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 - 14

Lecture 14: Item-Based Approach

Welcome back. We continue with our last lecture on collaborative filtering that to neighborhood based collaborative filtering. And we have already discussed about similarity and distance measures. Then we also covered about user based similar user based collaborative filtering. Now we are going to talk about item based similarity item based collaborative filtering. So now in item based collaborative filtering, which is similar to your user based collaborative filtering, we use in user based similarity matrix is a I am just starting this again.

In case of item based model, we construct the similarity matrix comparing the items rather than the users. In case of user based what we did? We made the similarity matrix comparing the users. So rest of the steps are similar in case of item based collaborative filtering when we compare it with that of user based one. Now the phases are exactly same.

You have to prepare the data, reduce the dimension, form the neighborhood and generate the recommendations. Now the data preparation in case of item based collaborative filtering is exactly same that of user based collaborative filtering. However, here when we try finding the when we try normalizing the data, what we were doing that case? Suppose this is our user item matrix. So now item wise we are supposed to take R U i cap. So here the ratings.

Normalization ⇒ $h(r_{ui}) = r_{ui} - \overline{r}_i$, **Prediction** ⇒ $$\hat{r}_{ui} = \overline{r}_i + \frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij}(r_{uj} - \overline{r}_j)}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|}.$$

Normalization ⇒ $h(r_{ui}) = \frac{r_{ui} - \overline{r}_i}{\sigma_i}$ **Prediction** ⇒ $$\hat{r}_{ui} = \overline{r}_i + \sigma_i \frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij}(r_{uj} - \overline{r}_j)/\sigma_j}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|}$$

Let us say this is some user U. So this R U i. So there will be many such with respect to a particular item. So item wise you have to take the rating. And similarly while making the prediction, you will be adding this item rating mean.

And the process is same in case of Z-score normalization where additionally this standard deviation is considered. So here again we measure the similarity using typical measures and similarity between the items can be computed. Now coming to this computation step. This is one example with adjusted cosine similarity. So as you can see we are now finding the similarity between item 1 and 3.

This is one mean centered data and we are finding the similarity between item 3 and item 1. So which are the so here which are the common correlated items? This one, this one, this one, this one, this one, this one. So here there is a missing value, here there is a value, here there is a missing value. So those things will not be considering. So which means ultimately for item 3 and 1 we are trying to compare.

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

So item 3 vector is 1.5, minus 0.5 and minus 1. And corresponding for corresponding user entries in item 1 it is 1.5, then minus 1.5, then minus 1. So 1.5, 1.5, minus 1.5, minus 0.5, minus 1, minus 1 and rest of the things are self-explanatory. Here also we have top K filtering, threshold filtering and negative filtering for neighborhood formation. In case of as we have already discussed in the last lecture top K filtering means we will be selecting the top few items. So now this is the cosine similarity of item 1 with item with all the j items, item 1 with all the j items. So which are the most similar to item 1, item 3 and this one.

So if we go by top K these two are the top two. If we go by threshold let us say our threshold is 0.5, then again these two. Suppose we go by negative filtering then these three will be removed. Then writing prediction, exactly same. Everything is that this is now item-item similarity. And comparing this in that case in case of user-user you were comparing rows and finding the similarity. Now you will be comparing columns and find the similarity. And concept of this H inverse is again I have we have already discussed last time. If whatever kind of transformation to the data you apply depending on the kind of similarity computation you use, so that same transformation has to be brought in.

So if at all you are doing it mean centered then mean has to be added and so on. So this is about rating prediction. This was the Orison matrix. This is the mean centered matrix. And when you use this similarity now who are the top two? For user 1 what is the problem? Item 1, this is R31.

Item-Id ⇒ User-Id ⇓	1	2	3	4	5	6
1	1.5	0.5	1.5	-1.5	-0.5	-1.5
2	1.2	2.2	?	-0.8	-1.8	-0.8
3	?	1	1	-1	-1	?
4	-1.5	-0.5	-0.5	0.5	0.5	1.5
5	-1	?	-1	0	1	1
Cosine(1, j) (item-item)	1	0.735	0.912	-0.848	-0.813	-0.990
Cosine(6, j) (item-item)	-0.990	-0.622	-0.912	0.829	0.730	1

$$\text{AdjustedCosine}(1,3) = \frac{1.5 * 1.5 + (-1.5) * (-0.5) + (-1) * (-1)}{\sqrt{1.5^2 + (-1.5)^2 + (-1)^2} \cdot \sqrt{1.5^2 + (-0.5)^2 + (-1)^2}} = 0.912$$

This value is missing. For R36, this value is missing. So for this who are the top two? For this one item 1, top two are this one and this one. For item 6, top two are this one and this one. And their corresponding ratings are getting multiplied from the original data and you get these values.

We here also we have this voting based mechanism. I want to be discussing further because we have already discussed about this voting based mechanism for discrete rating. Now in both user based and item based collaborative filtering, dimensionality reduction and discovery of latent feature is important in case you would like to have a different view of computing similarity. In case the data is sparse, so directly so this can be performed using SVD, singular value decomposition. But in case of data sparsity, you can deal with the sparsity problem using something called UPD composition with appropriate computational steps.

$$\hat{r}_{ui} = \frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij} r_{uj}}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|} \qquad \hat{r}_{ui} = h^{-1}\left(\frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij} h(r_{uj})}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|}\right)$$

So UPD composition we will be discussing in one of the future lectures, but this is one of the slide which I have already shown you. Now if through SVD you can actually connect users with concepts. So you what are the concept here? Concept here was of course this latent concepts cannot be named all the time, but in the particular case this was with certain as general G1, this was certain other general G2. So we saw that this particular group liked G1, this particular group liked G2. Or in other words, if you look at the characteristics in terms of these latent features, they are distinct for both the groups.

Right? Similarly, for items also we got the ideas the latent features connected to the movies. And we saw that through this dimensionality measure, dimensionality reduction measure, even if when we remove some of, look here many of the entries were blank. Assume that all these blank entries do not have any value. And we would like to and the rating scale is between 1 to 5 and these 0 values actually indicate nothing. I mean the non-availability of the data.

IBCF: Rating prediction

Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6
1	1.5	0.5	1.5	-1.5	-0.5	-1.5
2	1.2	2.2	?	-0.8	-1.8	-0.8
3	?	1	1	-1	-1	?
4	-1.5	-0.5	-0.5	0.5	0.5	1.5
5	-1	?	-1	0	1	1
Cosine(1, j) (item-item)	1	0.735	0.912	-0.848	-0.813	-0.990
Cosine(6, j) (item-item)	-0.990	-0.622	-0.912	0.829	0.730	1

$$\hat{r}_{31} = \frac{3 * 0.735 + 3 * 0.912}{0.735 + 0.912} = 3$$

$$\hat{r}_{36} = \frac{1 * 0.829 + 1 * 0.730}{0.829 + 0.730} = 1$$

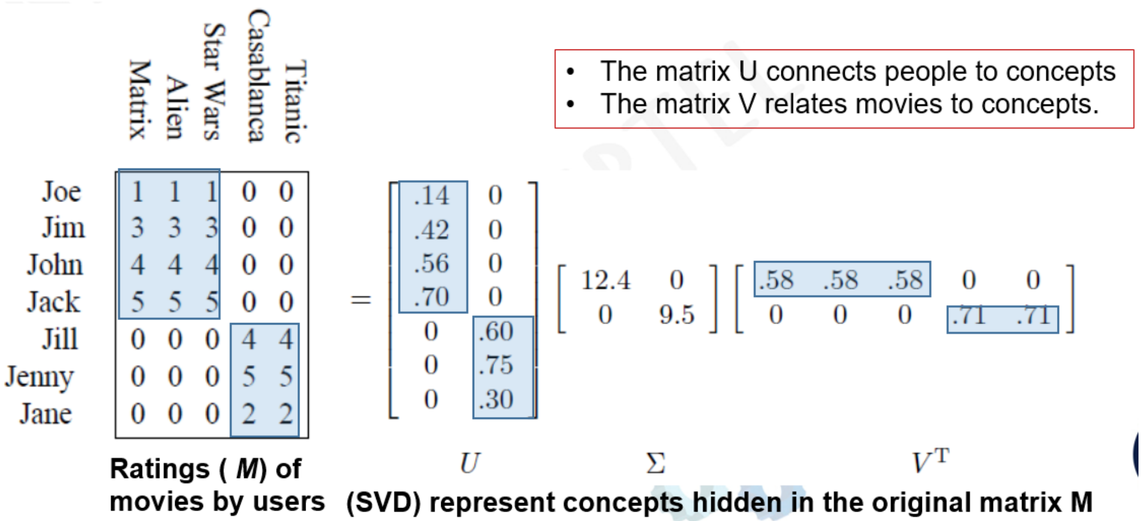
Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6
1	7	6	7	4	5	4
2	6	7	?	4	3	4
3	?	3	3	1	1	?
4	1	2	2	3	3	4
5	1	?	1	2	3	3

So in that case if we drop this latent feature as well as corresponding singular value and we multiply the matrix, we will be getting a complete matrix. So when we have this complete matrix, now two users and two, I mean all these ratings are basically predicted. Similarly, if we would like to use these two for making user-based similarity, you have values for all the users. So you can make user-user similarity using this. You do not have to even do this computation.

So similarly using these two you can compute item-item similarity. Okay? Now to, in case this data is sparse, such straightforward computation may not be possible. So therefore, to discover this U and V matrix, now this is U matrix, this is V transpose. So actually we did not give this matrix. What was this? This is singular value matrix and without giving the singular values, what we lost is the orthonormality criteria of these features.

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

So doing little compromise with this orthonormality criteria, getting rid, I mean not using, not having this orthogonality, orthonormality criteria, we can do some kind of learning of these values by assigning some random values to this and iterating by minimizing the error based on the known rating. So this UV decomposition also probably we are going to see while talking about model-based collaborative filtering. Now when it comes to item-based collaborative filtering and compare with user-based collaborative filtering, item-based collaborative filtering does a lot of task in the offline page because this item data is basically available to the merchant. So therefore, available to the e-commerce site. So therefore, calculating the similarity offline is quite possible.



Whereas in case of user-based approach, when an active user comes and active user is not already identified, then the similarity has to be computed online as well, which will never happen in case of IBCF. So this similarity computation, which is a very computationally intensive task will be done in the offline and similarity matrix will be ready with the top k most similar items. So in the online page, it becomes very easy for prediction of the rating because you have to take the simple weighted average. So you take the weighted average based on this available similarity and the ratings, which the active user is giving to the items. So with this our discussion on item-based collaborative filtering is over.

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

Ratings (*M*) $\quad U \quad \Sigma \quad V^{T}$

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

Ratings (*M'*)

So once again, we are referring these two books and the examples etc that I considered and the formula are from these books only. So this is my concluding remark. This item-based collaborative filtering predicts the unknown rating based on item-based similarity. The steps involved here in are exactly same that of UBCF. Only the difference is you have to now compute the item-based similarity.

Now major computation in case of item-based collaborative filtering happen in offline manner. So therefore, this is a quite preferred way of providing recommendation in case of e-commerce applications. Thank you.