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 - 15

Lecture 15: Additional Topics in Neighbourhood Based Approach

Hello everyone. We are going to continue collaborative filtering and in this regard we have started discussing about neighborhood based collaborative filtering and today we are going to talk about few additional topics based on whatever lectures we have done so far. So, these are the topics to be covered. We will be comparing neighborhood based methods that has been studied so far that is user based and item based methods. Then we will look at a regression view of the neighborhood based method, neighborhood based models. Then we are going to talk about graph based neighborhood methods.

We will just give a brief introduction to this topic. We can compare user based collaborative filtering and item based collaborative filtering on many aspects. So, first of all we start with accuracy. Accuracy of a neighborhood recommendation method depends mostly on the ratio between the number of users and number of items in a system.

Typically, in a e-commerce system like that of Amazon and all, the number of users are much more than the number of items. In such situations, item based collaborative filtering is likely to give better accuracy and why it is supposed to give better accuracy? Better accuracy of a model depends on the data availability. When the number of users to remind you let me remind you we have users here and we have items here and in this space we have ratings. So, if the numbers of users are more, then per user rating will be quite less and per item rating will be more. So, in case of a item based collaborative filtering, the number of ratings are going to be much more in case number of items are less.

So, we have more data here and more data will definitely improve accuracy. However, this may not be the situation all the time. Now consider the case where these items may be more. One such situation could be the news recommendation where every day some news, new news at a time article comes. So, which means we have very large number of features here.

So, in case of news recommended system like situation, we have very large number of items compared to users. So, under this setting the number of ratings available per item is going to be much less than the number of ratings available per user. So, therefore, user based collaborative filtering is going to give you better accuracy. Then next we compare both UBCF and IBCF based on efficiency. Now, what is efficiency? Efficiency can be we can think of efficiency in terms of both memory utilization and computational efficiency.

How fast the computation is taking place? Now recommended system here also depends on depends on the ratio between the number of users and number of items. Now when the number of users exceed the number of items, your item based collaborative filtering which actually finds the similarity between the items will be much faster. So, therefore, the such system will be scalable. Now if we look at the time complexity in case of online recommendation phase both the user based and item based methods are going to be taking the same time. Because similarity computation is something which happens in the offline setting and IBCF requires less memory and time because number of items are less.

Now in practice the user rates only few of the available items. Therefore, the non-zero similarity weight will be I mean you only have to store the non-zero similarity weights. So, as a result the size of the similarity matrix also gets reduced in case of item based similarity systems. Now this number can further be reduced by storing as for each user the only the top n weights. What is this top n weights? One of the decision making scenario in case of recommended system is top n items.

So, in case you are supposed to predict top n items you can sort them based on the available and predicted rating which is based on the similarity values and you can store this. So, if you keep only I mean this is only possible if the users are known in that particular setting. Think of a situation like Netflix where you log in to the system or let us say Amazon Prime where you explicitly log in. So, your history is specifically stored. If your history is not stored, then against the new user top n weights definitely cannot be stored.

So, in the same manner if you have non-zero weights you can you can have computational efficiency without having to test each pair of users and items which makes the neighborhood methods scalable in case of very large systems. Now next we compare both based on another thing called another metric called stability. Now what is stability? How fast the recommendation process will change? So, if you have to choose between user based and item based approach it depends on how frequently the amount of information changes. In case of item based system which is more stable because the items do not change very frequently. So, the similarity computation will not be affected much even if new ratings come in.

On the contrary in applications where the list of available items constantly change for example, your news recommender system user method user based methods can be more stable. So, if the items do not change frequently item based methods are stable in case they change frequently user based methods are stable. Now this next thing next metric could be justifiability. How you can justify, how you can give explanation to the result of the recommender system? In case of item based collaborative filtering you compare items. So, therefore, you can say based on the items viewed by the user what other items of similar type can be predicted.

You must have noted this in case of let us say next Netflix based of recommender system. If you are watching comedy movie, then probably you will be getting recommended comedy movies only they will be writing this is recommended because you watched this. Now by modifying the list of neighbors and their weights and or their weights then it is become then it can be possible that you can actually narrate to the user how this recommendation took place as I have told just now. But in case of user based method your rating reflects the rating of other people. Now who are those other people? User may not be aware of who those other people are and why they are supposed to be similar with him.

So, explaining this process to the user may be little difficult. Now next metric is serendipity. Of course, we are going to talk about various types of metrics which will be used in case of recommender system setting at a little later in this course. But serendipity just to let you know it is about the surprising element in the recommendation process. So, it may so happen that a particular user is looking at let us say laptops in consecutive measures because he is interested to buy a laptop.

So, therefore, he is searching for laptops viewing various data about the laptop looking at the user comments and so on. But it is ridiculous to think that that particular user will always buy laptop because laptop probably in few years he will be buying once. So, in such situations serendipity and related other measures other metrics which are consequence of this become very important. Then we go for item based collaborative filtering item similarity measure. So, you tend to recommend the items which are similar to that particular item.

When I mean in situation like that of let us say movie it may be a safe recommendation because maybe that particular user is always interested in watching comedy movies. But in case you want to provide some kind of diversification to the recommendation then you are supposed to look at what other people are with similar interest are watching. So, as a result UBCF is going to give better value for this particular metric. So, with this knowledge of how we can compare user and item based collaborative filtering. Now let us look at a completely renewed view of recommender system neighborhood based recommender system.

So, a regression model of neighborhood based recommender system. So, let us try remembering what the regression model was. In regression model there used to be one response variable and it was supposed to be computed from some attribute some features. So, you modeled your predicted value like this and with known values of y you were trying to minimize the error. So, you were trying to minimize and in this process you were discovering this model parameter.

So, we have a situation similar to this in this neighborhood based method setting. Now look at this in case if this particular model we used for predicting some rating for user u for item j where of course, probably we used Pearson burst similarity where it was mean centered. So, this was the mean value of that particular user then we found out the similarity using some kind of similarity method let us say Pearson similarity. Then we were from the like minded users in the set of p u j where these users have also rated item j and v is one such user we were trying to find out the rating for r u j following this formula. So, in this formula it was basically our assumption that similarity is a good measure for the weight.

So, basically this is a weighted average. So, what is what was our assumption? Our assumption was similarity is a measure a good measure for the weight for the people who have already rated j. Now I have not rated j, but my similar neighbors have rated item j. So, based on their rating some rating will be predicted to me, but whatever may be the case assuming that this thing is similarity value is not known what is our m? Our m is to predict this similarity. So, if in case of a user based and item based what we did? We computed this user based similarity beforehand and we were using it.

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u(j)} \text{Sim}(u,v) \cdot (r_{vj} - \mu_v)}{\sum_{v \in P_u(j)} |\text{Sim}(u,v)|}$$

Now in a regression based model what we will do? We will be assuming that this similarity value is supposed to be predicted from the regression model that we make and we do not have to explicitly compute it. So, therefore, more formally in user based collaborative filtering we I mean in both user based and item based collaborative filtering we assume that we have a linear function to make the prediction and in this linear function the similarity values are the weights. Now how can we make this regression model? So, in case of user based regression nearest neighbor regression model look here we were using similarity and it was it was getting normalized as well. So, instead of that we can use a simple value and this may be the normalized equivalent of the normalized similarity, but here we are supposed to we do not know this value. So, we are supposed to determine this value.

So, just like in ordinary multiple linear regression what we did? We took the so now, to for a regression we need certain example patterns. So, what are our what are our example patterns here? Our example patterns here are the known rating. So, for all the ratings users from user 1 to m for all the ratings for each user which is rated by this person we will be subtracting this known rating by its estimated value or the model that we have decided. Now while doing so the unknown will be this one. So, and we are minimizing the square error.

So, for all the users taken together we can minimize this or for individual users also we can minimize this one. In that case this particular term will not exist and we will be doing it only for this term. So, for individual users we can do it and in this particular case we have shown it that for taking all the users together we can also have this optimization problem. Now given this optimization problem rest of the things we do not want to discuss because during this machine learning introduction we know that to solve such kind of problem we use gradient descent or some or some variation of gradient descent like stochastic gradient descent.

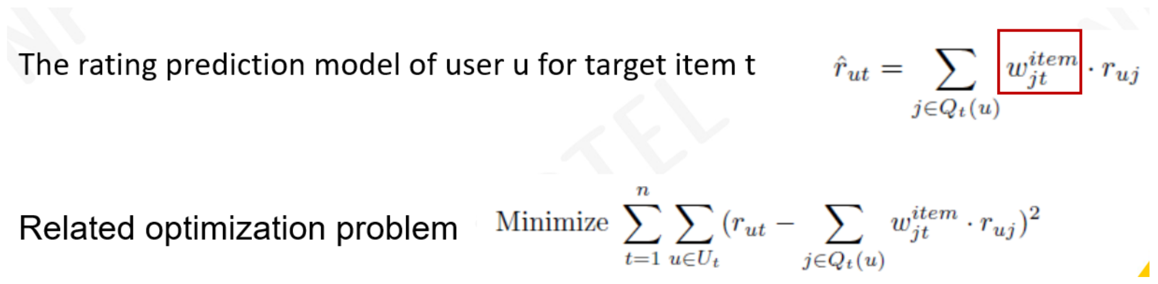
$$\hat{r}_{uj} = \mu_u + \sum_{v \in P_u(j)} \boxed{w_{vu}^{user}} \cdot (r_{vj} - \mu_v)$$

$$\text{Minimize } \sum_{u=1}^{m} J_u = \sum_{u=1}^{m} \sum_{j \in I_u} \left(r_{uj} - \left[\mu_u + \sum_{v \in P_u(j)} w_{vu}^{user} \cdot (r_{vj} - \mu_v) \right] \right)^2$$

So, we are not going to discuss. So, we limit ourselves to only the definition of the optimization problem. Now similar situation arises in case of item based method. In case of item based method with respect to each item T where user rating is available only for few items I mean the only for few users the rating it available for item T. This is the non rating, this is the estimation and we take the squared error and we try minimizing. Now moving ahead, a more generalized model is sparse linear model which is also called the slim model.

So, this slim model refers to family of sparse models because they use the concept of regularization to make sure that sparsity is maintained in the data. So, why sparsity has to be maintained? So, what exactly we were trying to determine through those two previous user based and item based regression models? We were trying to find out the weights associated I mean the user user weights and item item weights. Now considering the fact that the rating matrix is most of the time sparse this values are going to be sparse as well. I mean the most of the entries in the similarity matrix. Similarity matrix what is the dimension of similarity matrix? In case of user user it is let us say number of user is m and number of items are n then m cross m will be the dimension of user user similarity.

Similarly, n cross n will be dimension of the item item similarity. Now in both the similarity matrices maximum entry will be 0. Why? Because in the original m cross n feedback matrix or m cross n rating matrix many of the items are missing. So, therefore, you cannot find out similarity ok. So, now, one assumption of this sparse linear model is rather basic requirement of sparse linear model is it works best with non-negative rating value non-negative rating value.



The rating prediction model of user u for target item t $\hat{r}_{ut} = \sum_{j \in Q_t(u)} \boxed{w_{jt}^{item}} \cdot r_{uj}$

Related optimization problem $\text{Minimize} \sum_{t=1}^{n} \sum_{u \in U_t} (r_{ut} - \sum_{j \in Q_t(u)} w_{jt}^{item} \cdot r_{uj})^2$

So, therefore, there is no need that you have to make your data mean centered because the moment you make your data mean centered some of the ratings are going to be negative ok and they will represent dislikes. So, therefore, this method is most appropriate for the situations where non-negative ratings are not available as such or you do not have to normalize the you do not have to make the data mean centered. So, this is mostly applicable to implicit feedback setting where the missing values are treated as 0 and the available ratings are basically represent the stops like that of click through rate, sales data, click through rate, sales data and so on. So, these kind of positive preferences about the users action is collected and if it is not collected the value is said to be 0. So, under such situation this slim model is very appropriate.

Now, let us look at the slim model. Now, if you look at the slim model just like your regression model this one also works with the available rating. Now, for user u equal to 1 to m for whatever items it has rated which is available through your model u this is your model through your model u I mean the u represent your model and here you are supposed to determine this one. So, this item based similarity of jth item with tth item has to be determined. Now, while determining it one of the requirement for making the model little bit less computationally burden we can think of making the values this w jt values to be 0 when j equal to t. So, which means let us not compute the similarity between 2 items because unnecessarily it will give you very high value and that same item will keep on getting recommended.

So, we will be setting this in the constraint. So, look at this though we did not discuss about the optimization problem in great depth let me tell you optimization problem can be unconstrained or it can be constraint. So, in a unconstrained problem you have only objective function in case of constraint problem along with the objective function we also have certain constraints. So, these are supposed to be these 2 are supposed to be the constraints.

So, these 2 are constraints. So, our objective function is this. So, our objective function is to minimize the difference between the actually available rating and rating predicted from this model under this constraint and this what does this constraint tell? We are interested for the items with positive similarity. So, this is a basic assumption of this model. So, that is being taken care of by this particular constraint. So, now along with these 2 more terms are appearing.

Try remembering what we discussed at the time of learning about when we discussed about machine learning foundations. We are supposed to add some regularization terms. Why do you add regularization terms? Our aim is to avoid over pitting. So, to avoid over pitting we were talking about adding this L2 regularizer this is L2 regularizer ok. And this L2 regularizer is also called your ridge the regularizer corresponding to ridge regression.

At the in addition to this we also have this lasso component. And what is the and this is your L1 and about this L1 and L2 norm etcetera we discussed when we were talking about the distance matrices. So, this is L1 norm. So, we will be taking both L1 and L2 norm and corresponding regularization parameters which we can change to get different kind of solutions. So, therefore, in case of in case of L1 regularization what we try to control? We try to control how many values will be 0.

So, either it will be straight away you can make it 0 or there will be some value. So, to enforce this sparsity in the data L1 regularizer is a very good approach. And both of them taken together is called the elastic net regularizer. So, these two together is called elastic net regularizer. And using this when you solve it this one try avoid help you to help you to avoid over pitting and this particular term makes many of these W j t values to 0 if they are not significant.

As a result, you get considerably very good values of those. So, next we talk about the graph based the graph based neighborhood methods. So, this is particularly important when you have the sparse data. So, in case you have sparse data which means directly the user is not giving any rating, but look at a situation like this. The data this data may be sparse many of the ratings are not available.

However, if we can and when we are here in this directly when we are doing from the rating matrix our neighborhood was considered to be those users who have correlated the products. There is some co rating where available. For example, consider user U 4 and U 2. There is absolutely no overlap see this person has rated these two this person has rated these two.

So, they do not have any correlated items. But when you represent it in the form of a graph how do we make this? This is a user item bipartite graph. So, how do we make this graph? This side you have users this side you have items and if any user rates any item you have a link otherwise you do not have a link. Now in this setting if we look at user U 1 sorry user U 2 and try to find out the similarity with U 4 we may have a transitive way of connecting both user U 2 and user U 4. So, user U 2 is connected to this this one is connected to this this one is connected to this. So, this one is connected to now U 6 U 6 connected to this this is connected to U 4.

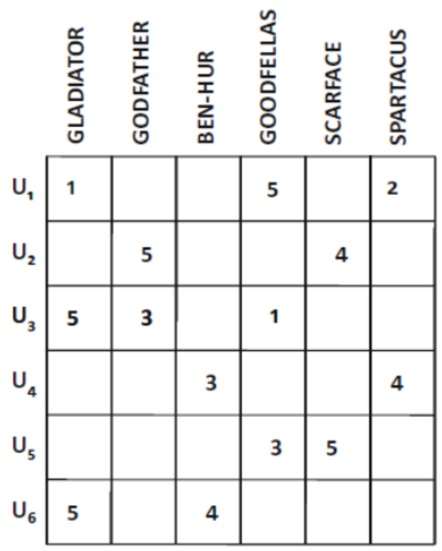
$$\begin{aligned}\text{Minimize } J_t^s &= \sum_{u=1}^{m}(r_{ut}-\hat{r}_{ut})^2+\lambda\cdot\sum_{j=1}^{n}(w_{jt}^{item})^2+\lambda_1\cdot\sum_{j=1}^{n}|w_{jt}^{item}|\\ &= \sum_{u=1}^{m}\Big(r_{ut}-\sum_{j=1}^{n}w_{jt}^{item}\cdot r_{uj}\Big)^2+\lambda\cdot\sum_{j=1}^{n}(w_{jt}^{item})^2+\lambda_1\cdot\sum_{j=1}^{n}|w_{jt}^{item}|\\ &\text{subject to:}\\ &w_{jt}^{item}\geq 0\quad\forall j\in\{1\ldots n\}\ \text{ and }\ w_{tt}^{item}=0\end{aligned}$$

So, there is at least a path starting from U 2 and connecting to U 4. So, maybe you this path is very long and you can give very less importance, but at least you have some way to fill up this matrix. So, when we recommend items using this graph representation we utilize two properties propagation and attenuation. The graph that cannot be directly the nodes that are not directly connected directly influenced each other they are not connected influence each other. Then by propagating information along the edge of the graph they can indirectly connect to each other.

Just now we saw the example that when we have item user U 2 and U 3 do not have any correlated product still then also we were able to find a path. Now, second property is we have to take care of the attenuation. What is attenuation? As the path size of the path becomes long so, which means the transitive relationship becomes very weak as we get a long path the let us say we try as finding certain similarity value between those two users this is going to be very less. So, the second property says that. Now, what are the strategies? So, here the proximity of a user U to an item I in the graph is used directly to evaluate the rating of U for I.

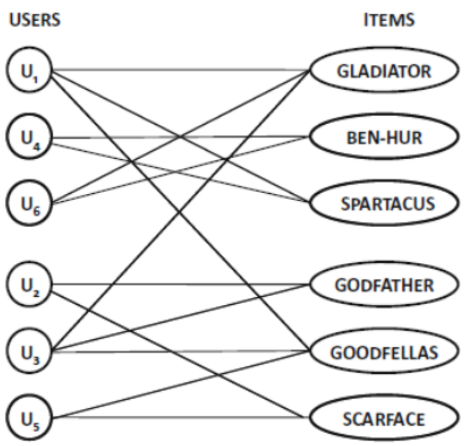
Following this idea the items recommended by U the system U by the system are those that are closest to U in the graph. So, those items which are close to U are going to be recommended to U. Then proximity of two users or items which are represented nodes in this graph is also used as a measure of similarity. So, this similarity this value can be used to compute this weights like user user weight or item item weight and we can use them in the directly in the neighborhood based recommendation method. Which are the neighborhood based recommendation method? User based and item based IBCF and UBCF.

Now, while computing this similarity there are many ways. So, this path based similarity takes care of the distance between two nodes in the graphs and represent in the function of a is a function where the input is the number of paths connecting the two nodes. So, there are many ways you can compute this similarity using shortest path, number of paths, random walk based similarity, item rank based similarity and so on. To start with when we talk about the ah shortest path this is basically is used while finding the similarity between two users. Now, from a ah user item bipartite graph looking at how they are connected through the edges we can connect create one user user graph. So, this user user graph you can say it is a kind of implicit network among the users.



	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

(a) Ratings matrix



(b) User-item graphs of specified ratings

Now, in this user user graph when you construct two things we have to keep in mind. So, these two things like hoarding and this hoarding and predictability. So, these are the two notions which are used while constructing this user user graph. While ah it is not that every user will be connected to every other users. So, if these two properties are satisfied hoarding which is a symmetric relationship between two users that that is satisfied if these users have rated similar items.

So, similarly there is called something called predictability. The relation of predictability are presented as directed edge in the graph such that there is a directed edge from u to v if v predicts u. Accordingly, a directed path connecting the two users u and v represent the transitive predictability of v for rating ratings of u. So, now with this once you decide whether an edge exists between two users then from the new implicit user graph you can find out the shortest path and this shortest path can be used as a measure of similarity. Second is number of paths. So, this when you talk about the number of paths there will be numerous paths.

So, you have to decide certain upper limit of how long I can go. The last example that we are discussing with many transitions we were connecting from u 2 to u 4. So, in that case we can now decide what is the maximum number of transitions we will be looking at. This when if you are considering user user similarity starting from user node and trying to find out ah predict certain item then if it is odd k value of k is odd then you will be reaching at the item side if the value of k is even you will be reaching at the user side. So, for the prediction purpose take the value of k to be some odd number and try moving in the graph and find all the paths of size k. Of course, all these problems are quite complex, but this we are just talking about the notion.

Random work based similarity. So, in case of random work based similarity the movement within the graph is modeled as a first order Markov process. In a Markov process the transition takes place from one node to the other. Now looking at this transitions how we can go from one user to the other or one item to the other or in the movement in the bipartite network we can construct a transition probability matrix. Now from this transition probability matrix we can follow the principle of random work which you have to study little bit more to know about Markov process and how random works are generated from Markov processes.

So, using this as a similarity measure you can go ahead. This item rank based similarity is quite ah close in the idea to the page rank algorithm which is typically used for predict ah web pages during search. So, in this case also you create or transition probability matrix by connecting like looking at how the ah user u I mean this approach is racks the preference of user u for new item I as the probability of u to visit I in the random work of a graph in which the nodes corresponds to the items of the system and the edges connects items that have been rated by the common user. Now from this probability matrix with item rank we can assume if there is a direct connectivity with probability alpha a person can move or user can move which means the user can rate or user can go to the other item and with a probability 1 minus alpha it can make a random move which is not a part of this transition probability matrix. We can also compute again the transition probability matrix and the concept of Markov chain comes in here from the transition probability matrix you can find out the first passage time. What is this is the average number of steps needed by a random worker to reach node j for the first time when it starts from node any other node any other node i which is not j.

So, considering this first passage time we can also compute the similarity value. So, with this we wind up this lecture. So, these are the books I referred and specifically most of these contents I have taken from first two books. These are my concluding remarks.

The accuracy of neighborhood method neighborhood based recommend recommendation method depends mostly on the ratio between number of users and items in the system.

Neighborhood based models are heuristic variant of the linear regression models in which the regression coefficients are heuristically set to be the similarity values. Now this can be modeled as a regression model where the similarity values can be determined this coefficient values can be determined this weights can be determined from the model. The sparsity of observed ratings can cause a major problem in the computation of similarity in neighborhood based methods. Now graph based methods provide certain approach in which the transitive relationships can be considered and similarity can be computed as a result similarity matrix will not be very sparse. Thank you.