Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 03
Lecture - 13

Lecture 13: User-Based Approach

Hello everyone. We continue our discussion on neighborhood based collaborative filtering. And in this context, we first talked about distance and similarity measures which are a very important component of this particular approach. Now, when it comes to collaborative filtering in the neighborhood, in the context of neighborhood based approach, we have two variants user-user based and item-item based. In this lecture, we are going to look at user-user based collaborative filtering. So, in both UBCF and IBCF, UBCF stands for user based collaborative filtering and IBCF is item based collaborative filtering.

So, both in UBCF and IBCF, the rating matrix is the only input. And both these are both these approaches are similar in the sense, the steps are almost same and UBCF utilizes user-user based similarity while predicting the rating whereas, IBCF utilizes item-item based similarity while predicting the rating. So, because rating matrix is a very has very unique characteristics and in both the methods, this rating matrix is the input. Let us just have a look at the characteristics of this rating matrix.

So, each user here explicitly in the context of recommender system, each user explicitly rates few items. Hence, the rating matrix is very sparse. This we have already discussed. This rating matrix can have different kinds of can have values from different kinds of measurement scales. There are certain continuous rating, but mostly the rating would be some discrete value.

When it comes to discrete value, either it is a ordinal like 0, 1, like 1, 2, 3, 4, 5, the Likert scale or 1, 2, 3, 4, 5, 7 like when the person starting from one person completely dislikes to or strongly dislikes to strongly likes. So, similarly there are unary scales. Unary is can be you person only likes, but 0 is actually insignificant here. In the binary scale both 0 means likes, 1 means likes and 0 may be dislike. So, this rating matrix can be explicitly obtained from the user or it can be implicitly created by the software where looking at the observation of the making the observation how the users are behaving.

Or it may so happen that you may be able to combine both explicit and implicit rating to generate some kind of hybrid rating. But whatever may be the case, only very few items are going to be rated frequently and very less item less number of items are frequent are very infrequently rated. So, as a result if we try arranging the data in terms of the items and their frequency we have these long tail phenomena. This we have already discussed. So, because of these long tail phenomena there are many issues.

What are these implications of long tail? In a neighborhood based approach most of the time if you are not little careful the neighborhood will be decided based on frequently rated item. So, as a result mostly the popular items will be with high frequency will be predicted. So, therefore the matrix such as coverage and diversity let me tell you we will be formally talking about various matrix to be used in a while evaluating recommender system. But coverage is something which tells you let us say in your item list you have total n number of items. So, what fraction of the item is your recommender system is typically predicting.

Normalization ⇨ $h(r_{ui}) = r_{ui} - \overline{r}_u.$ **Prediction** ⇨ $\hat{r}_{ui} = \overline{r}_u + \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv}\,(r_{vi} - \overline{r}_v)}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}$

Z-score normalization: While mean-centering removes the offsets caused by the different perceptions of an average rating, *Z*-score normalization also considers the spread in the individual rating scales.

Normalization ⇨ $h(r_{ui}) = \frac{r_{ui} - \overline{r}_u}{\sigma_u}.$ **Prediction** ⇨ $\hat{r}_{ui} = \overline{r}_u + \sigma_u \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv}\,(r_{vi} - \overline{r}_v)/\sigma_v}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}$

So, that makes the coverage and diversity. What varieties combinations of item it is predicting. So, both of these are going to be impacted because of the existence of this long tail. So, in order to make the algorithms more versatile in terms of coverage and diversity we have to adopt certain mechanism. So, that that items which are towards the longer tail can be accounted for.

Besides in an online setting if an item is has high frequency basically it is of low profit margin whereas, the merchants may be interested in the items with high profit margin. So, if those items are not recommended through the recommender system it is not advantageous for the merchants. Of course, the number of items sold altogether might be more, but high valued items will be sold very less. So, with this backdrop let us look at what are the steps involved in user based collaborative filtering. So, it has four stages.

Upon this second stage is not discussed in most of the literature most of the books that you find, but this is also an important phase. So, the phases are data preparation where you take care of the removing the biases in the data and you try normalizing it. Then you reduce the dimension in case the you would like to you are you are interested in latent feature of the user or item you try bringing the preference matrix to a lower dimensional latent space. And here through this you can also conveniently deal with sparsity and scalability problem. Then comes your neighborhood formation.

- Cosine based $sim(i, j) = \frac{\sum_{u \in U'} r_{u,i} * r_{u,j}}{\sqrt{\sum_{u \in U'} r_{u,i}^2} * \sqrt{\sum_{u \in U'} r_{u,j}^2}}$
- Pearson correlation $sim(i, j) = \frac{\sum_{u \in U'} (r_{u,i} - \overline{r_i})(r_{u,j} - \overline{r_j})}{\sqrt{\sum_{u \in U'} (r_{u,i} - \overline{r_i})^2} * \sqrt{\sum_{u \in U'} (r_{u,j} - \overline{r_j})^2}}$
- Adjusted Cosine $sim(i, j) = \frac{\sum_{u \in U'} (r_{u,i} - \overline{r_u})(r_{u,j} - \overline{r_u})}{\sqrt{\sum_{u \in U'} (r_{u,i} - \overline{r_u})^2} * \sqrt{\sum_{u \in U'} (r_{u,j} - \overline{r_u})^2}}$
- Binary based $sim(i, j) = \frac{N_{ij}}{N_i * N_j}$

For an active user you have to compute the similarity between all the users and the active user to form a proximity based neighborhood or the similarity based neighborhood with a number of like minded users. And finally, based on this neighborhood you are supposed to generate recommendation. Now come to the first task. Moving ahead let us look at start karayum. Moving ahead let us look at the steps involved here.

So, first step is data preparation and where you are supposed to do some kind of normalization. Now this normalization can be done in two ways in this context. One is mean centering. Here the idea is to determine whether the rating is positive or negative by comparing it with the mean rating. And the second one is z score normalization where it is mean centered as well as you also divide it by the standard deviation.

Item-Id ⇒ User-Id ⇓	1	2	3	4	5	6	Mean Rating	Cosine$(i,3)$ (user-user)	Pearson$(i,3)$ (user-user)
1	7	6	7	4	5	4	5.5	0.956	0.894
2	6	7	?	4	3	4	4.8	0.981	0.939
3	?	3	3	1	1	?	2	1.0	1.0
4	1	2	2	3	3	4	2.5	0.789	-1.0
5	1	?	1	2	3	3	2	0.645	-0.817

$$\text{Cosine}(1,3) = \frac{6*3+7*3+4*1+5*1}{\sqrt{6^2+7^2+4^2+5^2} \cdot \sqrt{3^2+3^2+1^2+1^2}} = 0.956$$

$$\text{Pearson}(1,3) =$$

$$= \frac{(6-5.5)*(3-2)+(7-5.5)*(3-2)+(4-5.5)*(1-2)+(5-5.5)*(1-2)}{\sqrt{1.5^2+1.5^2+(-1.5)^2+(-0.5)^2} \cdot \sqrt{1^2+1^2+(-1)^2+(-1)^2}}$$

$$= 0.894$$

Now the first one and when here what I am trying to show is that suppose Rui is your rating user u is giving rating for item i. Then you have a rating matrix and this is your user u. This is your item i. There are other users and other items as well. So, while doing mean centering in case of user based collaborative filtering you will be getting.

$$\hat{r}_{ui} = \frac{\sum\limits_{v \in \mathcal{N}_i(u)} w_{uv}\, r_{vi}}{\sum\limits_{v \in \mathcal{N}_i(u)} |w_{uv}|}. \qquad \hat{r}_{ui} = h^{-1}\left(\frac{\sum\limits_{v \in \mathcal{N}_i(u)} w_{uv}\, h(r_{vi})}{\sum\limits_{v \in \mathcal{N}_i(u)} |w_{uv}|}\right)$$

So this is your Rui and if you take all the Rui values across all the items and sum this Rui for i equal to let us say 1, 2 if you have n number of item and take the average you get Ru bar. So this Ru bar is will be subtracted from this Rui in every case. There will be all the values. So this mean will be subtracted from all. Now because you are subtracting this mean from everything then while you do the prediction you have to do the reverse.

Item-Id ⇒ User-Id ⇓	1	2	3	4	5	6	Mean Rating	Cosine$(i,3)$ (user-user)	Pearson$(i,3)$ (user-user)
1	7	6	7	4	5	4	5.5	0.956	0.894
2	6	7	?	4	3	4	4.8	0.981	0.939
3	?	3	3	1	1	?	2	1.0	1.0
4	1	2	2	3	3	4	2.5	0.789	-1.0
5	1	?	1	2	3	3	2	0.645	-0.817

Using Pearson correlation based similarity with raw rating

$$\hat{r}_{31} = \frac{7 * 0.894 + 6 * 0.939}{0.894 + 0.939} \approx 6.49$$

$$\hat{r}_{36} = \frac{4 * 0.894 + 4 * 0.939}{0.894 + 0.939} = 4$$

Using Pearson correlation based similarity with mean centered data

$$\hat{r}_{31} = 2 + \frac{1.5 * 0.894 + 1.2 * 0.939}{0.894 + 0.939} \approx 3.35$$

$$\hat{r}_{36} = 2 + \frac{-1.5 * 0.894 - 0.8 * 0.939}{0.894 + 0.939} \approx 0.86$$

Reverse in the sense when you do prediction using the similarity of the other users this Wuv is the similarity with the user v in the neighborhood of user u where you take the weighted average. About this we are going to discuss shortly. And we take all these positive weights and add them up to take the average value. This average value has to be now added to this mean value because you have already mean centered it. Suppose you do not mean center it, it is also possible that you do not mean center it.

So if you do not mean center it then you do not have to add this part. Your prediction can be done through this only. So whatever transformation you do at the time of data preparation that has to be taken care of at the time of prediction as well. So in case of z score normalization same thing this is the mean and there will be something called standard deviation. Standard deviation formula we have already discussed.

$$v_{ir} = \sum_{v \in \mathcal{N}_i(u)} \delta(r_{vi} = r)\, w_{uv} \qquad \hat{r}_{ui} = h^{-1}\left(\underset{r \in \mathcal{S}'}{\arg\max} \sum_{v \in \mathcal{N}_i(u)} \delta(h(r_{vi}) = r)\, w_{uv} \right)$$

This is the data value minus mean square of that sum them up and take the root over and 1 upon n minus 1 because it is a sample. Now, once you do this at the time of again rating prediction instead of actual rating a mean centered value you will be multiplying with similarity. This is your similarity. This will be done across all the users which who are there in the neighborhood. So when it talks to the similarity measurement methods, all the similarity measurement and distance measurement methods that you have that we have gone through are applied depending on the scale of the data the measurement scale of the data.

- User 1 and 3 are most similar (considering top two neighbors)
- The predicted rating is

$$\frac{\text{user 1's similarity*rating + user 3's similarity*rating}}{\text{Sum of similarity weights}}$$

=(0.85*3+0.7*4)/(0.85+0.7)=5.35/1.55=3.45

- $V_{item5,rating3}$ =1, $V_{item5,rating4}$ =1 (any one can be selected)

	I1	I2	I3	I4	I5	
UA	5	3	4	4	?	
U1	3	1	2	3	3	sim = 0.85
U2	4	3	4	3	5	sim = 0.00
U3	3	3	1	5	4	sim = 0.70
U4	1	5	5	2	1	sim = -0.79

However, in case of neighborhood based similarity mostly cosine based, Pearson correlation and adjusted cosine these are used predominantly and in case the data is binary we use the binary measures that we have already discussed. Now let us have a look at the similarity computation. So when we tried why do you need similarity computation? Because based on these similarity values you will be predicting. So how many users we have? We have 5 users and 6 items. Let us say our target user is user 3.

So with user 3 you have to compute the similarity with other users to find out the neighborhood. Now when you take the similarity with respect to other users you may make the data mean centered you may not make the data mean centered. In case of cosine similarity suppose you are not as such you do not use mean centering. So individual elements like you if you are taking the that is the cosine similarity between 1 and 3 who are the users? This is user 1, this is user 3. You are supposed to compute the similarity.

Now if you look at user 1, user 1 has rated for all the items. But when it comes to user 3, user 3 has rated only 4 items. So therefore while computing the similarity using any of the methods it is a basic requirement that both the vectors are supposed to be of same length. So here what are the vectors? For the user 1 it is 7, 6, 7, 4, 5, 4. And in case of user 3 it is no rating is given then 3, 3, 1, 1 here there is no rating.

So we are supposed to predict these values this place and this place we are supposed to predict. So for that prediction we need to find out the similarity between u1 and u2 u1 and u3 how do I find similarity based on correlated items. So who are the correlated items here? These four are the correlated items. So while comparing user and user 3 our vectors will be these two. These two not because they are not correlated.

	I1	I2	I3	I4	I5
UA	5	3	4	4	?
U1	3	1	2	3	3
U2	4	3	4	3	5
U3	3	3	1	5	4
U4	1	5	5	2	1

sim = 0.85

sim = 0.00

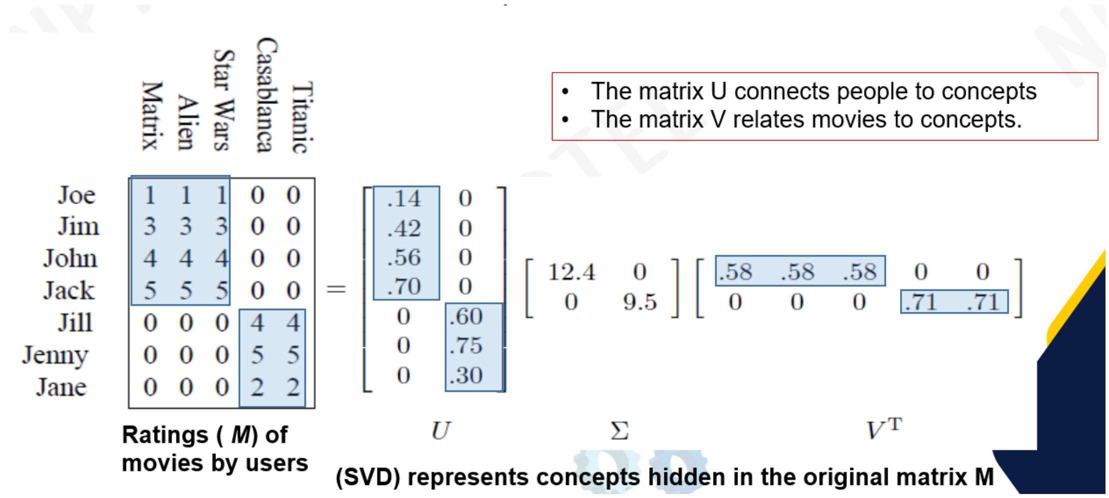
sim = 0.70

sim = -0.79

Similarly consider user because we are trying to find out the similarity between users then user 3 and user 2. So when we consider user 3 and user 2 the vectors are 6, 7, no value 4, 3, 4 and this one is again no value 3, 3, 1, 1 no value. So this is your user 2 this is user 3. So how do you take? What is the vector now for comparison? The values exist here these are correlated here as well as here. So the while comparing u2 and u3 u2 will be 7, 4, 3 and u3 will be 3, 3, no 3, 1, 1.

Whereas look at this here both of them represented by a 4 dimensional thing and here they are represented by 2 dimensional things. So this is for convenience for convenience we have to consider only correlated items. So this is how all these values with respect to user 3 are similarity values are created. Now look at this. This is we used cosine similarity here Pearson and in Pearson some of the values are negative because they are not positively related.

Now out of all the neighbors that you find in this small example there were how many neighbors there were for a user 3 there were 4 neighbors. But in case of a very large data set there will be hundreds of neighbors. So in you have very large number of neighbors then you have to decide which neighbors to select while making the prediction. So this is done with 3 approaches top k filtering, threshold filtering and negative filtering. In case of top k filtering each user or item for each user or item only a list of n nearest neighbors and their respective similarity weights are kept.



So in this context of this particular example if this is the similarity who are the top 2 neighbors? Wherever the similarity values are high so this is the first neighbor this is the second neighbor first neighbor second neighbor. The values may be different in both the similarity computation but in this context both these neighbors are same. Of course here also we have I mean this is itself so correlation the value similarity value is 1. Next is your threshold filtering. In case of threshold filtering instead of fixing the number of nearest neighbors you keep all the neighbors whose similarity weight has magnitude higher than certain threshold.

For example in this setting if we decide that we will be keeping all the neighbors who give the rating above 0.5 then who will be the neighbors? Neighbors will be all 4 and if we keep the setting that it should be higher than 0.8 let us say then who will be the neighbors? The top 2 will be the neighbors. If we put it 0.

6 top 3 will be the neighbors and so on. So this is your top k filtering. Then this is threshold filtering. Then you have negative filtering. What is negative filtering? As intuition says if two users give are negatively related which means they are the way they are giving giving giving the rating are just the opposite. So therefore, you may think of completely neglecting those users whose ratings are having negative correlation.

So it is quite intuitive it may so happen that if they are negatively related probably they belong to completely two different groups and if they are positively related they become to a common group. So in the context of our last example who are if we do negative filtering of course you here you cannot apply because all the values are positive in case of cosine but here these two will be automatically dropped because they are negatively related. So therefore, only top 2 will be considered. So after you find out the rating sorry find out the similarity values how do you get the similarity values? By comparing the comparing, it with other users and after comparing with the other users you compute a similarity matrix which is a if you have n number of users you have n cross n similarity matrix. So comparing all the u v u v u v I mean all the users who might be there might be many more users.

$$
\begin{array}{c|c}
 & \begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic}\end{array} \\
\begin{array}{c}\text{Joe}\\ \text{Jim}\\ \text{John}\\ \text{Jack}\\ \text{Jill}\\ \text{Jenny}\\ \text{Jane}\end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0\\
3 & 3 & 3 & 0 & 0\\
4 & 4 & 4 & 0 & 0\\
5 & 5 & 5 & 0 & 0\\
0 & 2 & 0 & 4 & 4\\
0 & 0 & 0 & 5 & 5\\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
.13 & .02 & -.01\\
.41 & .07 & -.03\\
.55 & .09 & -.04\\
.68 & .11 & -.05\\
.15 & -.59 & .65\\
.07 & -.73 & -.67\\
.07 & -.29 & .32
\end{bmatrix}
\begin{bmatrix}
12.4 & 0 & 0\\
0 & 9.5 & 0\\
0 & 0 & 1.3
\end{bmatrix}
\begin{bmatrix}
.56 & .59 & .56 & .09 & .09\\
.12 & -.02 & .12 & -.69 & -.69\\
.40 & -.80 & .40 & .09 & .09
\end{bmatrix}
$$

$$
\text{Ratings } (M) \qquad U \qquad \Sigma \qquad V^{T}
$$

$$
\begin{bmatrix}
.13 & .02\\
.41 & .07\\
.55 & .09\\
.68 & .11\\
.15 & -.59\\
.07 & -.73\\
.07 & -.29
\end{bmatrix}
\begin{bmatrix}
12.4 & 0\\
0 & 9.5
\end{bmatrix}
\begin{bmatrix}
.56 & .59 & .56 & .09 & .09\\
.12 & -.02 & .12 & -.69 & -.69
\end{bmatrix}
=
\begin{bmatrix}
0.93 & 0.95 & 0.93 & .014 & .014\\
2.93 & 2.99 & 2.93 & .000 & .000\\
3.92 & 4.01 & 3.92 & .026 & .026\\
4.84 & 4.96 & 4.84 & .040 & .040\\
0.37 & 1.21 & 0.37 & 4.04 & 4.04\\
0.35 & 0.65 & 0.35 & 4.87 & 4.87\\
0.16 & 0.57 & 0.16 & 1.98 & 1.98
\end{bmatrix}
$$

$$
\text{Ratings } (M')
$$

So comparing all of them you compute a similarity matrix. So if you have n number of users your u your u w matrix which you are you are considering as your similarity matrix w is the similarity matrix so this will have n cross n values. And in fact because it is a similarity matrix it will be symmetric. So only it is a diagonal matrix with similarity values. But whatever may be the case similarity you have to take the weighted average multiplying similarity with the rating of the second user and taking the average by dividing it with the positive mod of all the values.

$$
\begin{bmatrix}
5 & 2 & 4 & 4 & 3\\
3 & 1 & 2 & 4 & 1\\
2 & & 3 & 1 & 4\\
2 & 5 & 4 & 3 & 5\\
4 & 4 & 5 & 4 &
\end{bmatrix}
=
\begin{bmatrix}
u_{11} & u_{12}\\
u_{21} & u_{22}\\
u_{31} & u_{32}\\
u_{41} & u_{42}\\
u_{51} & u_{52}
\end{bmatrix}
\times
\begin{bmatrix}
v_{11} & v_{12} & v_{13} & v_{14} & v_{15}\\
v_{21} & v_{22} & v_{23} & v_{24} & v_{25}
\end{bmatrix}
$$

Now if at all you apply any kind of normalization to the data corresponding normalization has to be taken care of. So as we have seen min if you have subtracted min, min has to be added and here if you have used min centering min has to be subtracted. If you have used z score, then it has to be divided by the standard deviation and so on. So this is the example. So this example we have already seen and in this example we found out the cosine similarity and Pearson similarity with respect to for user 3 with respect to rest of the users.

$$w_j = \log\left(\frac{m}{m_j}\right) \quad \forall j \in \{1 \ldots n\}$$

$$\text{Pearson}(u, v) = \frac{\sum_{k \in I_u \cap I_v} w_k \cdot (r_{uk} - \mu_u) \cdot (r_{vk} - \mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v} w_k \cdot (r_{uk} - \mu_u)^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} w_k \cdot (r_{vk} - \mu_v)^2}}$$

So following the formula which was here in the first case if there is no min centering we can find it out in this manner. But in case of second case because we were actually subtracting the min value from everywhere min has to be now added. The min got added. So whose min? Min of user 3. What is the mean of user 3? He has given rating for 4 items, 2 items 3, 3, 1 item, 2 items 1, 1.

So 3 plus 3 plus 1 plus 1 divided by 4. So 6 plus 2, 8 by 4 that makes it 2. Now comes a different way of predicting the rating. So this is through a voting mechanism. You look at this in both these cases your rating was not in the scale in which it was the data is there. What was the scale in the data? In what scale the data was there? It was in a ordinal scale.

The ratings were given the values between 1 to 7. People were giving ratings between 1 to 7. Now you are supposed to put them in exactly this scale. In that case, voting is a very good option. So in case of voting what you do? In case of voting you figure out for each observation, each type of rating what is the vote. So this is one example of, this is one example in which let us say with this first user, this first active user these are the similarity values we have computed.

And based on this we have now predicted the rating. So when we predict the rating it is no more in this 1 to 5 scale. So what do you do? Now look at this item 5 rating 3, item 5 rating 4. What are we predicting? Item 5 and item 4. So this value is 3.54. So this 3.54 whether it will be 3 or whether it will be 4. So now with rating 3 we have one observation, with rating 4 we have one observation. So frequency of rating 3 is also 3, frequency of rating 4 is also, sorry frequency of rating 3 is 1 and frequency of rating 4 is also 1. So therefore it will be, you can choose any one.

So this can be approximated by either 4 or 3. So this is one way to discretize if we have this weighted average scheme. But in case you do not go for this weighted average then we can, looking at the similarity we can find out a vote for individual items. So here individual rating. So here what is the scale? 1, 2, 3, 4, 5. So this is the scale. So in the neighborhood how many times 1 has occurred? 1. How many times 2 has occurred? 0. 3 has occurred? 4 has occurred? So 1, 3, 4, 5 have vote 1 each and 2 has 0 vote. So now all of them 1, 3, 4 and 5 are candidates to fill this place.

Because all of them had got same vote. Now next thing you have to look at is your similarity value. Out of all these elements which are voted the highest value of similarity is this. So you multiply this with their corresponding similarity values.

So for 3 it is 0.85. For 5 it is okay. So who has given these 3? 3 is given by this user with whom the similarity is this. This rating 5 is given by this. So similarity this. So with all these similarities you will be multiplying. In case more people have given rating 5 you would have multiplied with similarity and would have taken the average. So now because there are only ratings given by only one user you do not have to take the average as well. So wherever the said rating is highest so that becomes the rating of this. So what is the predicted rating of this? Predicted rating of this is now 3.

For this 3 this is the highest. So predicted rating is 3. Okay so now in user based collaborative filtering the dimensionality reduction and discovery of latent feature also is very important. So I think I will be stopping here and directly going. So few slides I will be adjusting in next lecture. So in case of user based collaborative filtering we also have to take care of the long tail phenomena. So in case of long tail phenomena we have to make some kind of adjustment to the similarity matrix.

For example, here we can use the notion of inverse user frequency. In this case and if mj is the number of ratings of item j and m is the total number of users then you have to give a weight wj to the item by taking the log of m divided by mj and this you have to multiply while computing the similarity. For example, in Pearson's this is getting multiplied. So with this process effect of long tail phenomena can be taken care of up to some extent. So in this lecture we have covered two very important books. One is Recommender System by Charva Grewal and Recommender System Handbook and these two books I have also prescribed you as the textbook for this course.

So this is my concluding demand. User based collaborative filtering predicts unknown ratings for an user. So now we conclude the lecture. The user based collaborative filtering predicts the unknown ratings for an item for a specific user based on the ratings of the same item by the like minded users. This like mindedness is found out by computing the similarity. The steps involved in UBCFs are data preparation, dimensionality reduction, neighborhood formation and recommendation generation.

And we also discussed how to take care of the long tail phenomena. And the and because of this we are supposed to for the items which are at the fag end of the tail we are supposed to give them certain higher weight and accommodate it while computing the similarity. Thank you.