Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 01
Lecture - 03

Lecture 3: Types of recommender system-II

Hello everyone. We are now ready to continue with our lecture on Types of Recommender System. So, basically once again we are going to continue on various types of recommender systems. And the second type the already we have covered talked about the popularity based recommender. And in case of popularity based recommender recommendation system, basically the kind of data that I was discussing what was the data like? The data was in the form of three matrices. This was preference matrix and here we had users and here we had items.

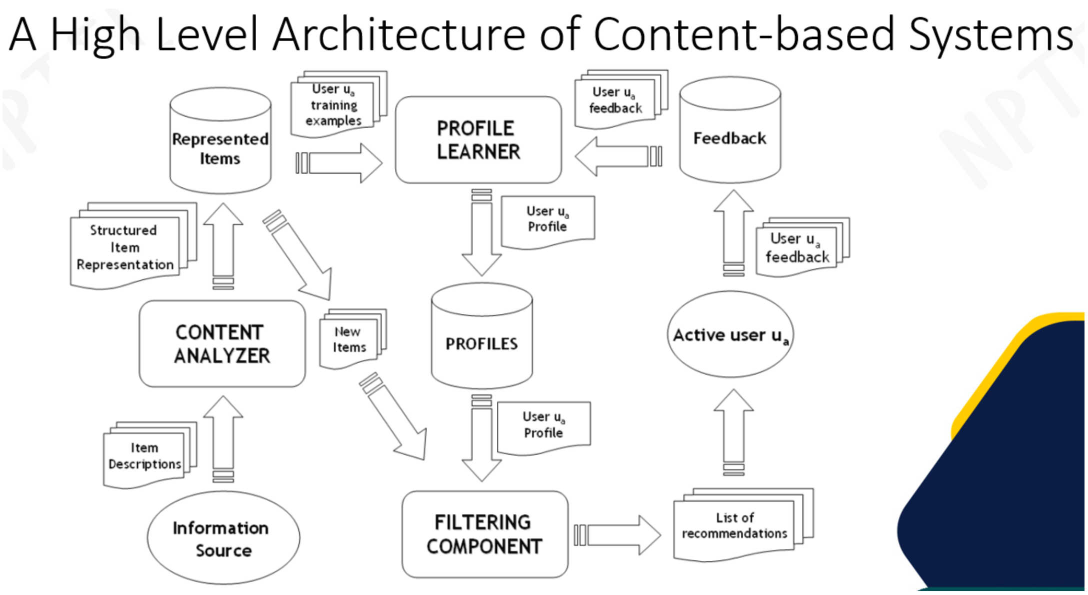
Let us say some m users or some kind of n number of items. And there were many places where we were supposed to predict the rating wherever the question marks were there. Try to remember the slide. So, now, in this content based information filtering system, the content of the item is very important.

Now, what we mean by content of the item? Consider the case of a news recommender system. In case of a news recommender system, the feature of the items are the text features. This text features again can be collected in various ways, but whatever may be the case you have let us say news item 1, news item 2. So, all of them will have some text based feature. Let us say some kind of rating call it TF-IDF.

So, when you have these items, this item description as I told you in case of news items, these are the text features. You have a content analyzer which extract these features. Then once it extracts these features, then these features are represented in terms of vectors. So, this is item 1, this is item 2 and these are the features. So, feature 1, feature 2 and so on.

This one will also have some values for feature 1, feature 2 and so on. So, once you have represented the items which was very in a very unstructured form, now you have represented in a structured form. So, once you have represented in a structured form, what next? You have to build something called the user profile. So, try to remember the data. For each user in a typical data setting, when you have the items over here i 1, i 2 etcetera and let us say this is your target user T.

And this target user T has given ratings in some of the places and not given rating in few places. So, whatever places he has already given rating, it is given, it is given, here it is given, but it is not given here, not given here. So, these makes your training example. So, for item i, it is there. For item 2, it is there.



Let us say this is item x, item x, it is also there. So, your training pattern consists of item i, item 2, item let us say whatever item I wrote item some x and here these are the feature vectors and these are the features and here you have your rating. So, this rating for this particular user. So, you have some rating 1, rating 2, rating 3. So, this data set that you make connecting this user rating with the item feature, you have to use some kind of learning model. That learning model for individual users will be building one user profile model. So, these profile models will be stored here. So, once this profile models are profile model, let us say you have used a decision tree or something profile model is stored here. Then when a new item comes in the term of new item, what exactly is coming? In terms of new item, certain item features are coming. So, i new, feature 1, feature 2 those are coming.

Now, this model is ready here. So, to this model this becomes the input and once you give this input, this filtering component will be deciding whether this particular what is the possible rating for this new item by the user. And once you know this new item rating, then based on this rating there will be many new items. So, you take for all of these you predict the rating using this user profile model. Now, once these ratings are decided, now prediction problem is solved, prediction is one decision problem.

Second is you can find out the top n item out of this. So, this top n items now you show to the user. Now, once you show this top n items to the user, it may so happen that user may like or may not like these items. He may click the new item, he may not click the new item. So, which means you are again implicit in generating user's feedback.

So, this feedback now comes and you have now generated a new training. What is that training pattern? Your choice, your feedback plus these item features taken together. So, new now this new item feature plus your new rating which is implicitly or explicitly generated now enters into this profile learner. So, through this profile learner through this new rating your profile learner has to be updated to build a new profile. So, this process continues and the system becomes robust and more and more examples come. And it can be also used if we have many such users, we can always find out who are the top few users with very high rating and we can for a new item we can always target those users for target advertisement maybe. Next comes your collaborative filtering. In fact, if you look at the history of recommender system collaborative filtering is one of the very first algorithms which was proposed long back maybe around 90s. So, here the recommendation is not like that content based system. Here the user user correlation based on their test is used.

So, this is initially this used to come used to be called as social information filtering. So, the input here the input in the previous case in your content based system what was the input the item features and user preferences. So, which two matrices you used? You used the preference matrix will try to remember we we try talking about three matrices one is user feature, second one is your item feature and the third one is the matrix which connects users with items. So, here in this case this matrix this preference matrix this is the preference matrix. So, this preference matrix is solely used for providing recommendation. What is the what is the mean by providing recommendation? Once again it is determining or estimating the values the the preference values for the items which the user has never seen. Let us say this is user some active user user A. So, for some of the items the user has given certain rating and for some of the items you are supposed to find out the rating. This is given this you have to find out. So, now the question is when I say we have to use user user correlation based on the test and the users matrix is not given what is given to you is this. So, therefore, based on this you are supposed to make top n predictions and based on that you have to make top n recommendations. So, this is a very personalized recommendation system this is one example. Let us say this is the target user this is what I was telling you just now. So, these are some of the ratings which are already given by other users for this particular item that is item 5. For item 1 to item 1 2 3 and 4 this target user has already given rating.

So, its similarity based on this rating with user 1 2 3 and 4 can be found. And based on this similarity using certain procedure will be able to because these users have already rated this. So, some kind of weighted average value will come here weights being the similarity. Now, here the question few questions how do we measure the similarity? So, we have to learn about how to compute this similarity values then how many neighbors to consider? How do we generate the predictions from neighbors neighbors rating? So, well while considering this model besides similarity is there any other model using which we can create the rating behavior recreate the rating behavior of an user? So, all these questions will be looking in detail as we move ahead.

collaborative filtering: Example

	Item1	Item2	Item3	Item4	Item5
Target User	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

- How do we measure similarity?
- How many neighbors should we consider?
- How do we generate a prediction from the neighbors' ratings?
- Can we model the rating behavior of an user?

Then comes your association based recommendation system. So, here what we use is the feature of the item and the rating matrix as well I mean the how the items are bought together. So, it is basically co-occurrence of the items which are bought together. So, this is feature of the item in the sense feature of the item when we say it is about finding this from the rating matrix. So, how the users are how the items are connected by these 5 feature I say how the items are connected we can find it from the rating matrix and 2 types of prediction we can find out and top n recommendations also we can find out.

So, now comes your comes your comes this example. So, here actually we have many transactions that has happened and these are the items bought in this. So, in this association based recommendation system we have to find out which are the items which are frequently purchased together let us look at AB. How many times AB has been purchased together only one, but if we look at let us say AD how many times AD has purchased. So, here is one instance AD has purchased has been purchased here is one instance AD is purchased here is third instance when AD is purchased together. So, AD is a has higher frequency than that of AB.

So, we have to find out certain ways to identify those frequent patterns and from this frequent patterns we will be making something called association rules. So, about this we will be knowing in detail later. So, next is your demographics based recommender system. In case of demographic based recommender system as we know what is demographics? Demographics are this user features. So, here we are supposed to use user user based correlation we have to identify based on demographics. In case of collaborative filtering what we did we found out this user users correlation, but that was from the rating matrix, but here we will be using this user demographics. So, information used here are the individual user preferences from this and again one more thing. When we find out this demographics user has liked so many items these are the ratings from this rating matrix we are now have to derive the specific category liking of the user at a specific category level. So, we know that what are various categories this particular user likes let us say consider the case of movies. So, movies may be you have ah some kind of comedy movies action movies and there are so many genres. So, on individual genre level we have to use some kind of ah some kind of ah ah technique. So, that will be collectively determining from this determining from this rating matrix and the item features we can collectively determining what is going to the likely ah preference of the user at the item ah category level. So, we can so, we have now we are ready to make one machine learning model in which the input will be user demographics and outer output is the category. So, you must have marked if you have continuously let us say watching comedy movies in in ah Netflix you will get one line based on the comedies that are recent or others are watching and on. If you are looking at action movies probably you will be getting one line on the actions. So, so these are something which is determined based on demographic based recommender system.

Association based recommendation system

- Frequent patterns
- Association rules

Transaction-id	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F

The next one is your knowledge based recommendation recommendation. So, so far we have considered about popularity based which was not personalized, then we talked about the content based where we use the item features and the corresponding ah dependent variable we tried deriving from the rating matrix for individual users. Then we had collaborative filtering where collectively ah looking at the preference ah preference values we determined certain user user based similarity and based on that we made certain recommendation. So, also in case of association we understood how the items are purchased together and made recommendation.

Again in demographic based we user we also used the item features as well as the the feature which ah was getting derived from the preference matrix and item category. Why I am repeating this? Because I want to let you know in all these previous cases what we considered is actually those basic three matrices. What are those three matrices? Three matrices are the user features, item features and preferences. Now we have a new type of recommender system which tries explicitly capturing users requirement and one additional source of information is used which is the domain knowledge about the certain item ok. So, this is a bit different from the other where some additional some side information is used. So, if you compare this collaborative and content based which are the basic classical type of in recommendation algorithm. In this case we are using user rating and and their user rating in the sense you how the user are giving and the community rating. Community means what the like minded users based on the user similarity they are giving. In case of context based we are using item feature and user rating. And in case of knowledge based we are using item attributes, users specification collected explicitly and domain knowledge which is infused based on some other ah sources.

So, these are the typical questions that are answered in a knowledge based recommender system. It is bit different from collaborative filtering and others that we discussed so far. What kind of domain knowledge can be represented in the knowledge based? So, you have to maintain a knowledge base and you must know what kind of domain knowledge has to go into it. What mechanism can be used to select and rank the items based on users characteristics? So, the ranking procedure which you already used to find out the top n etcetera has to be now different. How do we acquire the user profile in domains in which no purchase history is available and how can we take the customer explicit preference into account? So, some good interaction strategy has to be adopted.

Now, which interaction pattern can be used in interactive recommender system? So, these are kind of interactive recommender systems. So, you will be shown something some on some user interface, you will provide your input based on that some modification will happen to the recommendations which were already shown and new set of recommendation will be given to you. We may make further modification to your preference and new set of recommendation will be coming up and so on. So, here it is a very very personalized and based on the preference elicitation process.

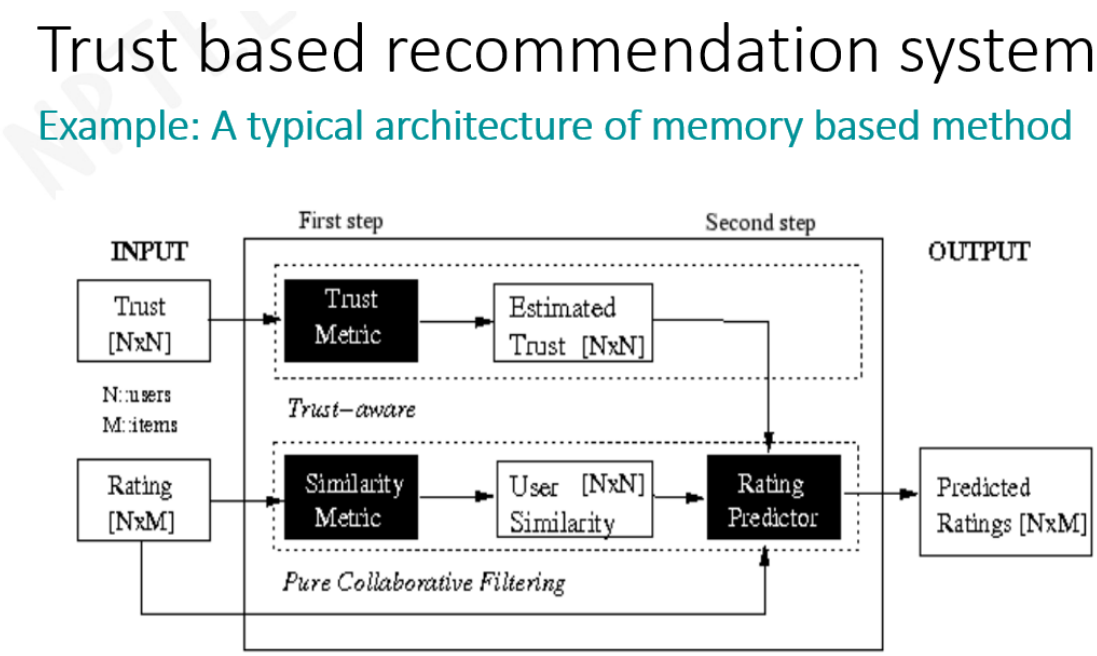
So, it is again this knowledge based systems are broadly classified into two categories constraint based and case based. In case of constraint based the user users need is presented in the form of a set of constraints and those constraints are satisfied from the rules fired from the knowledge base. In case of case based recommender system you maintain how similar in a similar setting other users have taken decisions and you create cases. These cases in the form of knowledge you store in the knowledge base and using this you make the recommendation.

Next is your trust based recommendation. So, in case of trust based recommendation this is as I told you in one of the initial slides there are many problems which arise in a recommender system and needs special attention. One such problem is cold start problem. So, this cold start problem can happen with respect to new user or a new item. In case of a new user he has not already given any rating. So, how are you giving going to what you were doing in case of collaborative filtering you were trying to find out the similarity of this users with other users and trying to give something or in there is another way of looking at it in collaborative filtering in which you compare the items and find out itemize, but this new user has not selected any item as well. So, how do we do it? One of the approach could be we find out from the user demographics what other users have done then try deal with the cold start problem. Here there is another way to deal with this cold start problem.

If we have some additional information regarding if this user is trusting some other user. So, this trust can be explicitly expressed by building something called a trust network or it can be implicitly derived let us say from a social network. From a social network in the sense let us say you are dealing with some user who has certain friends and you know what are the preference pattern of the friends and how well this user is connected in terms of liking the other liking the items which are provided by the friends. So, based on that you can make some arrangement to find out the trust between this new user and other user. Once this trust comes into place so, based on the other user to whom this user trusts you can make some kind of arrangement to make a give recommendation to a cold start user.

So, here basically besides rating matrix you also use something called trust matrix. Besides cold start user even for other users where the rating matrix is very sparse this trust information can help. So, look at this is just one example architecture of trust based system where you have this trust matrix and similarity matrix which is coming out of the rating matrix and this trust matrix is coming out the trust matrix. Now, this similarity and trust combine together to make some rating predictor which is nothing, but one algorithm and using this algorithm you predict the rating. Now, with this additional side information it is observed that the prediction accuracy and other performance criteria also improves.

Then comes the reputation it is similar to the trust based system where you use certain kind of reputation matrix. Reputation basically tells how you respect another user. Respecting in the sense most of the time you refer the user. So, either you have to explicitly again give this reputation matrix how you are respecting other users or this can be implicitly again derived from the social network using some mechanism. Once it is there just like in trust based network trust based recommender system you used one additional matrix that is trust matrix and combined it with user preference matrix to make one predictor ready.



So, here also using this reputation matrix and user preference matrix you can have some recommendation ready. Then comes the recommendation in network. A network when we consider a network a network consists of many nodes and many links which connects each other ok. So, this makes one network. Your Facebook, LinkedIn where you make your network you become one of the node and your your friend also another node and how you connect with your friend this makes your link.

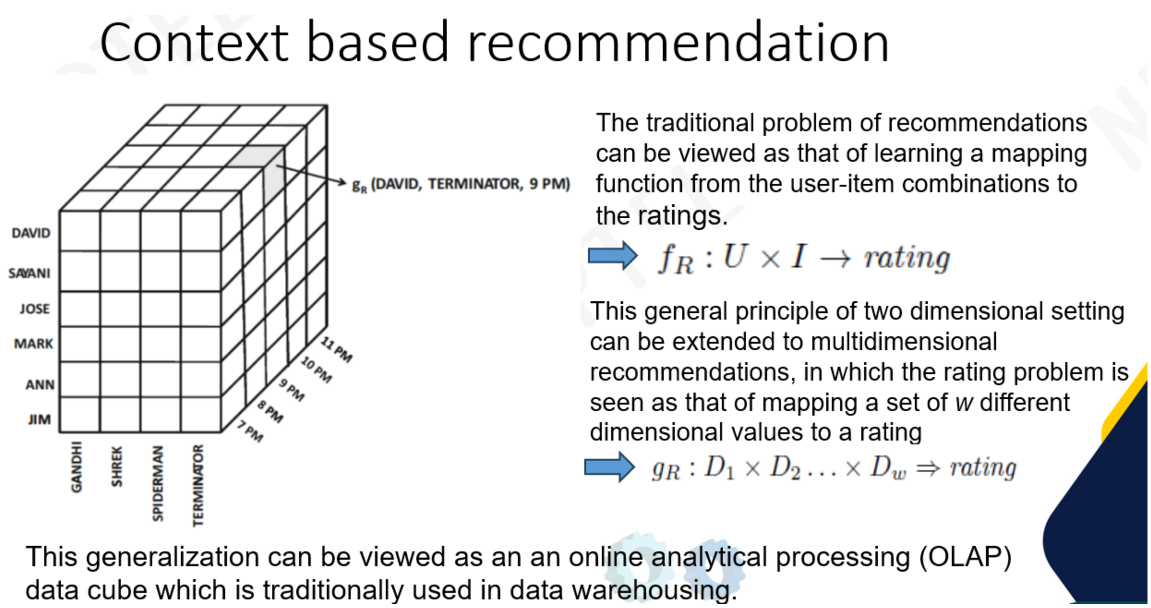
This is something which is explicitly is can be derived from a social network. Think of a network like that of I mean you can many times implicitly derive a network. How can you implicitly derive a network? For example, in a preference matrix in a preference matrix also you can you have many users and items. You can connect the users with items and make one network how the user is connected to items. From that you can again implicitly derive some kind of user network you can derive some kind of item network and based on that you can make recommendation.

Now here you can have various types of recommendation. First three are dealing with recommending the nodes on different settings. The first one is recommending the node by authority. So, in a social network setting there are people whom many people are connected to. Let us say are the prime minister of India many people will be following him ok. So, if he says something it impacts everyone. So, he is an authority. So, similarly let us say for certain product ah let us say some film star whom many people are following is telling something. So, it has to be quickly ah this idea has to be quickly propagated to the other users in that community who are following this particular user. So, these are some of the nodes with high number of links pointing to this node. So, many people follow them ok. So, if they say something others listen. So, they are basically can be targeted once we identify them they can be targeted for branding a new product. Now come to the recommendation recommending nodes by example. When we say example if we have let us say me as a network as a node in a in in Facebook then how similar I am to other users. So, if by chance I buy something I watch something I ah go through certain ah let us say news it is very likely that the people who are similar to me are going to like it. So, therefore, if I can find out other nodes which are similar to me similar to the example node here I am the example node similar to the example node I can target them I can target them for advertisement I can target them for some kind of selling and so on for showing some kind of news items and so on. The third one which is again about the nodes is recommending the nodes by influence and content. So, there are certain nodes let us say social network influencers which who tries propagating messages. They are so well connected that whatever they ah whatever you ask them to say they will be propagating it very fast to others ok. So, therefore, if you want to make something viral ok they can be used for this purpose.

So, here we used it for branding, here we targeted some user and here we tried making some content viral and the last one is not about nodes this is about the links. Typically a network will not be very dense all the nodes not not like this one just like rating matrix every network a social network is a sparse structure only few connections are there. So, more densities more such things can happen. So, how do you make it more dense by suggesting other nodes to whom the target node should connect. So, the friend suggestion that you get in the Facebook kind of setting comes under this link recommendation problem.

So, next comes your context based recommendation. In case of context based recommendation once again you use many side information. So, what are that side information and how how do you represent it we will look at this. Here in our typical traditional problem setting classical problem setting from user item matrix you generate rating. So, here are the users in this particular example and these are the these are the items. So, here there will be certain rating values and some of the spaces will be blank as we already discussed, but if we consider certain additional dimension like here it is considered time you can also have the location.

3D we can see we can have a 4D kind of situation in we we we also have location we can have something else something else. So, you can have multiple such dimension. So, from a flat matrix now we have arrived into a tensor a a high dimensional matrix from tensor structure. So, our problem is now like this from fewer ratings which are in one of such cubicle here let us say here rating is available here rating is available, but most of that places it is not not there. So, only from those few ratings how to predict other ratings and just like I discussed it is rating prediction as well as top n rating as well.



So, the general philosophy of this online analytical processing which is a part of data warehousing naturally comes comes into this. So, therefore, this multidimensional problem ah now this becomes a multidimensional problem which includes context like time location etcetera along with the rating. So, we have now a situation in we we have more information in terms of time location etcetera, but data becomes more sparse. So, we also have to deal with such setting. So, to deal with such setting there are many ways to do it contextual pre filtering, contextual post filtering and contextual modeling.

Here in where where the first two what they do they make the model they make the ah this multidimensional problem bring it to a two dimensional setting and make the recommendation. Whereas, in case of the last one it is actually dealt as a multidimensional problem ok. So, with this now I wind up this lecture these are some of the references which I consider and as I told you this is one this is this lecture is the continuation of the last lecture. So, the conclusions are like this we talked about in last lecture and this lecture together we talked about different types of recommender systems and they depend on three major data sources user item and preferences.

Now, this ah here again we broadly classified into two categories some classical approaches like collaborative filtering and content based filtering.

Then we also discussed about other ah types of ah systems where we are also using various side information. So, with this I end this lecture. Thank you very much. See you in the next class.