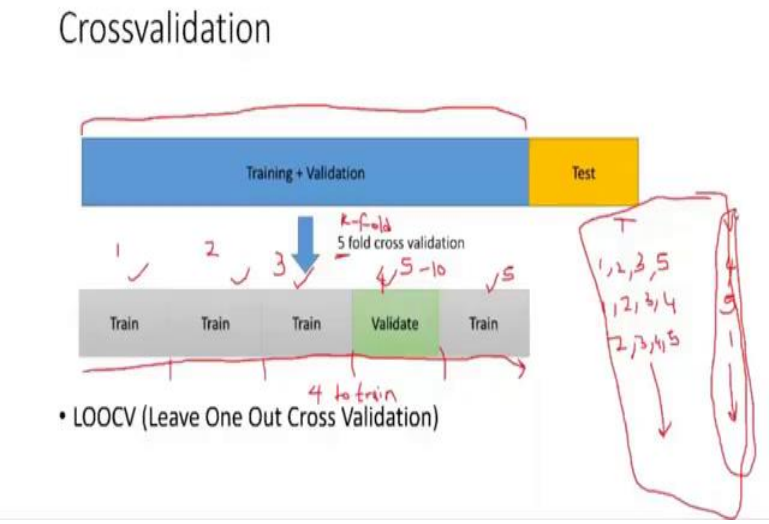
Now, to get true picture of how well you do when you see a new data, you do not want data that was used to make this method good in the first place, you use some data and the training data to figure out the co-efficiency. So, use some data and the training data to construct the actual tree or that actual network, you use some data and the validation data said to find tune what your parameter should be why. So, that you do a good job of fitting.

Now, you do not want to use the same data to tell me how good I will perform when I go when tomorrow someone comes to me and says here is input data can you come up with the prediction for me, to do that you want to really look at data that is never been looked at essentially this is you should think of this as data that is been kept in the vault and never looked at and you brining it out in the end after all the modeling has been finished, you use the training data to construct the tree, you use the select the validation data to fine tune it, you now have end product the end product regression could be a set of co efficient like the βs, then product in tree could be the tree structure, whatever it is.

You basically have a finalized end product and you want to now see how good the end product works, now pull out this data, pull out this data provide the input data from the test data set ask the model to make predictions compare it to the actual output and you are just going to report that. You are not going to use this data to make any more decisions in terms of which method to use or you are not going to use data to fine tune anything about your model.

Because, if you are doing that you are using the data to make the model better, but you are not giving any longer an accurate impression of how successful you would be if you were to completely see this data fresh and you were to make predictions, you are not giving an assessment of the model. So, in order to give the assessment of the model pull out this data that is never been seeing by model in the end and the only purpose of this data is to see how good the model is not to construct the model or not to fine tune the model. So, that is the test phase.

(Refer Slide Time: 25:13)



Now, we are going to talk about in the last part and approach called cross validation and it is really ties to this case that I showed you in the first slide, where training and validation are kind of put together, this is an approach that really kicks and when if you feel like do not have enough that much data for separate validation set and you want to kind a squeeze more out of it. So, a simple approach there is just have the training and validation set as a one large data set and you can break it up into n chunks.

So, out here of broken it up into five chunks going between five to ten is typical and the approach is called actually k fold cross validations. So, it is k fold where you have to choose a particular k and here I have chosen five and so basically broken the data set into five chunks; obviously, very important guys is that this break up is it is random. So, if you given the data in some order, you do not want to very you know conveniently just break it up into five chunks and because they might be some kind of you know implicit order may be the data was for instance chronologically order, you do not want to create training data set one of the early chronology data.

So, you kind of want to shuffle the data up and then break it up into five bins and the idea is to use any four bins to train four to train and one to validate. So, you might sit there and say hay you know you just sounds like, you are breaking up the training and validation at the latest stage than at a earlier stage. But, the answer is no there is one more step, once you do that shuffle them around meaning, now in this particular graph I

have shown you let us call them 1, 2, 3, 4 and 5 in this graph you are using 1, 2, 3 and five to train and you are validating. So, this is train and you are validating with 4.

Now, permute and the next step do 1, 2, 3, 4 and validate on 5 next step permute 2, 3, 4, 5 and validate on 1 and so you like that you keep going till you done all the five combinations, where you would a validate on step and you take the cumulative validation results. So, you will take the cumulative validation results to see how well to make decisions in terms of validation and so this is called k fold cross validation.

So, one question that comes up fairly frequently is what should the value of k be, in k fold cross validation, given you tentative idea there it is typically some are between five and ten, but again that is not something that is cast in stone and it is more important that you understand what it is means to choose high value of k verses low value of k. In the simplest sense our high value of k in a k fold cross validation leads to a model, where leads to an assessment I should say which is of very low bias, but of high variance and a very low k. So, k of like 2 or 3 or so.

So, essentially could have low variance, but a higher bias and one extension of for instance the cross validation, where you have a very high k, where you like it to have an unbiased assessment, but assessment with lots of variance is the case of the leave one out cross validation. The idea here is that if you have n data points I am going to use n - 1 data points to do the training and I am going to validate on that one data point that I left out and what I will do is I will keep just like in k fold cross validation, how you change the training sets in the validation sets, how you shuffle them, how you permute them, the same way here going to leave one data point out for validation train on the others and then predict on this and then keep doing that interactively.

So; obviously, with that if it is essentially like k is equal to n then the cross validation is almost completely unbiased, but can have high variance, because the n training sets are so similar to one and other and some approach like that is also computationally fairly cumbersome. But, if you have that approach then leave one out the cross validation is also something that you might consider.

So, I hope that gives you an idea of cross validation and more broadly the use of you know the idea behind validation as a means of model selection and this whole thing of breaking up training for creating models validation for assessing and kind of you know

selecting models and test for pure assessment, but no for the selection and decision making.

Thank you.