Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 07
Lecture - 32

Lecture 32: Resampling methods

Hello everyone. Welcome to the seventh module. In the seventh module, we have started discussing about Evaluation of Recommender System and in this context we have introduced the topic in the last lecture. And in today's lecture, we are specifically going to talk about the Resampling Methods. Now, before we talk about the resampling methods, let us try to understand where exactly are we going to use this resampling methods. Think of the evaluation strategies.

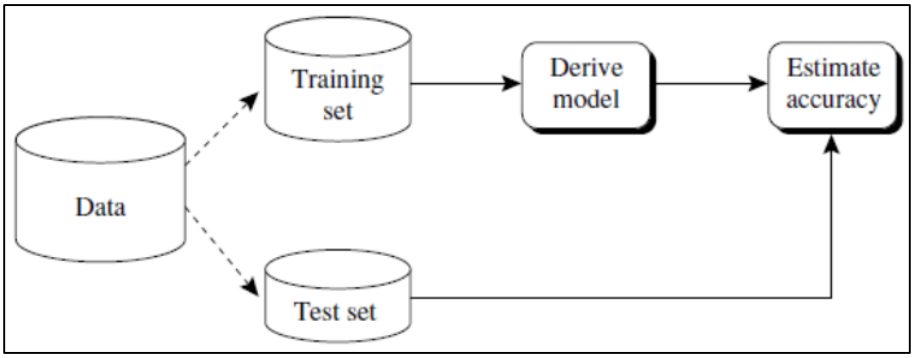
We have three types of strategies for evaluation. First the offline, then we can have user studies and online. In case of offline, we already have the data set available with us and we are supposed to test our algorithm using this data set. Now, the problem here is we really cannot take the real users or use an real online setting for this evaluation.

So, therefore, we are supposed to use the data which is available to us in such a manner, so that the way we evaluate shouldmake sure that whenever we encounter any unknownunknown entity, we will be able to correct the make the correct classification or correct the make the correct prediction as is required. So, this resampling methods now come into picture. Now, this resampling methods involve repeatedly drawing sample from a training set and fitting a model of interest on each sample in order to obtain additional information about the fitted model. So, through this approach, we obtain the information that would not be otherwise available from the fitted model if we do it only one using the original sample. Typically, there are three methods for doing this resampling.

Holdout method and which is has one extension called random sub sampling, then cross validation. Here, we have k fold cross validation and leave one out cross validation, then we have bootstrap. Now, in case of holdout method, the given data set is randomly partitioned into two independent sets or training set and a test set. And we test our model, we fit our model with the training set and we find out its validity with respect to a test set. So, as we will be seeing afterwards usually 75 percent or more are used in the training set 75 to 80 percent, then rest are used in the test set.

Then we have cross validation which can be used to estimate the test error associated with a given statistical learning method in order to evaluate the performance to select the based on certain selected appropriate level of flexibility. So, now the process of

evaluating the models performance is known as model assessment whereas, whereas the process of selecting the power the proper level of flexibility for a model is known as model selection. The next method is your bootstrap which is used in several context most commonly it is used to provide the information about the about the parameters of a model. Now, we start with the first method it is holdout, it is called holdout method or the validation set method. In this method, the given data set randomly partitioned into two independent sets or training set and a test set.



So, as shown in this figure this entire data which we call as the training set, I call as the which is available to us is randomly partitioned. It is not that only the first part will be used for training and first 75 percent next 25 percent will be testing. No, it is randomized then around 75 to 80 percent usually it is two-third is used here and one-third is used for testing. Now, this is used to derive a model and this models accuracy is assessed with respect to the test set. Now, here the estimate is pessimistic because only a portion of the initial data is used to derive the model.

A little variation of this is called random sub sampling. What it does is it carries out this holdout method in which this procedure this holdout procedure is repeated k times. The overall accuracy is estimated taking the average of this accuracy is obtained from each iteration. So, which means in this case this is one iteration then again starting from the full data you make one more two-third one-third partition and find out the accuracy and you continue this for k number of steps. Now, the overall accuracy is estimated as the average accuracy obtained from each iteration.

Now, to understand the process of this holdout method in this example let us saywe have we have we have to make two sets blue is the training set, red is the test set and this test set in this particular example has 30 percent of the data points and we are we are trying to fit a regression line. So, as per our as per the evaluation criteria little bit we have studied earlier maybe we can use the RMSE to find out. So, with respect to these red points we will be finding out the error. So, next is your next sub sampling method is your cross validation. So, cross validation can be k fold cross validation.

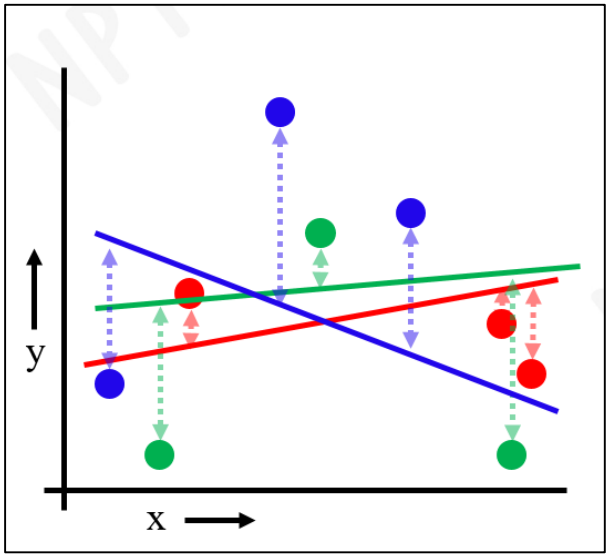
In case of k fold cross validation, the initial data are randomly partitioned into k mutually exclusive subsets or folds. So, when we say it is mutually exclusive subsets or folds once again we do not say that we have to be taking suppose this is my entire data set I will be starting from the beginning this is my first fold this is my second fold. So, you with equal number total let us say I have total n number of records. So, from the beginning I will be taking n minus k first n minus k second n minus k and so on. So, I will not be doing that rather randomly this I will reshuffle it and we will be taking one partition with those randomly shuffled data.

So, let us say there are total k number of such partitions and all these k number of partitions are of equal size. Then just like we did our this method now for partition 1 we will be carrying out this for in this case in holdout this is the entire data. In case of k fold cross validation out of those k number of partitions we will be using k minus 1 to train the data and another one to test the data. So, what happens in the and this continues what happens in this case in the first iteration you will be using let us say D 1 to let us say we are using D 2 to D k to train the model and D 1 for test. So, this becomes my training set.

So, next time what happens I will be making D 1 D 3 leaving D 2 as my train set and test will be on D 2 and maybe next time I will be making D 1 D 2 D 4 up to D k as my training set and I can test with test with D 3 which I left. So, every time I will be considering k minus 1 partitions put together to train and another one to test. As a result, we make sure that all the data points participate in both training set as well as in the test set. So, through this process we can have total k estimates of test error. So, mean square error 1 mean square error 2 mean square error k and average of that will be giving us the cross validation k fold cross validation mean square error.

So, this is what I was trying to say suppose we have these these dots are our data points these are our data points. Now, let us say we are making 3 partitions of this data points there are total 9 points we made 3 partitions. So, the red green and blue are 3 partitions. So, first we will be leaving k and fit the model with blue and green as a consequence we get this regression line. Then we keep let us say green and we keep the green aside and with red and blue we train.

So, this is the model and this is the third one. So, taking for each with respect to each of these now we will be getting some mean square error. So, here k is equal to 3 k is equal to. So, this is 3. So, we will be getting mean square error 1 2 and 3.



$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i.$$

Now, if we take the average then we will be getting this cross validation mean square error. Now, there is a special type of cross validation ah k fold cross validation where the total number of data points is set as the value of k. So, which means if we have total n number of data points then all of them make one partition each. So, if you make all of them one partition each then what happens you have total n number of data points 1 2 up to n. So, in the first time you will be keeping n as your test and with 2 to n will be your train set.

Second time keep 1 make 2 to your test set you test with respect to 2 only and 1 3 up to n leaving to make your training set ok. And in this process how many times you will be doing n times because every time only one of the observation you will be keeping aside for testing with remaining you will be training. So, with each training and while testing with the left out observation you will be getting one mean square error. And average of this over all the data points will be giving you the leaf one out cross validation error in terms of this mean square. So, this is the example where we have again total number of 9 data points the here the first data point, second data point look first one is now included with the first data keep aside the first data point.

So, keeping this aside now this is my regression line. So, now, with respect to this regression line I find out the error of with respect to this. Similarly, in the second iteration this is the point. So, with leaving this out now with the remaining with the remaining points we have this line green line fitted. So, as a result we have to find out error with respect to this error with respect to this.

So, these are with respect to all these points which are left aside here, here, here, here, here and here. So, which means total 9 times I have to carry out this procedure. So, it makes sure that every point individually is taken care of. Now, the question comes we were finding out the mean square error in case we have a regression like model where we have the use let us say some observation y i minus y i cap we use to square then we use to take the sum. Now, in case of a classification problem we have to have a class.

So, whether thus now the situation is whether it belongs to the class or whether it does not. So, therefore, for classification problem the accuracy estimate is the overall number of correct classifications from k iterations divided by the total number of tuples in the initial data. So, in this method we have to find out how many times we have made the mistake. So, this sum of this error we have to take and normalize it. So, now, what is this term? This is an indicator variable that is equal to 1 if y i which what is y i? y i is your class variable if class variable is equal to y i cap and it is 0 if it is sorry if it is equal to 1 if it is I mean the error can happen when it is not equal to 1 and otherwise it is when it is equal the error is 0.

So, now, if we have 0 1 it is the problem. Now, if we have more number of classes what do we do? We do something called a stratified cross validation. In case of a stratified cross validation we can we have to make sure that the class distribution of tuple in each fold is approximately same as that of the initial data ok. So, what do you mean by class distribution? Suppose this value of y i there will be some n number of records. So, these are the features features and this is my class variable.

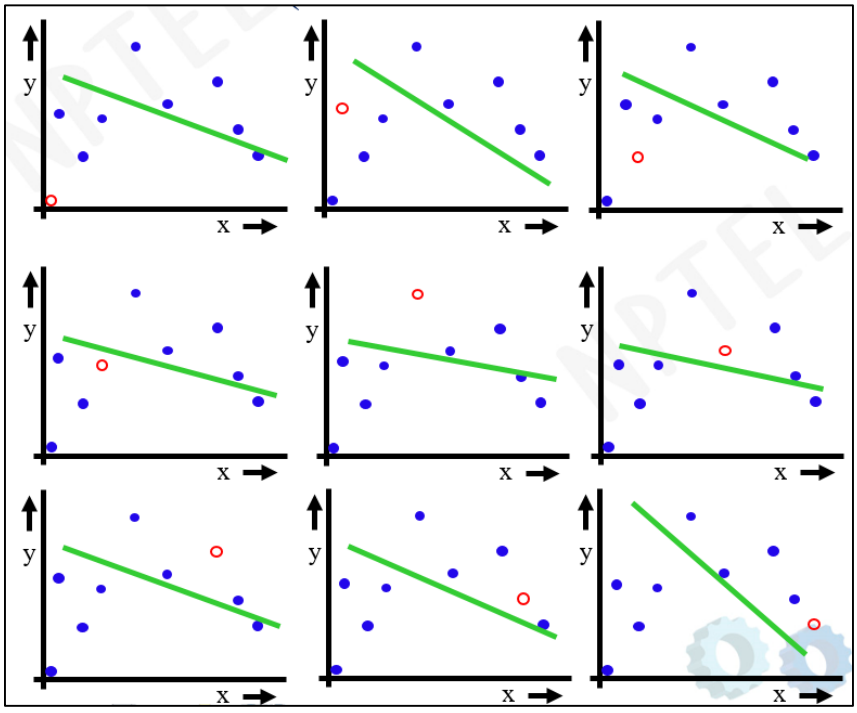
Class variable can take values let us say 0 and 1 or it can take even multiple values. Now, here there will be some 0s some 1s again some 0s 1s and so on. So, under this setting while making this folds it may so happen that all the 0s become. So, you have a fold of all 0s and another fold with all 1s. So, this is not a good idea what classification you will be carrying out because here every element is same.

So, what will you do? You will be making this randomization in a manner. So, that if you are making total k folds in each fold you have same distribution of 0. Let us say here total 0 occurs 60 percent and number of 1s occur 0s occur let us say 0s occur 60 percent of the time and 1s occur 40 percent of the time in this. So, in each partition again we will be sampling in a manner. So, that y i will be 0 in 60 percent of the observation approximately and y i will be sampling in 40 percent of the observations approximately.

So, there are some important considerations with respect to this cross validation. Now, unlike this holdout and random sub sampling method in cross validation each sample is used the same number of times for training and once for testing ok. So, which means all the elements participate in both training phase as well as in the testing phase. The validation set approach can lead to overestimate of the test error. Since in this approach the training set is used to fit the statistical learning method only using only two third of the observation of the entire data set.

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i.$$

Now, comparing k fold cross validation and leave one out cross validation it is seen that leave one out cross validation will give a slightly unbiased estimate of the test error. Why? So, because each training set contains n minus 1 number of observations which is almost as many number of observations in the full data set. However, there are issues with higher variance. If you remember while talking about talking about the foundations of machine learning etcetera, we discussed little bit about this bias and variance. So, variance is again of two type one is that of associated with the error the residual and another with the observations.



So, we are basically talking about that variance. Now, however, it is seen that this leave one out cross validation has a higher variance than k fold k fold ok. So, which means

when we perform this ah leave one out cross validation we are in effect averaging the output of n fitted model each of which is trained on almost the identical set of observations. So, therefore, the output are highly and positively correlated. So, therefore, the variance component will be ah more.

Now, when we perform k fold with k less than equal to n we are averaging the output of k fitted model that are somewhat less correlated with each other. Now, since the overlap between the training set and the model is small the variance is less. Now, to summarize there is a bias variance tradeoff associated with the choice of k in k fold cross validation. Typically given this bias variance ah tradeoff consideration one can perform k fold cross validation using k equal to 10 it is observed empirically that these two with k equal to 5 or k equal to 10 usually you get the lower test error rate. So, in this case neither there is excessive bias nor there is very high variance.

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Err}_i, \quad \text{where } \mathrm{Err}_i = I(y_i \neq \hat{y}_i)$$

$I(y_i \neq \hat{y}_i)$ is an *indicator variable* that equals 1 if $y_i \neq \hat{y}_i$ and zero if $y_i = \hat{y}_i$

Now, come to the bootstrap the bootstrap method samples the given training tuple uniformly with replacement. Now, what is with replacement suppose you have n number of records 1 to n. So, from this you choose a part for your training set. Let us say you have choice of 43, ah 24, 3 and 15. Let us say this is the part of your training set you have chosen.

Now, when you choose the next one you replace all these records I mean you choose one and then replace choose one then replace. So, as a result it is possible that you may get let us say 24 once more or 43 once more maybe it wants 24 itself some 3, 4 times ok. So, that is each time a tuple is selected it is equally likely to be selected again and re added to the training set ok. Now, this bootstrapping the commonly used bootstrapping method is called 0.632 bootstrap. So, here it works like this suppose you have suppose you have total d number of tuples d number of tuples ok. Now, the data set if you sample this data set d times with replacement d number of tuples and you bootstrap and resample it d times. So, with replacement. Now, the resulting bootstrap sample of training set will have exactly d sample. Now, it is very likely that some of the original data tuples will occur more than once in this sample and the data tuple that did not make it into the training set will end up in the test set.

And it turns out that if you run this procedure almost 62 point 63.2 percent of the original data tuples will end up in the bootstrap sample and remaining will be forming the test set. So, which means if you are taking if you are taking out of this d number of samples suppose you are carrying out the sampling procedure a number of times there

will be many repetitions. So, including all these repetitions it is not possible at all that all the d samples will go to the training set. So, even if you carry out it d times all the d number of samples will not go to the training set only this many will go.

$$Acc(M) = \frac{1}{k}\sum_{i=1}^{k}(0.632 \times Acc(M_i)_{test\_set} + 0.368 \times Acc(M_i)_{train\_set})$$

So, which means here there will be many repetitive patterns. Now, the sampling procedure is repeated k times where in each iteration the current set isset to obtain the accuracy estimate of the model obtained from the current bootstrap sample. The overall accuracy of the model is estimated like this. Now, what is this number? This number is coming out of this 63.2 percent. So, 63.6 0.632 into accuracy of the m ith test set and this into accuracy of the m ith trend set. So, this is the reference I have used many of the examples that I considered I have collected from here and ideas basically started with this Hannon camber every from everywhere I have used and this is a very good reference and in today's context this is a recent one as well. So, which gives a good survey on evaluating recommender systems. So, if you wish you can follow.

So, now, we are in a position to conclude our lecture. So, now, resampling methods involve repeatedly drawing sample from a training set and refitting a model of interest on the sample in order to obtain the additional information about the fitted model. Most commonly used methods are hand holdout method, cross validation and bootstrap. Now, in holdout method the given data are randomly partitioned into two independent sets training set and test set. The training set is used to build the model and test set is used to test its accuracy. In case of k fold cross validation, we partition the data into k mutually subsets or the folds.

Now, one of this fold is kept out as the tested and the training set is made using the remaining k minus 1. So, as a result to find out the overall accuracy you sum the individual ones and divide it by k. Now, in case of leave out leave one out cross validation it is a special type of k fold cross validation where k is set as the number of initial tuples. So, which means the total n number of tuples.

So, one at a time is used for testing purpose. Now, bootstrap method samples are given training tuples uniformly with replacement. With this we finish this lecture. Thank you.