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 - 33

Lecture 33: Evaluation metrics for accuracy

Hello everyone. Welcome to the third lecture in the series on the seventh module that is actually thirty third lecture of this entire course. So, here we are going to talk about the evaluation matrix that we are going to use in case of a recommender system specifically for accuracy. And if you remember while talking about evaluating recommender system besides accuracy there are many more measures, serendipity, diversity and so on. However, those measures are difficult toevaluate, difficult to quantify in case of offline setting. But as you know even if we have three strategies, when we have a new algorithm the first task that has to be done is to do to carry out the offlineevaluation and find out the performance of that algorithm one or more algorithm before we deploy them in thein the online environment, online and your experiment environment.

So, therefore, evaluating whatever matrix we are talking about mostly they are used for offline evaluation. Some of these can be used for online as well, but getting the online data readily is problematic. And that is why in the last lecture we saw if we are sticking to offline evaluation in the first phase then we have to adopt many kind of resampling approaches. Now, we start with the assumption that we are using this evaluation matrix adopting this resampling approaches specifically in case of offline evaluation.

So, what are these matrix? So, these matrix this matrix foraccuracy they actually measure how good or how accurate a method is at predicting. Now, this prediction accuracy is by far the most discussed property in the recommendation literature. A basic assumption here is that a system that provides the accuratethe accurate rating is supposed to be the best system which may not be true in many cases. For example, if we are thinking about the diversity and all simply by showing oneaccurate observation which will be very repetitive as we have discussed that time. Suppose I am watching comedy movies all the time I am showing the comedy movies or if because of somesome local situation if I am looking at a particular news article all the time showing showing the news article to that particular person or location probably will not be a good idea.

So, however, as the literature goes prediction accuracy typically is the widely adoptedmeasure for accuracy. Now, this prediction accuracy typically is independent of user interface and thus can be measured in offline experiment as well. And why offline experiment as well? It is most appropriate for offline experiment setting. Now, measuring prediction accuracy in the user study depends on observing the users and probably

collecting the their their preference data through a questionnaire. Now, in the context of recommender system specifically we have to assess three kinds of accuracy.

Accuracy of rating prediction, accuracy of usage prediction and accuracy of ranking of items. So, we will go one after the other. So, first task is measuring rating prediction accuracy. There are two measures for this root mean square error and mean absolute error. Now, in both in both these cases we use the predicted value this is the actual value.

So, where from this predicted and actual values are coming? These are coming from my my test set. So, from the test set I build the model using train set and I will be testing it using test set. So, this is the value I got from test set. So, for all the user item pairs that belong to my test set I will be calculating this value this is actual, this is this is the test set this is predicted ok. Now, here all the terms I mean for all the ratings in the test set we take this difference square and take the root of sum it and take the root over.

But in this case we take the absolute value. Now, coming to this looking at computation suppose ah suppose this is my actual value this is predicted value these are the residuals. So, these are the square residuals. So, we have to take the sum divide by we have let us say total 10 observation divided by 10 this is my value RMS is just the root over of that. But in case of mean absolute error I take the absolute value of all this.

So, so first few were actually first 5 are actually positive values, but next 2 were negative values. So, we have taken the absolute value for them and these are the remaining and we take the average. Now, the question is when to use what is the relative advantage of AME and mean absolute error and RMSE. Now, think of a situation like this suppose you have a test set consisting of 4 points 2, 3, 4 points. Now, suppose in there is error here error is 2 here error is 2 here error is 2 here and in one case it is 0.

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} (\hat{r}_{ui} - r_{ui})^2}$$

And there is another algorithm which is giving the error this is let us say which is giving let us say only error 3 here which is higher and rest are 0. So, what happens here actually in many of the data there is error, but in one there is no error, but here in one the error is high, but in rest it is 0. So, if we carry out both RMSE and mean absolute error RMSE will be giving better value here and mean absolute error will be better in the second case. Why? Because RMSE actually disproportionately penalize the large errors ok. So, if this error is large it will be penalized more even if in more number of cases we have better values.

So, you have to judiciously use this depending on the kind of data set that you are using. Now, these values can be extended further. For example, these concepts can be extended further. For example, normalized RMSE normalized MAE are the versions where we normalize these values by dividing them with this range. So, they are simply some scaled version of RMSE and AME.

$$\text{MAE} = \frac{1}{|\mathcal{T}|} \sum_{(u,i)\in\mathcal{T}} |\hat{r}_{ui} - r_{ui}|$$

So, the result ah. So, the result of the ranking algorithm is same as that of the ranking given by the on normalized measure. There can be average RMSE and average AME as well and they adjust well for unbalanced data set. So, if the test data set has an unbalanced distribution of items the RMSE and AME obtained from it might be heavily influenced by the error on very few frequent items. So, therefore, if we take this average value we ah that bias we get rid of.

This can again be further extended in at the item level. So, if we need to measure at the item level we have two matrix here good item MAE and good predicted item MAE. So, if we for each item we compute this then it is called good item MAE for the entire data set entire test data set and if it is for the recommendation recommended item lists only it is called ah good predicted item MAE. Now, one can also compute a per user average RMSE or MAE if the test set has an unbalanced user distribution and we wish to understand the prediction error ah understand the prediction error a random on a randomly drawn user. I mean the if it is unbalanced there can be some kind of bias associated.

So, which can be removed through this process. Now, measuring usage prediction we talked about three things one is we were measuring the prediction accuracy rating prediction accuracy ok. Now, we are predicting the usage prediction. Now, in many applications in the recommender system only predicting the preference users preference or the users rating on a particular item is not enough. We are rather would be testing whether the item is predicted or not.

Now, in an offline environment ah setting of usage prediction we typically have a data set consisting of items on for each user ah items which and each user has rated. Now, ah for this purpose for measuring this usage prediction we select a test user and hide some of her selections and ask the recommender to predict a set of items that the user will use. So, what is your trend set here? User has let us say given ratings for ah some set I U of. So, I U is the items rated by the user. Now, out of this I U now you partition again you make a trend set and a test set ok.

So, now, in the trend set a subset of this will go and you will be predicting the items for the you will be predicting the ratings for the item in the test set. You will be predicting whether the test set items will be recommended or not. So, on such setting you can create something called a confusion matrix. So, in this confusion matrix suppose you have the user has actually used and you have also recommended it through your algorithm then you have two positive cases and it is not used by the user, but your recommendation algorithm recommends the item. So, it is a false positive case.

Actual	Predicted	Residual	Squared Residual
87	85	2	4
92	90	2	4
65	62	3	9
78	76	2	4
55	53	2	4
89	91	-2	4
73	75	-2	4
96	94	2	4
80	78	2	4
68	65	3	9

Mean Squared Residual
= (4 + 4 + 9 + 4 + 4 + 4 + 4 + 4 + 9 + 4) / 10
= 45 / 10 = 4.5

$RMSE = \sqrt{4.5} \approx 2.12$

Mean Absolute Error (MAE)
= (2 + 2 + 3 + 2 + 2 + 2 + 2 + 2 + 2 + 3) / 10
= 22 / 10 = 2.2

So, similarly you have false negative when the user has used the item, but it is not recommended by the algorithm and you have true negative where it is not used ah by the user not used by the user in the sense not it is not in the test set, but it is also not recommended. So, it is a true negative. So, you have true positive, false negative, false positive and true negative. So, this matrix is called the confusion matrix. Now, based on this confusion matrix a number of matrix can be derived from this confusion matrix this is a confusion matrix.

Confusion matrix.

	Recommended	Not recommended
Used	True-positive (tp)	False-negative (fn)
Not used	False-positive (fp)	True-negative (tn)

So, precision is true true true positive by sum of true positive and false positive true positive and false positive. So, it is precision is here this way ok. So, this is this is where you have your precision. So, it is true true positive divided by number of true positive plus false positive ok. Now, recall true positive rate true positive divided by true positive and false negative.

So, this way true positive divided by true positive or false negative. So, similarly your false positive rate which is 1 minus specificity which is false positive divided by true positive and false negative and this is actually 1 minus specificity and there is something called F score which is otherwise also known as F 1 score which is basically the harmonic mean of precision and recall and is calculated in this manner. So, typically we can expect a tradeoff between these quantities. So, while allowing longer recommendation list typically the recall will improve, but it will reduce the precision where the number of recommendation in applications where the number of recommendations that can be presented to the user is preordained. So, which means it is pre decided and kept the most useful measure interesting measure of interest is precision at precision at n recommendation ok precision at nth recommendation.

$$\textbf{Precision} = \frac{\#tp}{\#tp + \#fp}$$

$$\textbf{Recall (True Positive Rate)} = \frac{\#tp}{\#tp + \#fn}$$

$$\textbf{False Positive Rate (1 - Specificity)} = \frac{\#fp}{\#fp + \#tn}$$

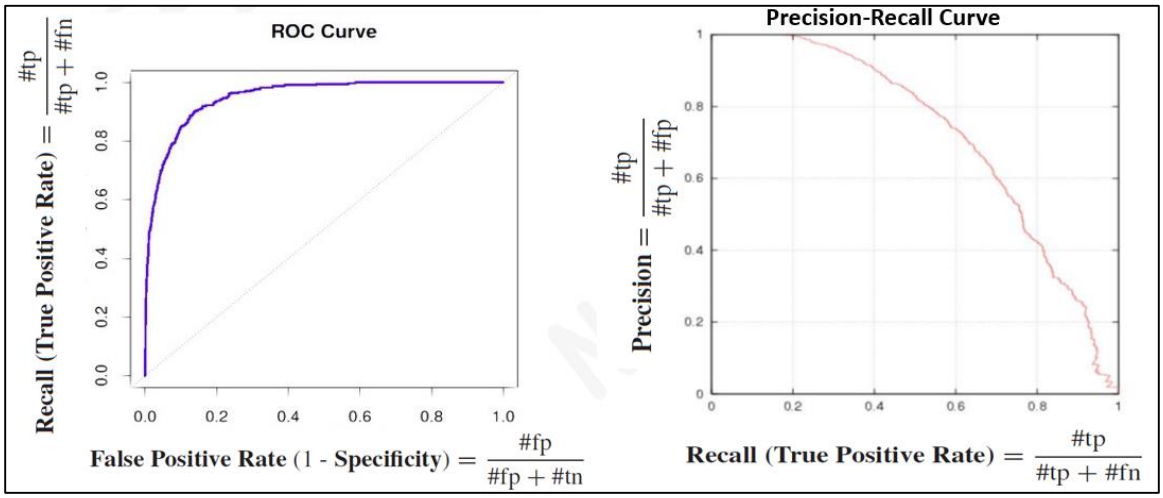
$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Now, come to the usage prediction. So, in case of usage prediction these are the matrix. So, now, we are supposed to see little bit more detail. Now, the applications where the number of recommendations that are presented to the user is not preordained. So, which means they are not pre decided and kept it is preferable to evaluate the algorithms over a range of recommendation list length.

So, when you increase this length consequently your precision and recall will keep changing as we saw here if we allow longer recommendation typically recall will improve, but your precision will reduce. So, such observations in a systematic manner can be seen through precision recall curve and there is another way in which we can we can also connect two other quantities ok. So, we can compare precision recall through this or we can true sensitive and sorry true positive and false false positive rate we can compare and we call it operating or receiver operating characteristic curve or ROC curve. So, precision recall curve and ROC curve are obtained by repeatedly taking the observation. In this case in this particular case with respect to top n list n equal to 2, 2 items are suggested, 3 items are suggested, 4 items are suggested and so on.

Now, both curves measure the proportion of the preferred items that are actually recommended then this precision recall curve emphasize on the proportion item proportion of recommended items that are preferred while ROC curve emphasizes the proportion of the items that are not preferred that end up being recommended. A typically

ROC curve will have a look where it will grow and then it will stabilize and here it will be a this is one increasing one and here it will be decreasing one. This side you will have recall this side you will have false positive rate and this is your ROC curve, this is your precision recall curve here you have recall here your precision here you have false positive rate and it is true positive rate true positive rate is your recall. So, now, while both the curves measure the proportion of the preferred items that are actually recommended, precision recall curve emphasizes the proportion of the recommended items that are preferred where are. So, in this context in this context precision recall curve what we exactly would like to see would like to see that we have recommended something and it is also preferred by the user.



In case of ROC we would like to see we have recommended something, but it is not used by the user it is not preferred by the user. So, this is a small computational task suppose our actual value of y this is y cap and this is the predicted value. Now, suppose based on this predicted value we will be recommending top view. Now, assuming that we have certain threshold. So, as the threshold becomes very high number of recommendations made on top n decrease.

For example, here when we set this to 0.7 our how many we are recommending 2 only we are recommending because here only we have more than I mean the 0.7 or above. Now, in case it is 6 then one more is added 2 more are added these 2.

So, here we have 4 and so on. So, here n top n. So, here n is equal to 2 here n is equal to 4. So, this is with some sample threshold. So, as you change this threshold this n will keep. So, with a very low threshold you are actually showing all the n ok.

So, now, how do I calculate this true positive rate? It is true positive rate is true true positive divided by true positive plus false negative and false positive rate is false positive divided by false positive plus true negative. So, these are now computed. Computed with respect to n equal to 2 n equal to 4 and this is n equal to 1 2 3 4 5 6 n

equal to 6 this is n equal to n equal to 7 this is n equal to 8 this is n equal to 9 this is n equal to all 10. So, these values these 2 values now we plot. So, if we plot we are likely we are going to get these are the values which are calculated here and we plot we get a plot like this.

Example: ROC Curve

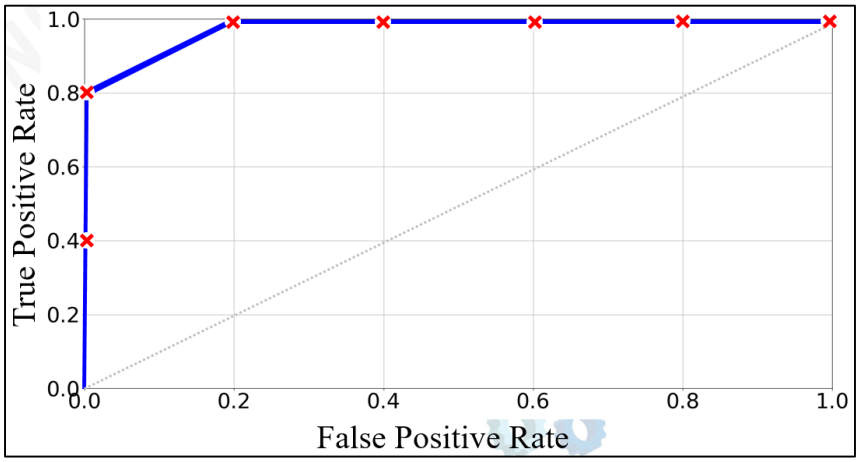
Threshold: [0.1, 0.13, 0.2, 0.4, 0.5,0.6, 0.7]

Actual (y)	Predicted Probability ($\hat{y}$)	$\hat{y}(0.1)$	$\hat{y}(0.13)$	$\hat{y}(0.2)$	$\hat{y}(0.4)$	$\hat{y}(0.5)$	$\hat{y}(0.6)$	$\hat{y}(0.7)$
0	0.20	1	1	1	0	0	0	0
1	0.70	1	1	1	1	1	1	1
0	0.10	1	0	0	0	0	0	0
1	0.70	1	1	1	1	1	1	1
0	0.13	1	1	0	0	0	0	0
1	0.60	1	1	1	1	1	1	0
1	0.50	1	1	1	1	1	0	0
0	0.50	1	1	1	1	1	0	0
0	0.40	1	1	1	1	0	0	0
1	0.60	1	1	1	1	1	1	0
TPR=		$\frac{5}{5+0}=1$	$\frac{5}{5+0}=1$	$\frac{5}{5+0}=1$	$\frac{5}{5+0}=1$	$\frac{5}{5+0}=1$	$\frac{4}{4+1}=0.8$	$\frac{2}{2+3}=0.4$
FPR=		$\frac{5}{5+0}=1$	$\frac{4}{4+1}=0.8$	$\frac{3}{3+2}=0.6$	$\frac{2}{2+3}=0.4$	$\frac{1}{1+4}=0.2$	$\frac{0}{0+5}=0$	$\frac{0}{0+5}=0$

$$TPR=\frac{\#TP}{\#TP+\#FN}$$

$$FPR=\frac{\#FP}{\#FP+\#TN}$$

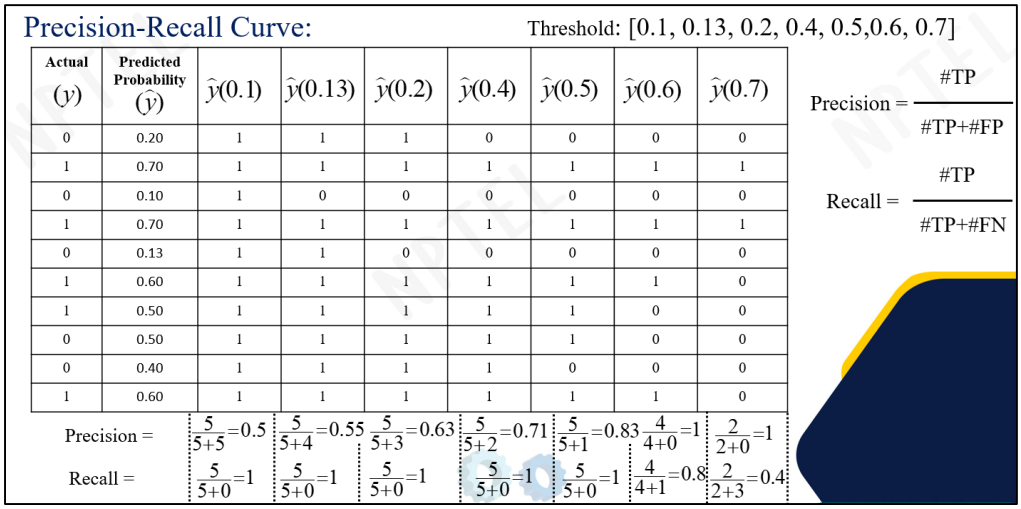
We have a very small example and few data points only. So, therefore, the curve looks little not so smooth like this one, but this is generally is the case when you increase the number of data points. Now, come to similarly we can find out the precision recall curve and with the same data with n equal to different values 2 4 6 7 8 9 10 ok. So, this is n equal to 2 n equal to 4 n equal to 6 7 n equal to 8 n equal to 9 and n equal to 10. We plot we calculate precision and recall using this formula and we get this precision recall curve.



So, it is again not looking very smooth like this one, but kind of showing the same behavior. Now, next thing is we have to measure the accuracy of the ranking. So, far what we have done? We have found out the prediction accuracy, then we have found out whether the item will be appearing in the top n, then now we are trying to find out where

in the top n. So, for this purpose what we are supposed to do? In fact, such kind of applications you will find in many places ok. Where suppose you are seeing movies in Netflix or looking at the items in looking at the items in case of Amazon what will happen? You will be seeing a list of movies or list of movies.

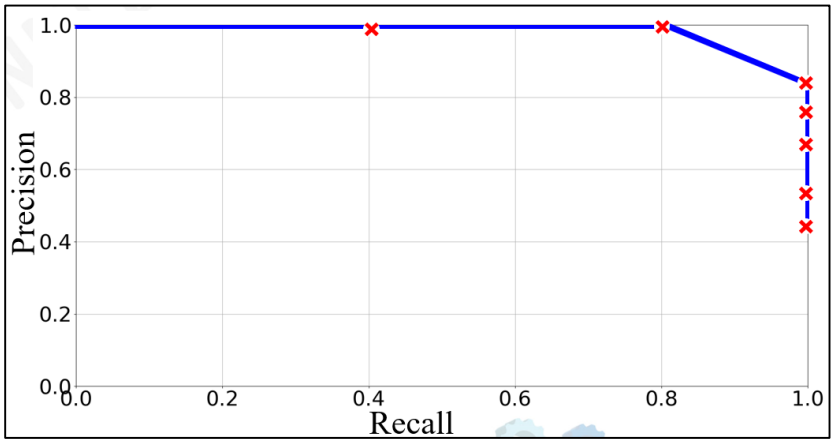
List of movies will appear on your screen. So, if you browse further you will be getting more movies, but do you have patience to browse or you will be going through this top view and top view go through the top view and find that you do not have something which is a good enough for you. So, therefore, to it is very important that you show those movies which the user is likely to like first. So, therefore, it is very important that where exactly you are putting the item in the ranked list. Now, this ranking metric consider the position of correct item in a ranked list that take into account. Now, relevant the relevant items are more useful when they appear earlier in recommendation list.



Particularly important in recommender system it is particularly important in recommender system as the lower ranked items are generally overlooked by the user. So, this can be done by two approaches one is reference ranking, second is utility based ranking. In case of reference ranking you already have a reference set available to you. With respect to which you will be evaluating. The second approach you have to construct some kind of utility function.

Look at this, this is you have recommended ok and this is actually the list actually the reference list. Look in the reference list this is the top item you have predicted it, but it position is in the second. This is not in the list, but you have predicted it this is also not in the list, but now it is position is 2 ok. So, now we have to have certain, but suppose there would have been another algorithm where this would have appeared in the top position this item would have been the number one position. What you would have done? That you would have assumed that in terms of ranking that one is doing better that the other algorithm is doing better.

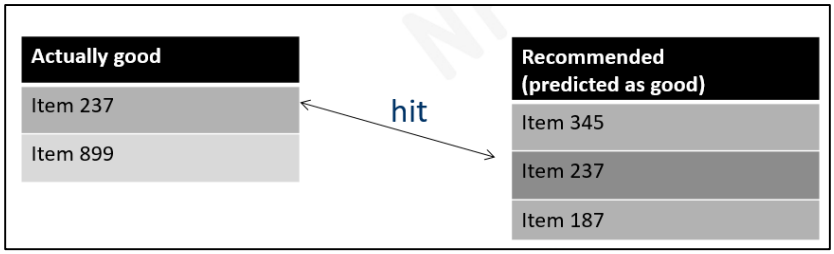
Now, one of the approach for measuring the accuracy of ranking items based on this preference is based on this already given preference list is average precision where you try finding out the average precision using the formula like this. What is this formula? Now, suppose these are the ranks given by you these are the ranks which are already given in the reference list and these are the items your rank and this is the reference. So, where all you had hits you had hit here, here and here. So, which means the item which was appearing in the first position you have shown it in the second position item which was appearing in the second position you have shown it in the third position which was appearing in the third position you have shown it in the fourth position.

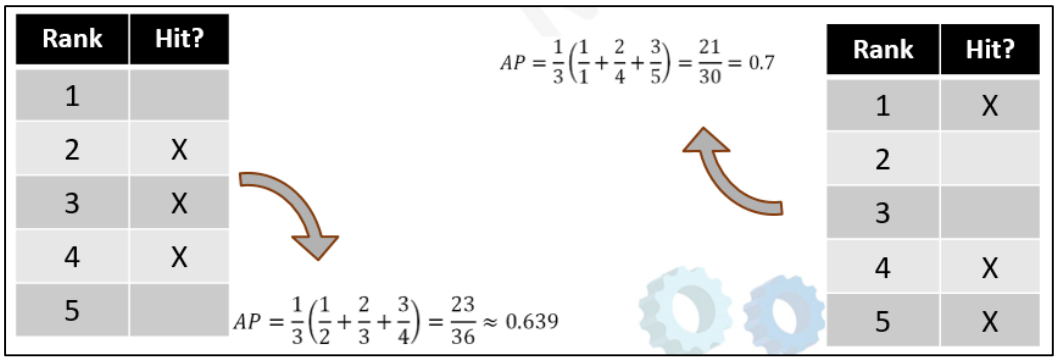


So, your average precision turns out to be this. Now, in this case the first one is correct in the second one is shown in the fourth position third one is shown in the fifth position. So, because this appears to be high compared to this because the first one itself is the hit. Now, sometimes there will be ties among the items. So, typically when there are tie among the ranks a good measure there are few good measures one is normalized distance based performance measure NDPM this is the first one second one is Spearman third one is Kendall. Spearman and Kendall correlation you already know and how do you compute them when you compute them you use both reference ranking and the ranking given by your algorithm and you compute these values c plus and c minus where they are concordant pairs discordant pairs and using this you calculate this cu and cs and once you calculate all these values you can compute the NDPM etcetera using this formula and these are the standard deviations of the actual ranks and your ranks and this is not the only formula there are other formula also we discussed in case of Spearman's correlation when we discussed about the similarity discuss about them in the context of similarity you can refer to that lecture.

Now, coming to measuring the accuracy of the ranking items in a utility based setting it is assumed that the utility of a list of recommendation is additive which means you can take the sum of the utilities of individual recommendation to find out the utility of the overall recommendation of for the algorithm. Now, utility of each recommendation is the

utility of recommended items discounted by a factor that depend on the position on the list. So, which means as you move ahead in the first position no discount second position. So, discount keeps on increasing more discounts are given on the items which are towards the end of the list.



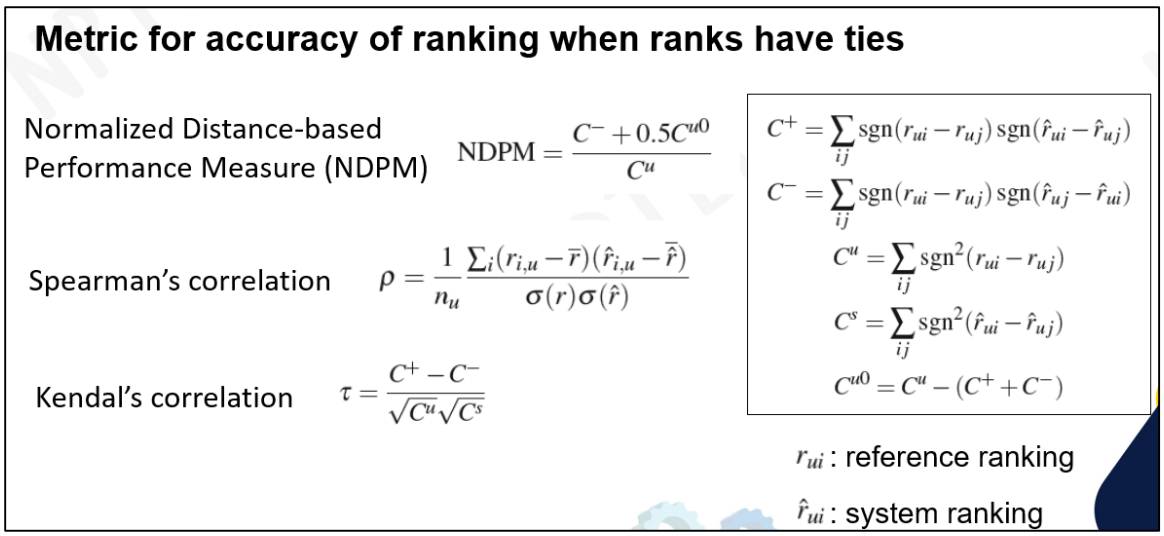


So, low to high. So, it is usually assumed that the users can recommended list from the beginning to the end. So, therefore, the high utility recommendation means the item is appearing towards the first and more penalized one with discounted appear towards the end. Now, these discounts can also be interpreted as probability that a user would observe a recommendation in a particular position in the list. So, under this interpretation the probability that a particular position in the recommendation list is observed is assumed to be assumed to depend only on the position and not on the item recommended. So, let us see two of such measures one is the R score metric it assumes that the value of recommendation declines exponentially down the ranked list to yield a R score.

$$R_u = \sum_u \sum_j \frac{\max(r_{ui_j} - d, 0)}{2^{\frac{j-1}{\alpha - 1}}}$$

So, this is the R score for user. So, where over all the items that he has ranked he has rated it is in the jth position this is the position this is the list of items rated by user u. So, over all the items that he has rated depending on where which position it occupies it will be now given a score. Now, let us say R u i is the actual rating of that item, but in the rank it has there in jth position. So, from there you subtract D what is D? D is the task dependent a task dependent neutral rating. So, which means let us say remember one

Likert scale the mid value where the preference is higher here and preference is lower here and you are neutral here.



So, from here suppose this rating would have been 5 you will be subtracting 3 from here. So, maximum of 5 minus 3 and 0 you will be taking. So, which means in case you have a lower value this part is going to be negative. So, you will be choosing 0 and that is divided by 2 to the power j minus 1 divided by alpha minus 1 where alpha is the called the half life parameter which controls the exponential decline depending on the value of the position in the rank list. Next you have normalized cumulative discounted gain NDCG it is also another measure for utility based ranking.

$$\text{DCG} = \frac{1}{N} \sum_{u=1}^{N} \sum_{j=1}^{J} \frac{g_{ui_j}}{\max(1, \log_b j)}$$

$$\text{NDCG} = \frac{\text{DCG}}{\text{DCG}^*} \quad \text{where DCG}^* \text{ is the ideal DCG}$$

So, this is actually to find out NDCG you have to find out this DCG values for the list of j items. So, now, for each user has a gain of GUI over being recommended an item I the average discount discounted cumulative gain DCG for a list of j item is defined as the sum of all the users all the end users and the items in the ranked list using this formula. Look at this log this log can take any base between 2 to 10 usually a logarithmic base 2 is commonly used to ensure all positions are discounted. Now, NDCG is a normalized version of this DCG and where this DCG star is the ideal DCG that you are expecting to have. So, these are the references I have used for this and these are my conclusions.

Now, in the context of recommender system we can assess the accuracy of rating prediction accuracy of users prediction and the accuracy of ranking of the item. Now, for

the first one that is assessing the rating prediction there are two major matrix RMSE and mean absolute error. The matrix derived from the confusion matrix are used for assessing the accuracy of users prediction and two import three important matrix which are very frequently used as precision recall and F1 score F score. Now, this rank matrix are different from the other two in the sense they consider the correct position of the item in the ranked list. So, there are two approaches for measuring this reference based using reference based ranking where already a reference list available to you.

Second one is utility based ranking which does not depend on reference ranking and in this context under each we saw there are many measures. So, here your NDCG here with reference rank ranking you had the many measures like that of us correlation based measures. So, you had here correlation based measures. So, with this we wind up today's lecture. Thank you.