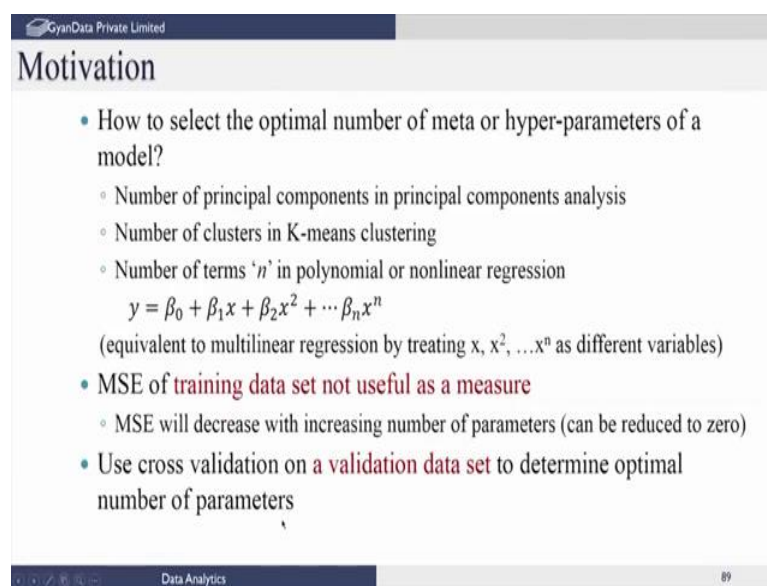**Python for Data Science**
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 31
Cross Validation

Welcome everybody to this last lecture on regression. In this lecture I am going to introduce a concept called Cross Validation which is a very useful thing in model building.

(Refer Slide Time: 00:27)

The main purpose of cross validation is to select what we call the number of meta parameters or hyper parameters of a model. Although we have not introduced principal component analysis in this series of lectures, nevertheless when you learn it in a higher advanced course on data analytics, you will come across this idea of principal components and one of the problems in principal component analysis is to select the number of principal components that are relevant.

We call this hyper parameter or a meta parameter of the model. Similarly later on in this course you will come across clustering in particular you will come across K means clustering and here again you have to choose the number of clusters required to group the data and the number of clusters is called the meta parameter of the clustering modeling. Similarly, if your building a non-linear regression model for example, let us take a

polynomial regression model, where the dependent variable y is written as a polynomial function of the independent variable x.

Let us assume there is only one variable x, you can write the regression model non-linear regression model as $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n$. where x^2, x^3 and so on higher powers of x. This is known as a polynomial model notice this the polynomial model can also be the parameters of this model can be obtained using multi linear regression.

If you take x as a variable, x^2 as a different variable and x^n as a different variable computed from data that you are given, treat them as different variables then you can use multi linear regression methods in order to estimate the parameters beta naught beta 1 and so on with the beta n. Here again we have to decide how many powers of x we have to choose.

So, if you choose higher powers of x, then corresponding to each power you got a extra parameter that you need to estimate. For example, in this case you have n + 1 parameters if you have chosen, x power n as your highest degree of the polynomial. The choice of the degree of the polynomial to fit is again the meta parameter of the non-linear regression model. So, in all of this case is you have to find out the optimal number of meta parameter, optimal number of parameters of the model that you need to use in order to obtain the best model.

Now, you might think that it is easy to actually obtain this by looking at the mean squared error of the training data. So, for example, if you take this polynomial model fit it and compute the mean squared error in the training data, that is nothing, but the difference between y and y hat after you have found out the best fit parameters of beta naught beta 1 up to beta n, you can predict for every data by substituting the value of x x squared and so on and you predict y hat and you compute the difference y - y hat and some over squared of all this is called the mean squared error ok.

So, the mean squared error of the training data you might think is useful as a measure for finding the optimal number of parameters, but that is not true. Because as you increase the number of parameters if you keep adding higher order terms, you will find that the mean squared error decreases. So, ultimately you can get the mean squared error to decrease to 0 as you increase the number of parameters, this is called over fitting.

So, you can always get the mean squared error of the training data to 0 by choosing sufficient number of parameters in the model. Therefore, you cannot use the training data set in order to find out the best number of optimal number of parameters to use in the model. So, we do something call cross validation, this has to be done on a different data set that is not used in the training.

We call this the validation data set and we use the validation data set in order to decide the optimal number of parameters of meta parameters of this model. So, we will use this polynomial regression model as an example throughout in order to illustrate this idea of cross validation.

(Refer Slide Time: 04:41)
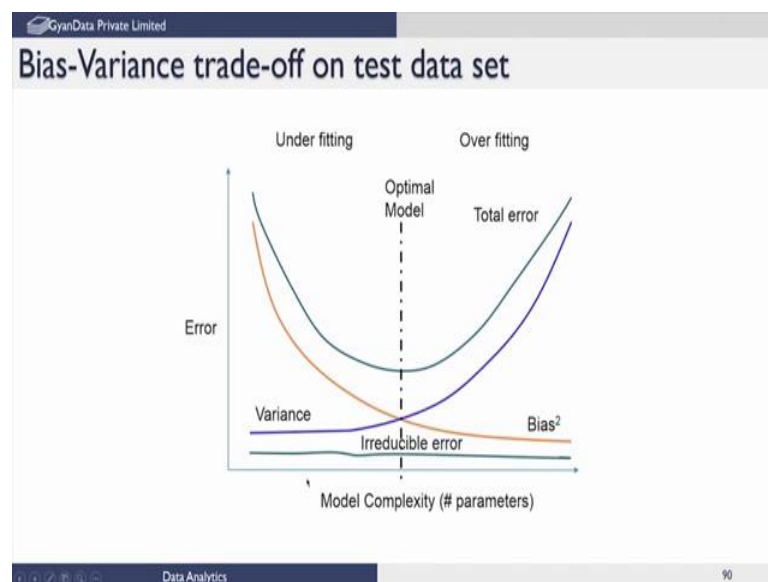
So, schematically what happens when you actually increase the model complexity of the number of meta parameters of the model. So, you will find that the mean squared error on the test set continuously decreases ok. So, we will goes to a 0 as we said on the training set. However, on the validation set what will happen is, if you look at the mean squared error on the validation set that will initially decrease as you increase the number of parameters, but beyond the certain point, the mean squared error on the validation set will start increasing.

So, the optimal number of parameters you should choose of the model complexity is you choose corresponds to the minimum value of the mean squared error on the validation set and this is called the optimal model. If you choose less number of parameters than the

optimal model, we call this under fitting on the other hand if you use more parameters in your model, then the optimal model value is called over fitting.

So, over fitting basically means you are using unnecessarily more parameters than necessary to explain the data on the other hand if you use less parameters, you actually are not sufficiently your model is not going to be that accurate. Typically there are 2 measures for determine the quality of the model. One is called the bias in your prediction error and if you know if you increase the number of parameters of the model this bias squared of the bias term will start decreasing; however, the variability in your model predictions that will start increasing as you increase the model complexity of number of parameters.

So, it is basically the tradeoff between these two that gives rise to this minimum value of the MSE on the validation set that is what you are looking for. So, you want this optimal tradeoff between the bias which keeps reducing as you increase the number of parameters and the variance which keeps increasing as the number of parameters of the model increases. So, this is what we are going to find out by cross validation.

(Refer Slide Time: 06:45)

So, if you have a large data set then you can always divide this data set into 2 parts, 1 use for training typically 70 percent of the data samples you will use for training and the remaining you will set a part for the validation. So, let us call the samples that you used for training as (x_1, y_1), (x_2, y_2) and so on where x represents the independent variable y the

dependent variable and the validation set we will denote it by the symbols $x_{0,i}$ and $y_{0,i}$ where there are n_t observations in the validation set.

So, typically as I said if you have a large number of samples, you can set up are the 70 percent of the samples for training and the remaining 30 percent you can use for validation. Now you can always of course, define the mean squared error in the training set after building the model.

So, this is nothing, but the difference between the measured observed value of the dependent variable - the predicted value after you estimated the parameters let say using least squares regression. So, this is the prediction error on the training data, square overall the samples and taken the average. That is the mean square error that we have seen before.

You can do a similar thing for the validation set also, you can take the difference between the measured value or observed value in the validation sample set - the predicted value for the validation samples and again you can take the sum squared difference between the observation - the predicted value for the validation set, squared over all samples and one the averaged average value. So, that is called the mean squared error on the test or validation.

So, this particular term as I said the MSE on training is not useful for the purpose of deciding on the optimal number of parameters of the model; however, this test MSE test or the mean squared error or the validation data set is the one that we are going to used for finding the optimal number of parameters of the model.

(Refer Slide Time: 08:49)

So, as I said if you have large number of amount of samples then you can actually divide it into a training set a validation set for finding the optimal number of parameters of the model, but and finally, if you want to assess how good your optimal model is you can run it on a test set.

So, typically you take the data set and you divide into three parts; one the training set the validation set where you are trying to use for finding the optimal number of parameters and finally, the test set for to see whether them optimal model you have built is good enough. We will not worry about the test set in this particular lecture we will only worry about this validation set.

Unfortunately, if you do not have large number of samples. So, you would actually generate validation set from the training set itself and we will see how to do this using what is called k- fold cross validation and bootstrapping and so on. So, this is what we will do if you do not have enough data.

We will first look at the case of where you have sufficient number of samples. So, essentially you have n samples and you can divide into a training set consisting of n t samples and the remaining samples you actually used for validation this is a hold out sample as we call it. So, you build a model using only the training set and after your build the model you test it, find the mean squared error on the validation set do this for every choice of the parameter.

So, if you have for example, a polynomial model, you will first see whether a linear model is good, then you will check as quadratic model and a cubic model and so, on you keep increasing the degree of the polynomial and for each case build the model using the training set and see how the MSE of that particular model is on the validation set, and plot this MSE on the validation set as a function of the degree of the polynomial. So, this is what we are going to do for one of the examples and then see how we pick the optimal number of parameters.

(Refer Slide Time: 10:49)
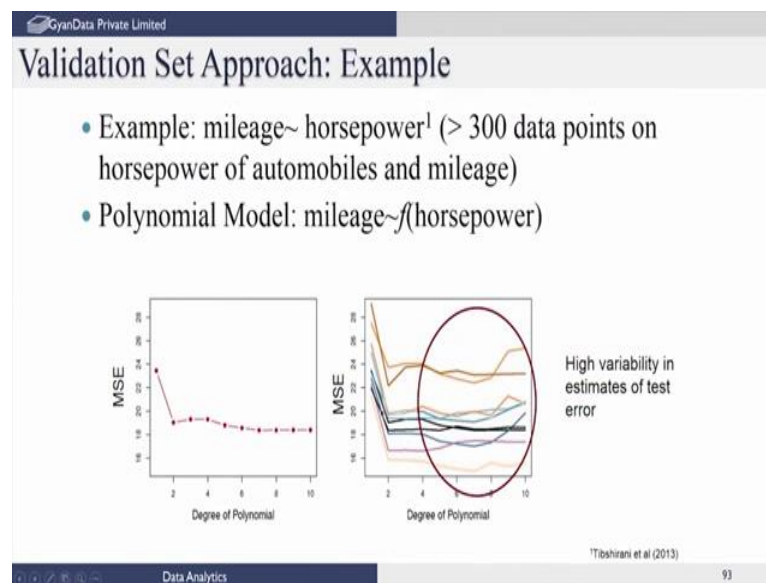
So, here is the case example of a mileage of some automobiles and the horse power of the engine. So, essentially this particular data set contains 300 data points, actually this is sufficiently large, but we are going to assume this is not large enough and we use the validation and cross validation approach on this data set. So, we have 300 data points or more of automobiles different types of automobiles, for which the horse power and the mileage is given.

We are going to fit a polynomial or a non-linear model between mileage and horse power we of course, we can also dry a linear model, but a polynomial model means you can also try quadratic and cubic models and so on so forth. So, this what we are going to illustrate. So, as we increase the degree of the polynomial, here what we have shown is the mean squared error on the validation set. So, suppose we take 70 percent of this is for training

and the remaining 30 percent 100 data points are so, for validation and we look at the mean squared error on the validation set for different choices of the polynomial order.

In this case the first polynomial degree is one implies we are taking a linear model and for two implies were to fitting a quadratic model, for three implies a cubic model and a quadratic model and so on so forth and we are tried polynomial up to degree 10 and we are shown how the mean squared error on the validation data set is as you increase the polynomial order.

You can see very clearly that the value reaches more or less a minimum at to and after which it does not significantly changed. Typically this should actually start increasing, but in most experimental data sets you will find that the mean squared error on the validation set flat ins out and does not significantly decrease beyond the point.

So, you can choose the optimal order degree of the polynomial to fit in this particular case as to, that is a quadratic model fixed this data very well that is what you actually conclude from this particular cross validation mechanism. Of course, on the right hand side we have shown plots for different choices of the training set. For example, if you choose 200 data points out of this randomly and perform regression polynomial regression for different polynomial degree and plot the MSE you will get let us say one curve in this case let us say the yellow curve you get here.

Similarly, if you take another random set and do it you will get another curve. So, these different curves correspond to different random samples taken from this 300 thing as training and the remaining is used as testing. You can see that as you increase the degree of the polynomial the variability or the estimates or the range of the estimates is very very large.

So, it indicates that if you over fit you will get a very high variability whereas, on the other hand if you choose order of the polynomial 1 or 2 you find that the variability is not that significant comparatively. So, typically if you over fit the model you will find high variability in your estimates that you obtain or the mean squared error values that you obtain ok.

(Refer Slide Time: 14:08)

All this is good if you have a large data set what happens when you have extremely small data set and you cannot divided into training and a validation set? You do not have sufficient samples for training. Typically you need a reasonable number of samples in the training set to build the model therefore, in this case we cannot set a part or divided into a 7030 what I call division and therefore, you have to do some of the strategies.

So, these strategies what are called cross validation using k- fold cross validation or a bootstrap that we will see. Here again we will predict the performance of the model on the validation set, but the validation set is not separated from the training set precisely, but on the other hand it is drawn from the training set and we will see how we do this. So, these methods K- fold cross validation is useful when we have very few samples for a training.

(Refer Slide Time: 15:05)

So, I will first start with leave one out cross validation what is called LOOCV. In this case we have n samples as I said n is not very large may be 20 or 30 samples we have for training. So, what we will do is, you first leave out the first sample and use the remaining samples for building your model; that means, you will use samples 2, 3, 4 up to n, it to build your model and once you have build the model you test the performance of the model or predict for this sample that you have left out and you will get an MSE for this sample.

And similarly in the next round what you do is leave out the second sample and choose all the remaining for training and then use that model for predicting on the sample that is left out. So, in every time you build the model by leaving out one sample from this list of n samples and predict the performance of the model on the left out sample. This you will do for every choice of the model parameter.

For example, if you are building a non-linear regression model, you will build regression model using let say only beta naught and beta 1 the linear term and predict a MSE on this left out sample. You will also build a model using the same training set quadratic model and predicted on this and so on so forth so, that you will get the MSE for the left out sample for all choices of the parameters you want to try out. Do this with the second sample being left out and the third sample being left out in term ok.

So, that every sample has a chance of being in the validation set and also be being part of the training set in the other cases. So, once you have done this for a particular choice of

the model parameters, let say you have building a linear model. You find out the sum squared value of the prediction errors on the left out sample over all the samples. For example, you would have got a MSE for this, MSE for this, MSE for this when the first sample second sample and third sample was left out that you are cumulating it here and taking the average of all that. This you do repeatedly for every choice of the parameters in the model.

For example, the linear, the quadratic, the cubic and so on so forth and you get the mean squared error or cross validation error for different values of the parameters which you can plot.

(Refer Slide Time: 17:33)
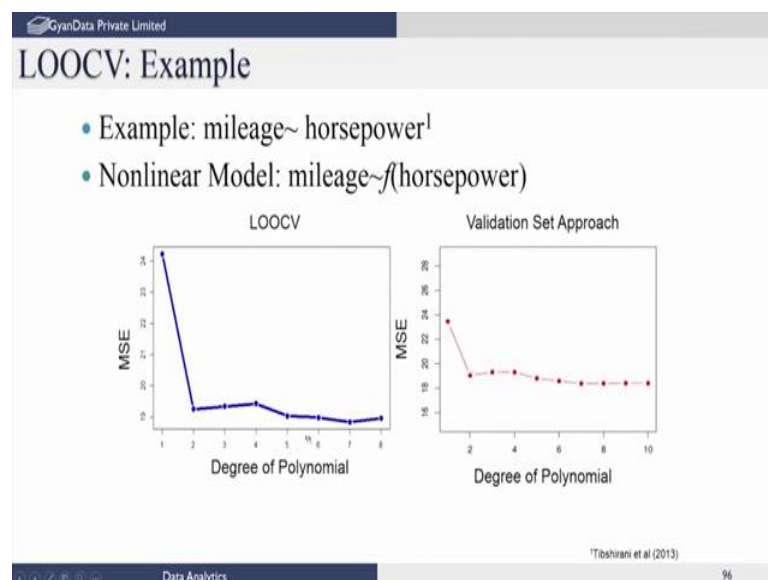
Here again we have shown the mean squared error for different choice of the degree of the polynomial for the same data set, in this case we have use to left leave one out cross validation strategy; that means, if you have 300 samples, we would have left out one sample build the model using 299 samples predicted on the sample that is left out, do this in term average over all of this for every choice of these parameter degree of the polynomial and mean squared error we have plotted.

Again we see that the MSE on the cross validation leave one out cross validation reaches more or less a minimum for a degree of polynomial equal 2 after which it just keeps remains more or less flat. So, the optimal in this case is also indicated as a second order a polynomial is best for this particular example.

(Refer Slide Time: 18:20)

So, leave one out cross validation as an advantage as compare to the validation set approach when 2 are we can show that it does not over estimate the test error rate as much as the validation set approach ok. It is comparatively expensive to implement because you are building the model n times one for each sample being left out and you have to repeat this for all choices of the hyperpara model parameter.

For example you have to do this for the linear model, the quadratic model, the cubic model and so and so forth. So, you have not only have to do this n times, but you have to do this n times for every choice of the number of parameters of the model. So, it is quite a lot of computation that it takes. In general it may actually sometimes fit the model which is slightly more than the optimal model by not always, but sometimes it is also possible that the leave one out cross validation procedure over fits the model.

(Refer Slide Time: 19:21)

We can also do what is called the k- fold cross validation, where here instead of leaving one out we first divide the entire training set into k folds or k groups. So, let us say the first group contains let say the first four data samples, the second group contains the next four and so and so forth and you have divided this entire n samples into k groups. Now instead of leaving one out we will leave one group out.

So, for example, in this first case we will leave the first four samples they belonging to group 1 and use the remaining samples and build a model for whatever choice of the parameters where we have use let say we have building a linear model, we will use the remaining groups build the linear model and then predict for the set of samples in group 1 that was left out and compute the MSE for this group.

Similarly, in the next round what we will do is leave group 2 out build the model let say the linear model that we are building with the remaining groups and then find the prediction error for group 2 and so on so forth until we find the prediction error for group k and then we average over all these groups. So, the MSE in this case for all groups whether there are k groups and 1 by k that will be the cross validation error for leave this k- fold cross validation.

Now, you can you have to repeat this for every choice of the parameter you have done this for the linear model you have to do this for the quadratic model, cubic model and so on so

forth and then you can plot this cross validation error for leave for this k- fold cross validation.

(Refer Slide Time: 20:59)

Notice that if k equals n you are essentially going back to leave one out cross validation in practice you can choose the number of groups equal to either 5 or 10 and do a 10 fold cross validation or 5 fold cross validation. This is obviously, less expensive computationally as compared to leave one out cross validation and as you see that leave one out cross validation we will do a model fitting n times for every choice of the parameter whereas, k-fold cross validation will do the model building k times for every choice of the parameter.

(Refer Slide Time: 21:35)
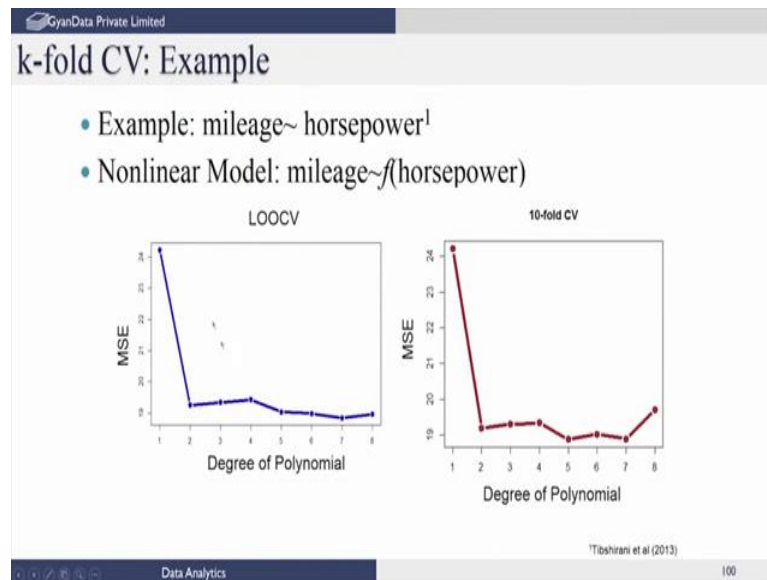
Again we have illustrated this k- fold cross validation for this mileage auto data, again we plot the MSE for different degrees of the polynomial. We have used to 10 fold cross validation and we are plotting this error and we will see that here also the minimal error occurs at 2 showing that the quadratic model is probably best for this particular data after which the error actually essentially flat and so.

So, cross validation is a important method or a approach for finding the optimal number of parameters of a model, this happens in clustering, this will happen in non-linear model fitting and principle component analysis and so on and its useful later on you will see in the clustering lectures, the use of cross validation for determining the optimal number of clusters.

Thank you.