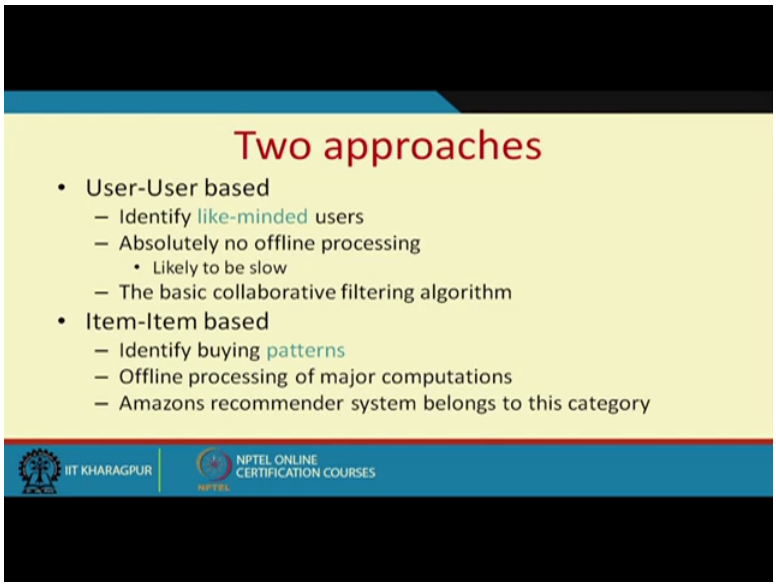
E Business
Professor Mamata Jenamani
Department of Industrial and Systems Engineering
Indian Institute of Technology Kharagpur
Lecture-54
Collaborative Filtering Based Recommender System

Welcome back, we continue our discussion with recommender system and in this particular class we are going talk about collaborative filtering based recommendation system.

(Refer Slide Time: 0:40)

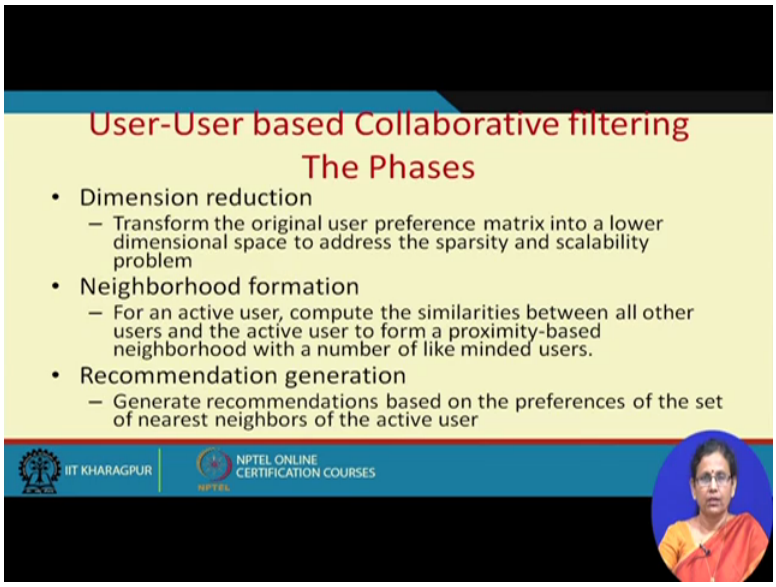


Here this collaborative filtering based recommender system can be broadly classified into 2 categories; user-user based and item-item based. In fact, there are other collaborative filtering techniques as well but we are limiting ourselves to only these 2 to understand the concept. Now in this user-user based collaborative filtering, the idea is to identify the like-minded users and finding those like-minded users and looking at the preference of one user, you recommend the item to a similar user who have seen who are similar in terms of their in terms of the items they are looking at, so it is about identifying the like-minded users.

So this collaborative filtering algorithm is actually the older of these 2, in fact the social information filtering that we were talking about which was a very first initiative for recommendation generation, this user-user based finding the like-minded users for generating recommendation was used. So now in this particular method which is not very popular in the commercial domain because of its calculation procedure where everything happens in a online environment, as a result the system becomes very slow as the number of users and number of items increase and it is extremely difficult to use them in the commercial setting.

However, for research setting many people still also use this user-user based approach, so it is about identifying like-minded users and this is a basic collaborative filtering, but the corresponding commercial application is bit different where the item-item based on the buying pattern of various users, the item-item based similarity based on the users buying pattern is first computed off-line, then whenever a user comes and looks at an item then that matrix that similarity matrix is preferred to suggest the item. In fact, Amazon in a commercial setting first started this particular item-item based recommendation. So in case of whether it is either item-item or user-user, few steps are followed.

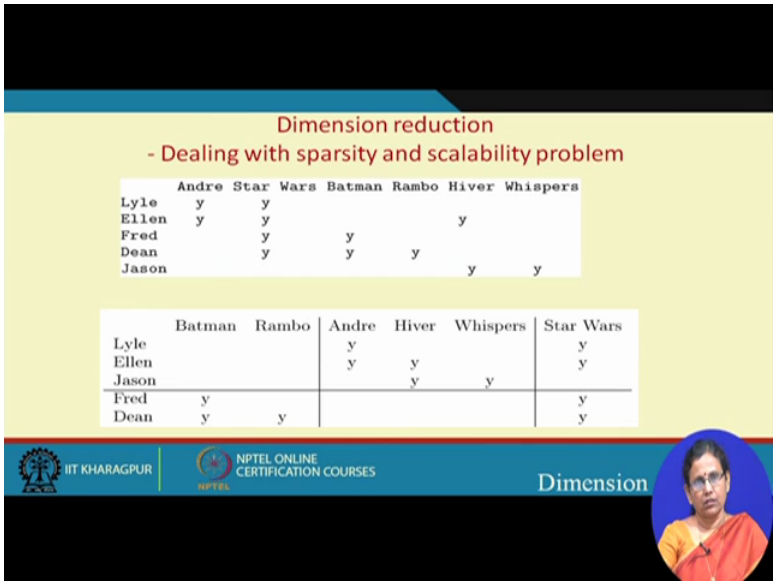
(Refer Slide Time: 3:37)



For example, in case of user-user first step is to reduce dimension. So it is about transforming the original user preference matrix into a lower dimension, user preference matrix if you remember for recommender system while discussing about in the very first-class about the basic framework, the important element in a recommender system was the user-user user-item preference matrix. This user-item preference matrix contains the ratings provided by each user to each item, when I said provided by either it is explicitly provided or it is implicitly generated, but whatever it may be the case we assume that we have the user item rating matrix with us okay.

Now, this matrix often will be very large so therefore transforming this preference matrix into all over dimensional matrix to address the sparsity and scalability problem is the first task to be addressed. Then the second is, for each user we have to find out who are the similar users or we will be forming the neighbourhood of each active user, then once we find out the neighbourhood who are the like-minded users then we generate the recommendation.

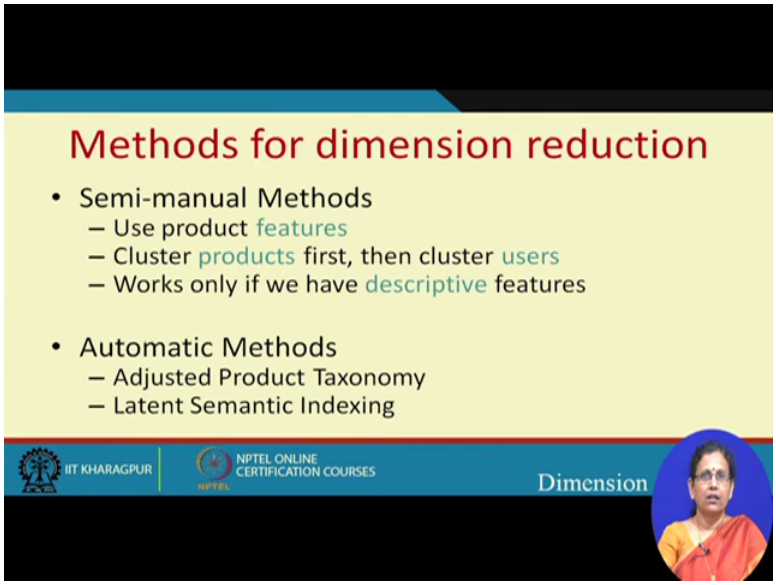
(Refer Slide Time: 5:20)



So for this dimensionality reduction may be single value decomposition or your PCA can be used. But here we are simply giving some very rudimentary example what exactly we mean by dimensionality reduction. Let us say we have some site users rating some 6 items, now looking at this ratings you can see not every user has seen every movie. So many of these spaces are blank, so this matrix is quite sparse, it is not that sparse but it is quite sparse. But as you can increase as you can imagine if the number of users grow, number of items grow and in each user might not be might not have seen all the movies so therefore only few movies each user must have seen so therefore as the size increases the sparseness will also increase, but the idea here is to see that how it can be grouped together in different groups.

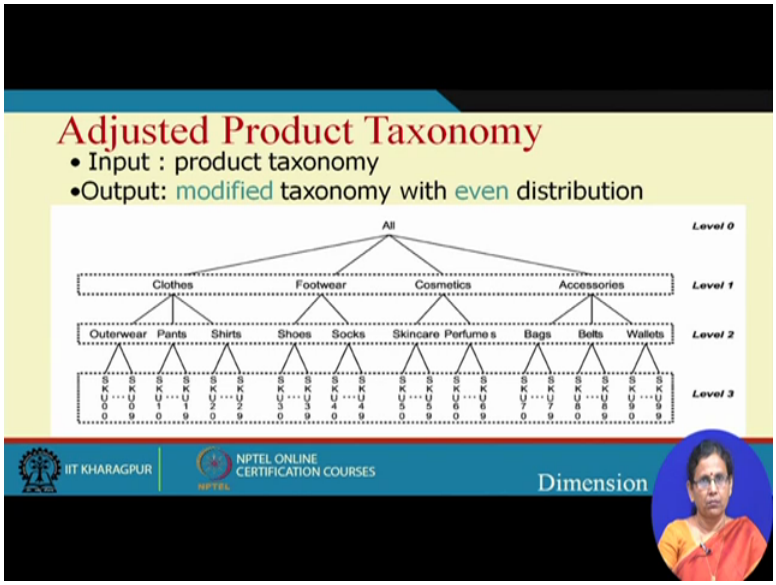
Look, now if we rearrange these rows possibly we can make some 3 groups like action, some outside foreign movie, some classic. And you can see the first group of users are interested in foreign movies and classic and second group are interested in action and classic movies. Now this kind of categorisation if it is done it can be done automatically, it can be but doing it manually is a cumbersome process but at least if we know the category of the movie from let us say some explicit dataset we can try this categorisation as well forgrouping the users.

(Refer Slide Time: 7:44)



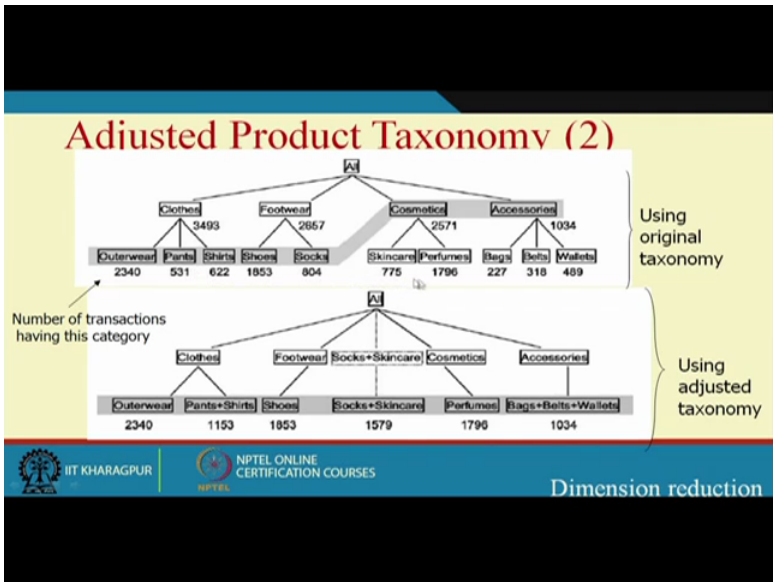
So these methods of dimensionality reduction can be either semi-manual or it can be automated. In case of semi-manual you can use products features like this movie example, whether the whether the movie belongs to a specific category you can try to do some kind of semi-manual activities. Then you can also cluster the products then you can cluster the users to find out the groups. So even if you have I mean only when you have descriptive features of the products available you can go this way, but if not you can actually go for some kind of automatic method, some of this method I mean that this method in general we are not going to discuss as such but some example situation we can see.

(Refer Slide Time: 8:45)



For example in case adjusted product taxonomy, look we have a dataset in which we have clothes, footwear, cosmetic and accessories. Manually this grouping is made and under each group against some subgroup is made and the number of items under each can be found out.

(Refer Slide Time: 9:10)



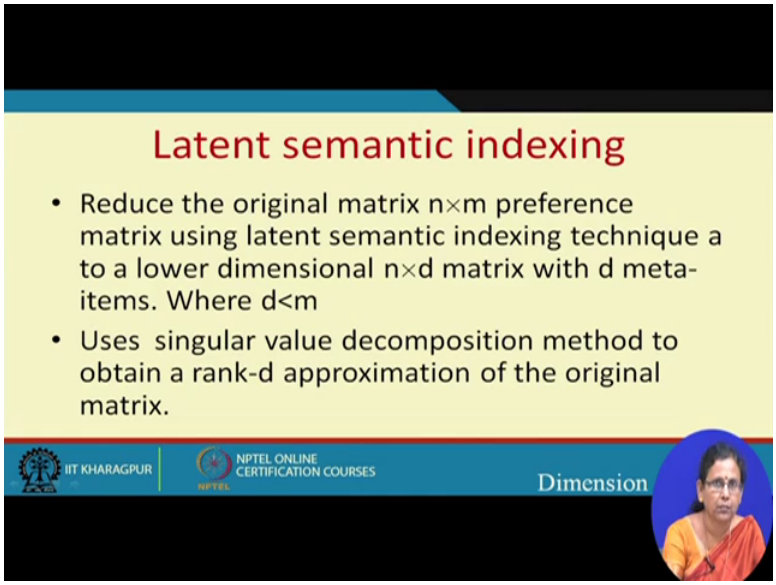
Let us say these are the number of items under each, but the thing is here there are more items, here there are less. So, even if I mean we will try to more or less balance this to for ease of search what can be done, some products can be moved even if they do not belong to this category they can be moved from here to here and make. For example, look clothes were these outerwear pant, shirt, et cetera, and footwear were shoes, socks and so on. Now, they can be adjusted like clothes can contain this, outerwear pant and shirt together because these 2 values were too less, they can be combined and put together.

Footwear can contain only shoes, then this socks and skin care can be put as one socks plus skin care category with these many products, cosmetics under cosmetic you can put only perfume because this value is very high you can put together and these numbers were very small these accessories, along with that I mean all these 3 things can be put together and made one thing. So instead of these many subcategories now we have broken it into a different subcategory though sometimes because no see this was required domain knowledge and user with domain knowledge has some developer with domain knowledge has manually divided this partition, manually partitioned this.

But if you do it automatically taking care of that syntactic things may not be possible but at least the program can take care of adjusting the products in each category so that items

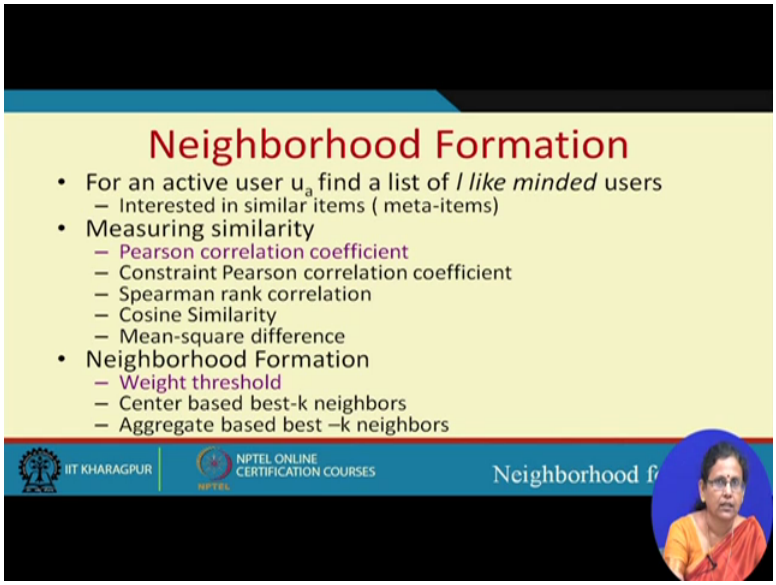
number of items are at least of same number so that they I mean the balance can be maintained by taking a search. So instead of these small-small things kept together they have been combined here, pant-shirt has been combined, then your skincare and socks has combined, perfume is considered as a separate thing and all the accessories has been combined, so that way search becomes little fast.

(Refer Slide Time: 11:52)



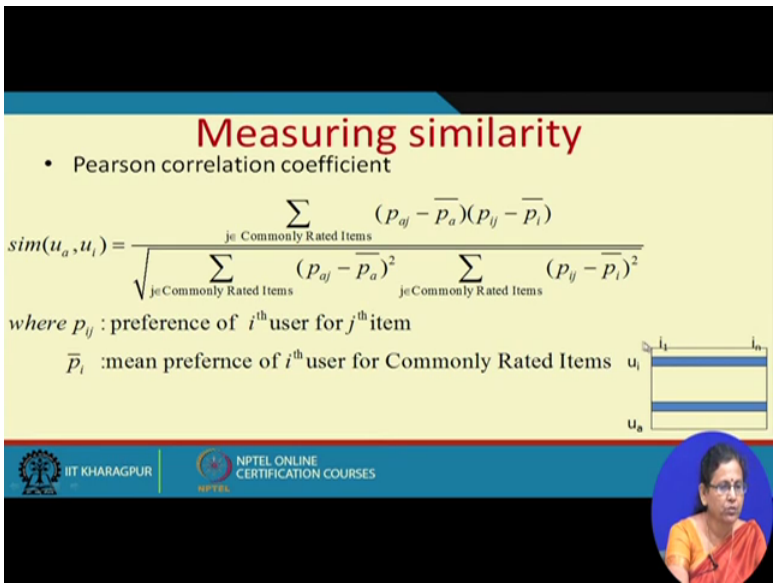
Then there can be some further automatic methods like your Latent semantic indexing, where the original matrix dimensions of the original matrix is reduced by finding the corresponding lower dimensional equivalent matrix okay. So it specifically it uses singular value decomposition method if you are acquainted with, but anyway you do not worry you do not have to be solving problems, this is just to let you know that such methods do exists.

(Refer Slide Time: 12:27)



Then next task was to, first was the dimensional reduction, next task was to find out the like-minded users, this is done by forming the neighbourhood. So for forming this neighbourhood again there are many methods, one such method is using the most simplistic method is using by the use of some correlation coefficient.

(Refer Slide Time: 13:00)



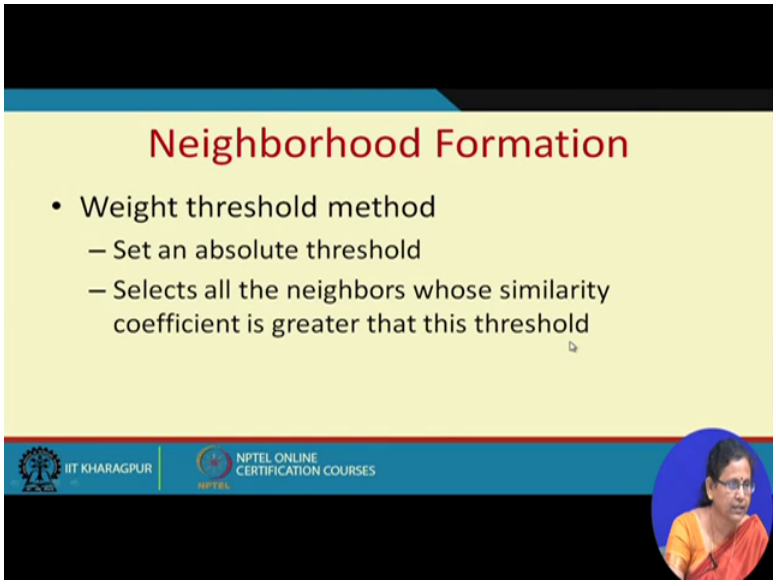
For example, if user u i and user u a, user u a is the active user whose whether he is similar to this user or not can be found out by computing the correlation between this row and this row. This row and this row, but see this person might have been rating many items, this person has already rated many items but all the items this person has rated this person has not rated. Similarly, all the items this person has rated this person has not rated so therefore will be to

find out coordination you need same number of elements in these 2 user vectors. So therefore, if something is rated here and not rated here and you are including, your dimension of the vector will not be same.

If you think of replacing that with 0 in fact, many of the authors do that you can very well do, if the item is not rated you can consider it as 0 but some of the authors view it as a because if the rating is from 1 to 5 let us say in the scale of 1 to 5 then 0 means giving a very lower rating, for computational purpose you can use that 0 but this may lead to certain erroneous result so therefore many of the authors suggest that you take you may win this vector is u i and u a for only co-rated items, once you have these co-rated items ready, both these will have same length.

So once they have the same length, you can use this formula and find out the correlation. Once I find out the correlation, this correlation will be treated as the similarity between the active user and some other user, some other user has already given his rating. So therefore, for this user a with all the remaining users you find out this similarity, once the similarity is computed then top few users who are most similar to this active user will be under his neighbourhood.

(Refer Slide Time: 15:34)

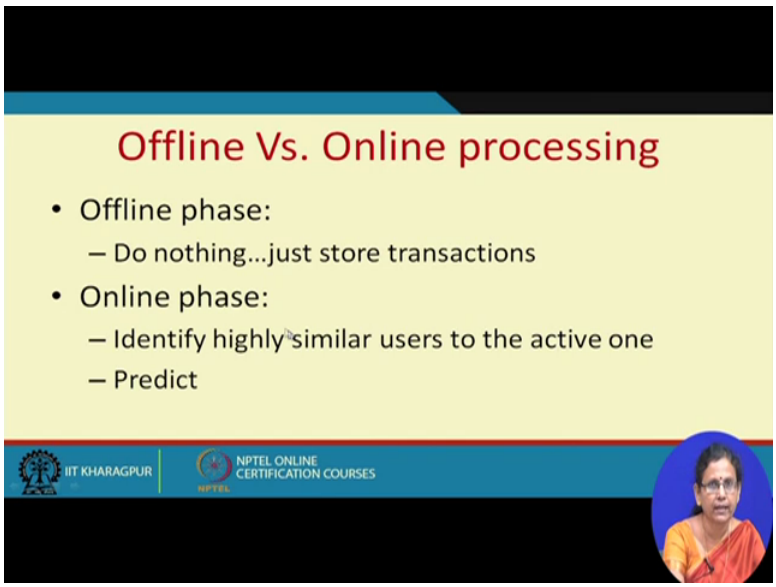


So once this neighbourhood is formed and you find out the nearest neighbours, next is generating the recommendation for a newer item, newer item you generate the recommendation. To generate this recommendation there are again see these formulas that I am showing, from author to author this formula vary little bit to adjust to a specific situation

or to incorporate a particular author's viewpoint. But for our application purpose here for our discussion purpose here please limit yourself to the formulas that we discuss, recommender system in a broad topic I have made it clear from the beginning and these ratings, et cetera, the calculation procedure et cetera may vary based on various authors opinions but here we use this.

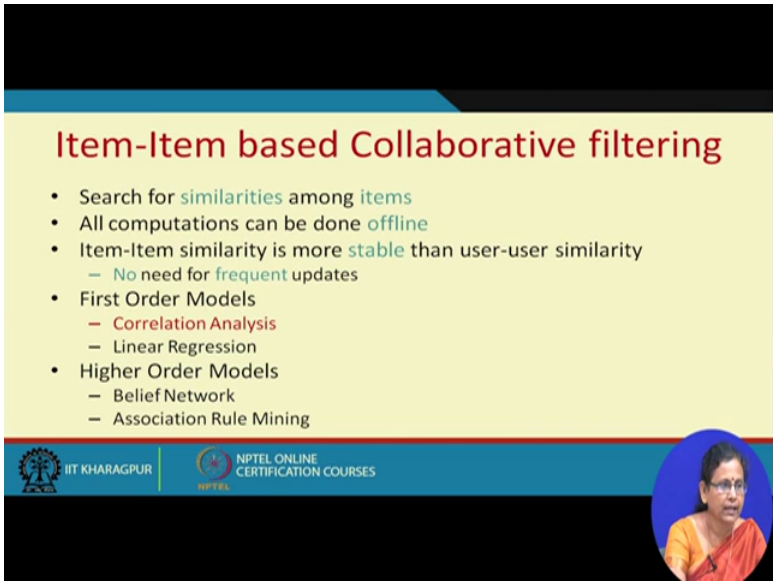
So to find out the predicted rating of the item j for the active user a, we take the average rating of the active user then the contribution from the similarity of the other user we add to this. So this is found out by the waited value of the deviation of each active user from that specific product rating, this is P ij is the rating given by the ith user for jth product. P i bar is the average rating given by the ith user so this this deviation from this means into similarity of user A with user i which you have already found out this weighted sum normalised to this sum of the similarity value is added to this average rating score of the user A to generate the predicted rating value. So this predicted rating for all the items which are not seen by user A are found and out of that the top few once are suggested to the user okay.

(Refer Slide Time: 18:33)



So here what is the off-line phase? Off-line phase there is nothing, just store the transactions of the past users, in online phase you find out the similarity of the active user with the other users then you predict. This is quite a time-consuming process, as the dataset group this is no more possible so therefore for commercial application this is not a right kind of scenario.

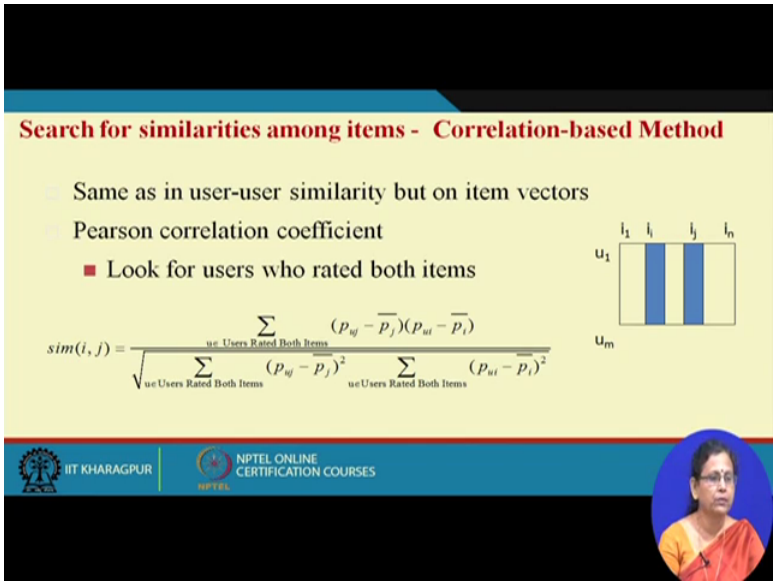
(Refer Slide Time: 18:59)



So next but this idea can be extended to a item-item collaborative filtering environment. Here the search for similarity among the item is first made and this all the computations for this generating this item-item similarity matrix based on the user preference as done in off-line manner. Then once this item-item similarity matrix is ready then it can be used for recommendation. By the way, this item-item similarity matrix is more stable, I mean they do not change very frequently like that of your user-user preference matrix because see in an online and environment you do not have really control over your users who is over user come today.

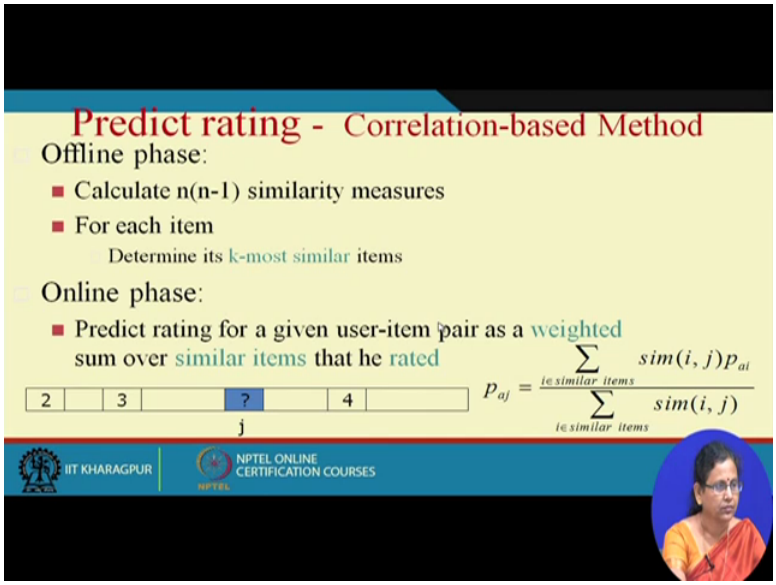
And moreover you do not keep track of your users because most of the users will not login into the site okay, so you can use cookies et cetera but that is again limited by the by the client computer by the client, client may decide not to use the cookies. But anyway, this item-item similarity is understood to be more stable, so there are many methods of doing so we are again going with that correlation base method which we have already discussed.

(Refer Slide Time: 20:27)



Here, instead of the user-user between item-item based on the co-rated users, if many users have commonly rated the items, wherever they have got ratings in both the places that part only is used for predicting the similarity score, the similarity score is nothing but your correlation coefficient. So after this is I am not going to discuss it further because we did it just now.

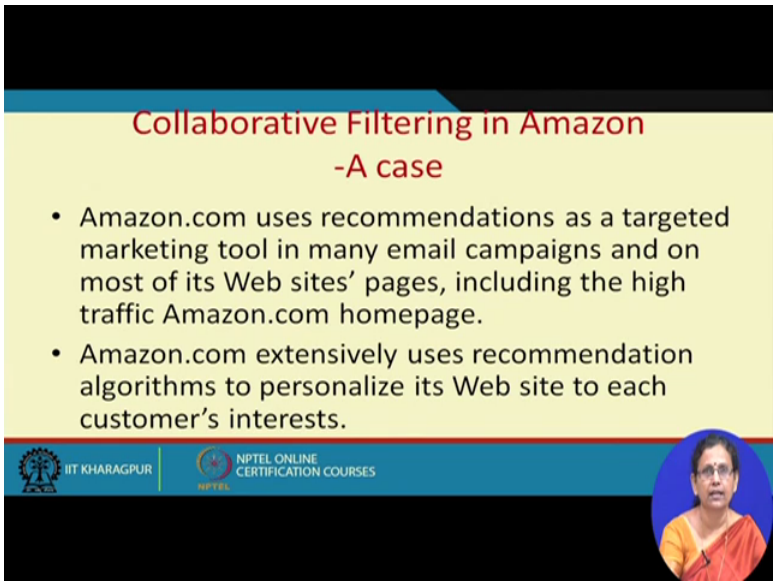
(Refer Slide Time: 21:02)



So once this is done you find out this item-item similarity, this is a n cross n matrix but because it is a n cross n symmetric matrix because the similarity of i to j is same as the similarity of j to i you have total n into n – 1 number of similarity measures from diagonal to upwards, diagonal matrix it is basically diagonal matrix. So for each case you determine what

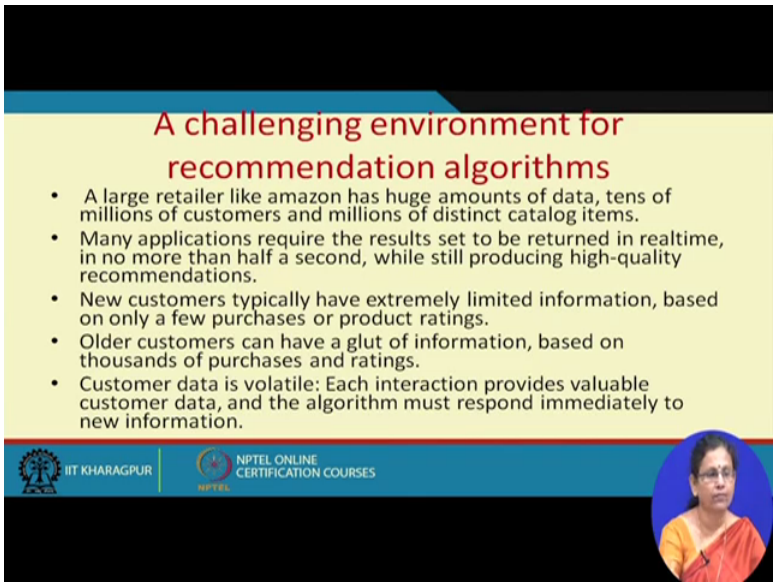
are the k similar items and you keep it together. Now in the online phase when a user comes, as soon as he puts one item into the shopping cart or as soon as he views one item, for that item whatsoever is the similar item you find out from this k most similar item list and display, okay.

(Refer Slide Time: 22:05)



So this is as I told you this particular method is adopted in fact suggested by Amazon, so Amazon uses this particular algorithm.

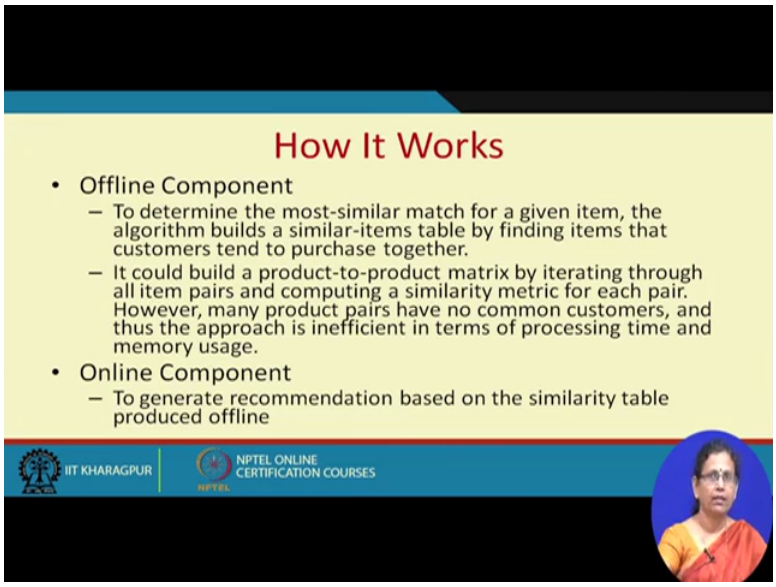
(Refer Slide Time: 22:23)



But a large retailer like that of Amazon has huge amount of data maybe tens of millions of customers and millions of distinct product items. Now many applications applications require

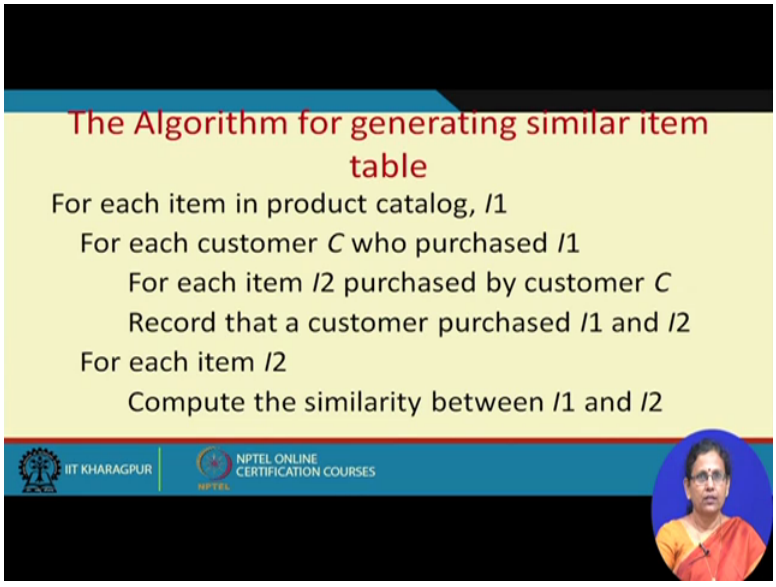
the results said to be written in real-time so therefore, maybe within few seconds you have to provide your suggestion. So in this deal to deal with this and environment they only suggested this item-item collaborative filtering algorithm.

(Refer Slide Time: 22:55)



In the off-line component, the computation of the item-item similarity matrix is made and in the online component as soon as user enters the item, the similarity is phased from the similarity table and items are suggested.

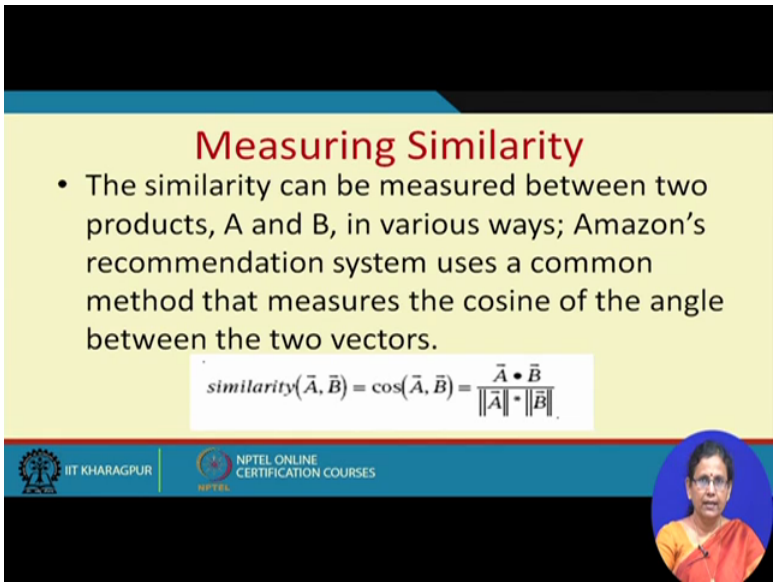
(Refer Slide Time: 23:21)



In this process we use the algorithm like for each item in the product catalogue specific product catalogue let us say I 1, for each customer who purchased this item I 1, for each item

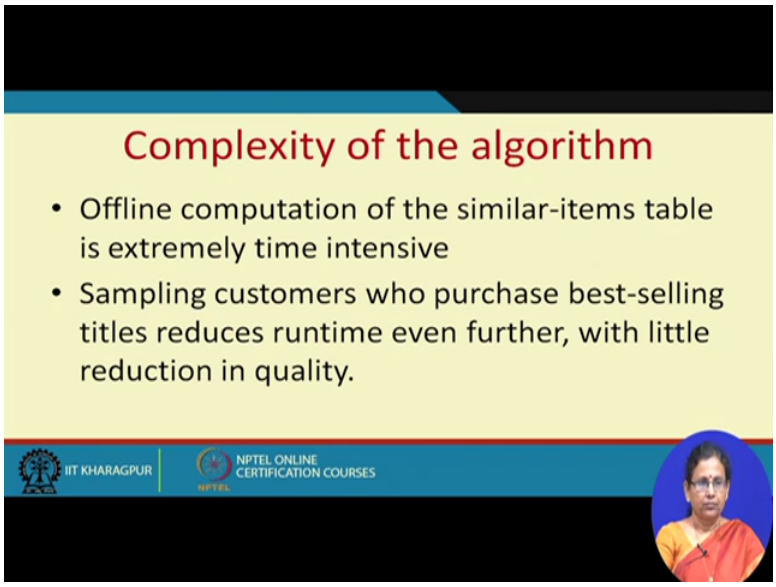
I 2 purchased by the customer C, record that the customer has purchased I 0 and I took, for each item I 2 compute the similarity between I 0 and I 2, this is their method for finding the similarity.

(Refer Slide Time: 23:56)



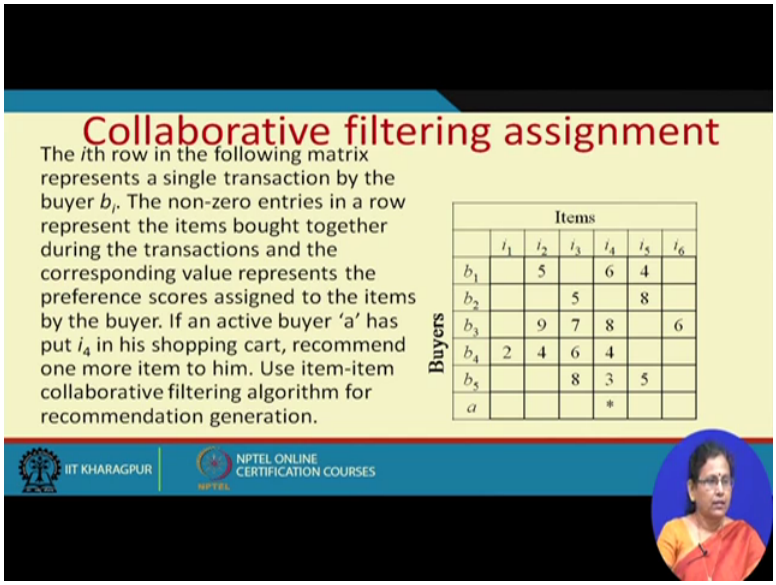
Next, after similarity once we find out this thing these 2 vectors where they are co-rated, Amazon specifically uses this cosine similarity. I have told you the similarity measures can be of many types, correlation was one but Amazon as such uses cosine similarity so there are many similarity measures.

(Refer Slide Time: 24:27)



So once this is done, the similarity id computed which is quite time-consuming, in the online environment the recommendation can be very weakly made which depends on I mean you do not really have to have a past history of the user because every user whenever he browses or purchases, their history is instantaneously captured that is kept, item-item similarity matrix is made. So whenever a new user comes in whether he registered or not registered does not matter, as soon have puts one item that item is the similar items to that item is phased from the similarity table and but back to him as recommendations.

(Refer Slide Time: 25:31)



So this is the assignment you try doing, here the ith row in this matrix represents the single transaction by a buyer B i. The nonzero entries in a row represents the items bought together during the transaction and the corresponding value represents the preference score assigned to the items by the buyer. Now if an active buyer A has put item i 4 in his shopping cart, you have to recommend one more item to him. You have to use item-item collaborative filtering algorithm for recommendation generation, use both the cosine similarity as well as Pearson similarity because you know both these to generate this recommendation.

So what you have to do, first you have to build an item-item similarity matrix, ones that item-item similarity matrix is made then you have to set, if an active user A has put an i 4 in his shopping cart, which other items have to be recommended to him. Look, there is a problem associated with collaborative filtering which is called cold start problem, we are not anyway going to discuss have a discussion on that . Cold start problem in the sense suppose a new item comes in, no user has rated the item so it similarity with the other items is not available then in all the recommendations it is never going to come so it has to be this particular

situation has to be dealt with separately, possibly using a content based recommender system recommendation approach you find out content wise how similar is this product with others and then you start.

And once you start suggesting it, people will view it, people will purchase it, item's history will be generated then it can have a place in the item-item similarity matrix okay. So with this we finish our lecture and next lecture we will be continuing on this recommender system with a new type of approach, thank you very much.