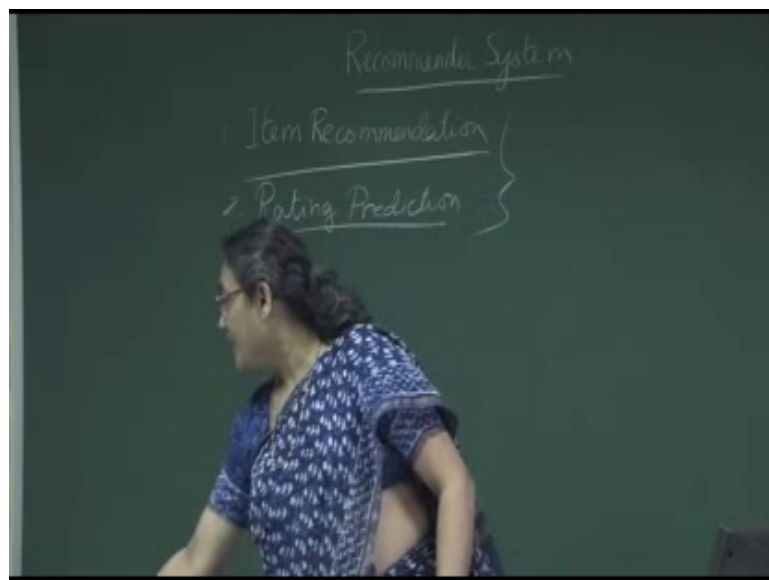


Introduction to Machine Learning
Prof. Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module - 3
Lecture - 13
Collaborative Filtering

In the Part 4 of this module, we will talk about Recommender System and we will show how collaborative filtering based recommendation system is a form of instance based learning. So, first of all let me introduce what we mean by a recommender system.

(Refer Slide Time: 00:35)



Many of you have used e-commerce sites like Amazon, Flipkart and other sites and you may have seen recommendation systems working. So, when you login to those systems, the system recommends certain items to you. So, that recommender system based looks at the past behavior of the users and the other data that it has and tries to recommend items to the user. So, let us formally define a recommenders system.

There are two types of recommender system or recommending methods, one is item recommendation that is based on past history of this user and other users and may be the type of the content the recommender system recommends a list of items to the user

which the user is likely to purchase or likely to consume. The second types of system that we often look at are called rating prediction system. The classic example is with movies. Suppose, a user is asked to rate movies, the users rate movies.

Let us say in the scale of 1 to 5 and different users give ratings for different movies, but the set of total movies of total movies is very large and one user may give rating to number of movies few tones of movies. Now, the objective is to predict that given a new movie what rating the user will give to that new movie that is called the rating prediction problem.

So, rating means rating on a new movie. So, of these are used to decide whether a person will hire a movie, buy a book, read a book who what gives stories and find attractive, recommend friends etcetera. Recommendation systems are. Secondly, in many e commerce almost all e-commerce systems have recommendation with them now you are familiar with search in search the user is actively looking for something user gives a query and the system returns results. User is trying to pull some information from the system. On the other hand, recommendation is a push system where this system pushes items to the user based on users history and may be based on the user's context also formally, we can define a recommendation problem as follows.

(Refer Slide Time: 03:50)

The chalkboard content includes the following text and table:

Recommender System

U : set of users
 S : set of items
 $p: U \times S \rightarrow R$
Learn p from data
Use p to predict the utility value of each item to each user.

User-based nearest nbr
Item-based

	s_1	s_2	s_3	s_4	s_5
u_1	5	-	-	2	3
u_2	3	2	2	-	4
u_3	2	2	4	-	-
u_m					

So, we have a set of users U . So, U is the set of users and we have items. So, S is a set of items and let p_u be utility function that finds out the rating of a user for an item, the usefulness of an item to a user. So, p is a function from $U \times S$ to let us say a real number a rating. So, p is a function from $U \times S$ to R that is for a user for an item the rating of the user for that item that is given by this function p .

Now, the objective is to learn p from data. So, what is the training data? The training data is past ratings of the user for the rating prediction problem which we have written or for the recommendation problem past purchase history or consumption history of the user. So, we learn p from data and based on p , we can predict the utility value of each item to each user. This is the recommendation problem.

Now, there are two broad types of recommendation systems, content based recommendation systems and collaborative clustering based recommendation system. In content based recommendation system, the rating prediction or recommendation is based on the content of the current item and we call content of the previous items that the user recommended or liked. So, based on the similarity of the content of those items that the user liked and those that he discarded we can try to predict whether the new item will be liked by the user or what ratings the user will give to the new item.

So, this is based on contents similarity, for example, if you are trying to predict movies trying to recommend movies to a user. Look at the users; look at the movies the user liked in the past, find out certain features about the movies, what was the genre, who was the director, who were the authors, who were the actors, location etcetera. So, look at the important features of those movies and try to find out what features the user finds it is attractive and based on that you try to predict which movie is the user will like.

The second type of recommendation systems are called collaborative filtering based systems. In collaborative filtering based system the users in order to predict movies for a user we look at what similar users liked. So, for user finds similar users based past data we looked at users, who were similar to the current user that is if we know which movies this user liked and which movies the other users liked, find out the similar users and if those similar users. So, some of the similar users have given a rating for this movie, use

those ratings to predict the rating of the current user, they are called collaborative filtering based method and as we see that this involves finding similar users which is what we did in k nearest neighbor or instance based learning.

In collaborative filtering we present each user. So, so we first construct our data in our data what we have is that, suppose we have a matrix where the different items S_1, S_2, S_3, S_4, S_n , these are the columns U_1, U_2, U_m these are the different users and for a particular user, may have rated some movies. Suppose, this movie is rated as 5, this is not rated, this is not rated, this is rated as 2, and this is rated as 3. For user 2, may be this movie is rated as 3, this movie as 2, this is not rated, not rated, this is rated as 4.

Similarly, we have for different users are presented with the movies and they have rated some of the movies and they have not rated some of the movies and what we want to do is that given a user item pair consisting of a movie, which the user has not rated try to predict the rating, for example, let us say U_3 is another user was rated 2, 2, 4, done. So, if we consider that u_{three} is a similar user to U_2 and U_3 has rated 4 for this movie that will help you to decide what rating you too will give this movie, this is the essence of collaborative filtering.

Now, collaborative filtering there are two types of collaborative filtering methods, one is based on user based nearest neighbor. As I told you that in collaborative filtering nearest neighbor based methods are used to decide, who the similar users are. So, in user based nearest neighbor given a user find similar users. Use the ratings of the test item for that those similar, users to predict the rating of the test item for the new user there is also item based nearest neighbor in item based methods the different items are compared.

Similarity between items are obtained by looking at who are the users, who have rated how, how the users have rated for the item there are two items which most which users have rated in a similar way then those items are considered to be similar, this is an alternative to content based similarity of items. We can use this matrix, we can compare the column vectors to find out the pair wise item similarity and we can use it for the recommendation in this way for a user we look at those items the user has recommended

user has liked and find out items which are similar to these items.

(Refer Slide Time: 12:09)

Collaborative Filtering for Rating Prediction

- User-based Nearest Neighbour
 - Neighbour = similar users
 - Generate a prediction for an item i by analyzing ratings for i from users in u 's neighbourhood

So, for this based collaborative filtering which uses item nearest neighbors can be used.

(Refer Slide Time: 12:13)

Neighborhood formation phase

- Let the record (or profile) of the target user be \mathbf{u} (represented as a vector), and the record of another user be \mathbf{v} ($\mathbf{v} \in T$).
- The similarity between the target user, \mathbf{u} , and a neighbor, \mathbf{v} , can be calculated using the **Pearson's correlation coefficient**:

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\sum_{i \in C} (r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sqrt{\sum_{i \in C} (r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})^2} \sqrt{\sum_{i \in C} (r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})^2}},$$

For such algorithms we have there are phases to these algorithms, look at this slide first

of all we have to find the neighborhood let us look user based nearest neighbor in user based nearest neighbor given a user. Secondly, item we first look at the user and we find the similar users. So, this is the first phase of the algorithm which is called the neighborhood formation phase. Suppose, the record of the target user is u , the target user u is represented by a vector and this vector is the rating that user gave to the items for which he has given the rating.

The user is represented by this vector and we find a similarity between the target user u and another potential neighbor v by using some similarity measure a popular similarity measure is the Pearson's correlation coefficient. Pearson's correlation coefficient can be used to find the similarity between the two vectors u and v as in the numerator. We have summation over all items rating of u . For item i minus \bar{r}_u is the average rating of the user for a item times r_{vi} rating of user v . For item i minus \bar{r}_v average rating of the user v , now why do we have \bar{r}_u and \bar{r}_v here.

Now, different users have different standard for rating, there are some users who are reluctant to give 5, you know if a movie, if the user likes a movie very much he can give a 3 or at most of 4 and not 5 and there are some users who are very liberal with ratings. So, they may give a rating of five to any movie that he likes. So, because users have different standard in rating movies, we want to somehow normalize them by looking at whether the rating of the movie is above his how much it is more than the his average rating. So, that is why in this formula we use $\sum (r_{ui} - \bar{r}_u) \times (r_{vi} - \bar{r}_v)$ r_{ui} is the rating of user u for item i , we sum it over all items.

In the denominator, we have root over summation over items $(r_{ui} - \bar{r}_u)^2$ whole square times $(r_{vi} - \bar{r}_v)^2$ whole square. So, this is for normalization and this is one similarity measure. This type of similarity measure is used to score the similarity between two user u and v . After the similarity of u with all the users has been computed we find the k -nearest neighbors that is k most similar users according to this formula.

(Refer Slide Time: 15:38)

Recommendation Phase

- Use the following formula to compute the rating prediction of item i for target user u
- $$p(u, i) = \bar{r}_u + \frac{\sum_{v \in V} sim(u, v) \times (r_{v,i} - \bar{r}_v)}{\sum_{v \in V} |sim(u, v)|}$$

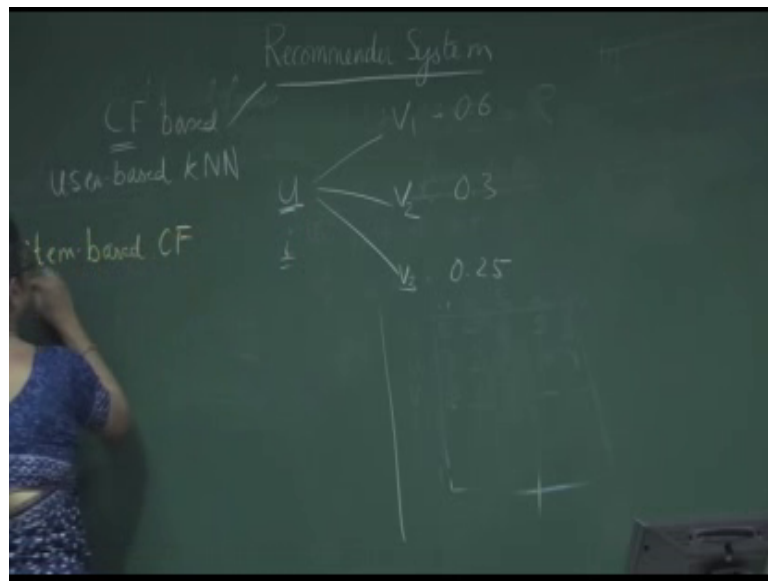
where V is the set of k similar users, $r_{v,i}$ is the rating of user v given to item i ,

And the second phase is the recommendation phase. So, in the neighborhood formation phase, we find the similar users and in the recommendation phase we use the rating of those similar users on the test item to recommend the item to the user.

So, for this, in the recommendation phase we use the following formula to compute the rating prediction of items i for target user u . So, $p(u, i)$ is the predicted rating of user u for item i , it is equal to \bar{r}_u again is the average rating of that user we want to predict whether it will be below average or above average or how much it will exceed or the different from the average and this computed as $\sum_{v \in V} sim(u, v) \times (r_{v,i} - \bar{r}_v)$ where V is the set of neighbor, capital v is the set of nearest neighbors.

So, for all neighbors similarity u, v , similarity u, v is as we computed in the previous phase times $r_{v,i} - \bar{r}_v$. So, for a user v , which is similar to user u , we look at the difference of the rating of v for i minus the average rating of the user. So, if the rating of item i for user v is greater than the average rating of that user this is positive, otherwise this is negative and the weight this particular neighbor is given away to based on how similar he is or she is to the user u .

(Refer Slide Time: 17:37)



Suppose, we have a user U , who is our target user and i is our target item. So, we find the neighbors, let us say V_1 , V_2 , V_3 and they are the most similar users and with each of them we have a value, suppose with V_1 , we have a value 0.6. Suppose, with V_2 , we have a value 0.3 and V_3 , we have a value 0.25 then we look at the prediction of the user. V_1 , for item i and we weight that by 0.6 by this similarity of V_1 with u .

We look at the rating of V_2 for item i weighted by point 3, look at the rating of V_3 for item, weight by 0.25 and to overcome the bias factor because different users have different standards of rating everything. We are doing in terms of the average as we in this slide. So, please look at the slide again. So, we have p_{ui} equal to \bar{r}_u average rating of user u times for all neighbors $\text{sim}(u, v)$ for all neighbors v $\text{sim}(u, v)$ times r_{vi} minus \bar{r}_v and in the denominator we have for all the others neighbors the summation of the similarities.

So, based on this we do our recommendation. Now, the similarity computed can be positive and negative. We take the look at the absolute value of the similarity in order to do this. So, this is collaborative filtering based recommendation system. So, CF stands for collaborative filtering, this is collaborative filtering based recommendation system using k-nearest neighbor or user based (Refer Time: 19:37) neighbor the problem of user

based formulation of collaborative filtering is the lack of scalability, if you have many users in a popular sites like Amazon or Netflix, they can have many users millions of users.

Now, it is not possible in real time to find out the most similar users because as we have seen instance based learning algorithms. So, this types of learning algorithms are lazy at run time you find the most similar users and if the total number of users very large it is difficult without the use of good data structures to find the most similar users in runtime and while several data structures, clustering etcetera can be used to make this more efficient and alternative to user based nearest neighbor is item based nearest neighbor or item based collaborative, filtering in item based collaborative filtering, what we do is that we find similarity between the items we compare the items.

(Refer Slide Time: 21:07)

Item-based CF

- The item-based approach works by comparing items based on their pattern of ratings across users. The similarity of items i and j is computed as follows:

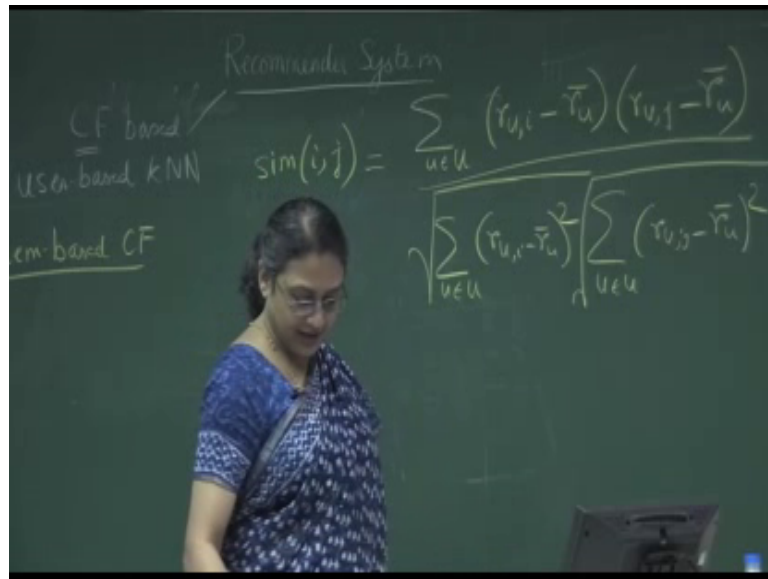
$$\text{sim}(i, j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_u)^2}}$$

CS583, Bing Han, UC

10

So, if you two items i and j .

(Refer Slide Time: 21:21)



Recommender System

CF based
user-based KNN
item-based CF

$$\text{sim}(i,j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_u)^2}}$$

The similarities between these two items are computed as the formula is very similar to what we used for user based collaborative filtering. Here, we sum over all users, we look at $r_{u,i}$ minus \bar{r}_u times $r_{u,j}$ minus \bar{r}_u . So, we look at the rating of user u for item i times the rating of user u for item j and again we use the normalization we subtract the mean for that user and in the denominator, we have root over sigma over all users $r_{u,i}$ minus \bar{r}_u whole square times sigma over all users $r_{u,j}$ minus \bar{r}_u whole square right. So, based on this we find the similarity between the items.

(Refer Slide Time: 22:53)

Recommendation phase

- After computing the similarity between items we select a set of k most similar items to the target item and generate a predicted value of user u 's rating

$$p(u, i) = \frac{\sum_{j \in J} r_{u,j} \times \text{sim}(i, j)}{\sum_{j \in J} \text{sim}(i, j)}$$

where J is the set of k similar items

CS583, Bing Liu, USC

Now, once we find a similarity of the items in the neighborhood formation phase then we go to the recommendation phase. In the recommendation phase, we select a set of k most similar items to the target items. So, we have a target item i and for the target item we find the most similar items.

(Refer Slide Time: 23:16)

Recommender System

CF based ✓
user based kNN
item based CF

$$\text{sim}(i, j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_u)^2}}$$
$$\hat{r}_{u,i} = \frac{\sum_{j \in J} r_{u,j} * \text{sim}(i, j)}{\sum_{j \in J} \text{sim}(i, j)}$$

So, target item is i . We find the k most similar items i_1, i_2, \dots, i_k based on this formula that we have written right and then we do the rating prediction. The rating prediction is done \hat{p}_{ui} as $\frac{1}{\sum_j \text{sim}(i, i_j)}$ take the summation over the similar items i_j are the k similar items we do r_{uj} times the similarity of item, i item j divided by $\sum_j \text{sim}(i, i_j)$ again over all neighboring items $\text{sim}(i, i_j)$.

So, based on this the rating is predicted, this item based collaborative filtering. So, with this brief introduction we come to the end of our recommendation systems, which is an application of k -nearest neighbors and we also come to an end of this module after this the next module that we start is on probabilities Bayesian learning, naive bayes and few such things of course, we will also have one class by our teaching assistant on how to use this nearest neighbor and instance based learning type of algorithms in some application.

Thank you very much.