Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 04
Lecture – 19

Lecture 19: Basic latent factor models

Hello everyone. We are going to continue our discussion on model based collaborative filtering. And specifically in today's lecture, we are going to talk about basic latent factor models. So, these are the concepts to be covered. First we will talk about baseline predictor, then we will be talking about SVD enhanced baseline predictor, which also is popularly called as SVD latent factor model. Then we are going to talk about it SVD plus plus, where along with the rating matrix we will also be considering some kind of implicit feedback.

To start with collaborative filtering models, capture the interaction between user and item. So, how this interaction is captured by the rating matrix. So, to both model based as well as neighborhood based approach the rating matrix is the input. Now, from this rating matrix we tried finding out the similarities and made the models that we called as the neighborhood based models.

The model $b_{ui} = \mu + b_u + b_i$

μ is the overall average rating

b_u is the observed deviations of user u

b_i is the observed deviations of item i

Suppose $\mu = 3$ $b_u = 0.34$ $b_i = -0.23$

The predicted rating of item i by user u $b_{ui} = \mu + b_u + b_i$

$= 3+0.34-0.23=3.11$

The problem: How to determine the user bias and item bias

Then in model based approach instead of using some heuristics like that of creating the similarity matrix and so on, we tried building the models. So, that from that model itself when a new user comes and looks at a few other items, then we can provide some kind of rating prediction. Now when in one of the earlier lecture I have also told you that this rating matrix can have many kind of bias. Specifically, for users there will be bias, some users will be very high raters and some will be always giving the low values and over the time it may vary and the items themselves some items are likely to get higher rating

compared to the others. So, all these biases we are going to capture then we have to build models accordingly.

The first model that is popularly known as baseline predictor or baseline estimator goes like this. Here this is the rating predicted rating and mu is the mean of the overall rating. So, if you consider a typical rating matrix there will be many users, there will be many items and there will be specific rating values of even if it is not full it is sparse still then also there will be rating matrix there will be ratings. Now mu is the average of such ratings, total number of ratings observed some of that divided by the total number of rating. Now B u is the observed deviation with respect to user u and B i is the observed deviation with respect to item i.

$$\min_{b_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_u - b_i)^2 + \lambda_1 (\sum_u b_u^2 + \sum_i b_i^2)$$

Where

$\sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu + b_u + b_i)^2$ Find the biases that best fits the available ratings

$- \lambda_1 (\sum_u b_u^2 + \sum_i b_i^2)$ Regularization to avoid overfitting

Suppose mu is 3, B u is estimated so far we have not estimated. Assume that we have already estimated this value and B u is 0.34 and B i turns out to be minus 0.23. So therefore, the predicted rating as per this particular model will be 3.11. Now the next question is for mu it is fine we can take the average, but how do we estimate the biases with respect to user and the item. So that is the next question we are going to look at in our next slide. So, if our observed rating is R u i what is u? u is the user, i is the item and if we consider all the ratings which are visible to us which means all the pairs of u i for which rating is available let it belong to set k. So, set k contains what all the valid user item pairs for which the rating value is available. So, this was as per our model this was B u i.

$$b_i = \frac{\sum_{u\in R(i)} (r_{ui} - \mu)}{\lambda_2 + |R(i)|}$$

Then, for each user *u* we set

$$b_u = \frac{\sum_{i\in R(u)} (r_{ui} - \mu - b_i)}{\lambda_3 + |R(u)|}$$

So, R u minus this estimated value plus what is this term? This is the regularization parameter. So, this is the actual error with respect to the biases this is the regularization

parameter. Now we can solve this particular problem this optimization problem and find out the values of B u s and B i s. So, how many B u s we are going to find out? As the number of users in the system. How many B i s? As the number of items in the system.

- Lets set $\longrightarrow b_u = 0$
- Find $\longrightarrow b_i$

New error function: $E = \min \sum_{i \in k} (r_{ui} - \mu - b_i)^2 + \lambda_1 (\sum_i b_i^2)$

$$\frac{dE}{db_i} = \frac{d}{db_i}\left(\sum_{i \in k} (r_{ui} - \mu)^2 - 2(r_{ui} - \mu) b_i + b_i^2 + \lambda_1 \sum b_i^2 \right)$$

$$= \sum_{i \in k} -2(r_{ui} - \mu) + 2b_i + 2\lambda_1 b_i$$

Now, $\frac{dE}{db_i} = 0 \longrightarrow b_i = \frac{\sum_{\mu \in R(i)} (r_{ui} - \mu)}{\lambda_1 + |R(i)|}$

where, $R(i) \longrightarrow$ Set of users who rated item i

Similarly after replacing b_i,

$$b_u = \frac{\sum_{i \in R(u)} (r_{ui} - \mu - b_i)}{\lambda_2 + |R(u)|}$$

Now, look at this how do we solve this? To solve this problem this is the problem to solve this we have 2 how many unknowns we have total number of B u s are same as number of users and this is number of items. So, if number of users are m and number of items are n. So, total m plus m number of variables you have whose value you would like to determine. So, therefore, what will be the typical approach? You will be taking the partial derivatives with respect to each of this variable setting it to 0 you can do it together to run gradient descent algorithm or you can do it individually to run stochastic gradient descent. However, we can even simplify the situation further.

So, if you simplify the situation further then we are going to shortly see if we consider only if we assume all the B u s to be known then we get the model in terms of only B i and the B value of B i turns out to be this. How we are going how it turns out to be this that will be seen shortly. Then for each user u we can replace this value of B i over here and again solving it in terms of B u we can find it out. Now, let us move ahead and see how to solve it. Now, initially as I said we set all the B u s to be 0.

So, if we set all the B u s to be 0 then with respect to B i how many B i s are there? We still have many variables how many B i s? We still have n number of B i s because m users m number of B u s and n number of B i s. So, we now take the partial derivative with respect to one of the B i. So, if we take the partial derivative with respect to one of the B i then with respect to the other terms it naturally other terms will be treated as will not be considered they will be treated as constants and they will not be. So, here actually this would have been the partial derivative. So, you will be taking the partial derivatives with respect to each B i set them to 0.

So, when you set them to 0 with this one if you set to 0 B i is going to come to this side and this one is going to go to this side up and here we have two terms B i plus lambda B i of course, 2 is going to be going away because you are setting it to 0. So, B i so, lambda from here lambda is coming now this B i how many? So, this is same as that of summed over 1 with respect to this particular set k what set contains k contains the set k contains all the user item available user item pairs. So, with respect to i how many unique i's are there. So, that makes that number is this one. So, similarly once we find out this i's then we will be putting this B i's we will be putting with respect to the original equation original equation original equation is this one.

So, here we will assume mu is known B i's are estimated. Now value of B u this will also be known. So, now what remains this one this one and this one and where this B i values are already estimated. So, this we put this value and the moving going in the same manner that we did here we take the partial derivative of E with respect to B u a specific B u and from this we set to 0 and from this we determine just like we did it here similar manner we can find out the values of B u. Now in this baseline predictor what was the idea? The idea was rating reflects three things the average behavior of how users rate the items that is represented by mu then users bias and item bias.

Now here it in this new model it considers that not only user bias and item bias we can also include the latent factor corresponding to users and items. So, to remind you what are the latent factors? This concept of latent factor was coming from singular value decomposition, singular value decomposition, singular value decomposition which was simplifies to UV decomposition and from UV decomposition we know that U matrix was representing user factors and V matrix was representing item factors. So, now in this U and V matrix if we consider then a specific rating which is given by user U on item I this considers the factor corresponding to user U and the latent factor of user item I. Now what is user U? Suppose you consider certain k k number of factors. So, this P u basically is a vector P u 1, P u 2 up to P u k small k and Q i was again having the terms Q i 1 to Q i k.

So, here also k latent factors here also k latent factors and you multiply this two. So, this Q I mean all of if considering both of them to be column vectors here I wrote as row vectors. So, this is so considering them as column vectors you multiply Q transpose that is this one with P that is the corresponding column vector and if you multiply you get these terms and when we consider look this is the regularization term this is the regularization term for what for items how many items the total number of items available and how many users total number of users available here. So, now for this item I and U both these are vectors. So, when you take norm of this basically what you are doing individual elements you are individual elements you are taking the square and adding them up here also individual elements you are taking and squaring and adding them up.

So, this is the L 2 norm of these two vectors ok. So, now moving ahead what is the strategy now we are supposed to find out the solution to all these values of all these parameters how do we do it again we go for gradient descent or stochastic gradient descent in gradient descent all the elements at a time you will try to find out the partial derivative with respect to all the terms and in case of stochastic gradient descent one at a time. So, considering that we are going moving ahead with stochastic gradient descent let us just see with respect to if we would like to modify B i what is going to be the corresponding error term ok. So, this is our R U i R U i cap this R U i cap R U i hat is the estimated value and suppose we increment what is our M our M is if we start with certain random B i in the next iteration we will be making B i plus some delta B i right. So, let that delta B i be here.

So, this original equation with respect to now B i is getting incremented by this delta B i and moving ahead if we this is simplified and moving ahead when we take the partial derivative with respect to delta B i. So, delta B i turns out to be this. So, this simplification this because we are taking it with respect to delta B i rest of the terms are going to be assumed to be constant. So, naturally derivative with respect to them will be will become 0. So, automatically you will be getting this is gone this is gone B U is gone only this B i will remain because delta B i is getting multiplied here.

So, consequently you have this in which with delta B i this is also there and what is this term what is this term this is the error term this is the error term where from the error term error term is this ok. So, though we have seen it we did it for only this one. So, similarly these parameter update equations for rest of the parameters can be determined in this manner. So, how many such values will be there? There will be m such values and here n such B i s and here how many n number of items with k features each this many parameters and here how many m number of users into k number of features ok. So, that is how you will be and every time in the first iteration this was the value this was the estimated value this difference turns out to be the error.

Second iteration again this will be the you will be re estimating this error. So, you will keep on updating. So, you start with certain random values of this parameter and you keep updating following these equations and follow this stochasticgradient descent algorithm. In fact, this stochastic gradient algorithms last lecture we have seen with respect to only latent factor models where this biases were not considered. Now we extend this idea of SVD what was SVD in this context SVD was that baseline predictor where latent factors were also considered that was popularly in is is called as the SVD based model SVD based latent factor model.

So, that model we are now going to extend how will you extend it? So, we will extend it by considering certain implicit feedback. If you remember what is implicit feedback? Rating is something which the user explicitly gives and implicit feedback is something

which has to be collected by observing the users activity. So, it is here one very nice ah actually ah before in the beginning I should have told you if you remember in one of the earlier lecture we were talking about the Netflix competition. So, all these models were developed by a team from the Bell lab and those three team those team of three people three researchers they tried developing all these models. So, as we move ahead starting from baseline then the SVD SVD plus plus and so on this is basically mostly the contribution of these three researchers.

The model

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

The error minimization problem

$$\min_{b_*, q_*, p_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda_4 (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2)$$

So, ah in SVD plus plus what we are going to do we are going to consider this implicit feedback matrix. Now this implicit feedback matrix can be generated by observing the behavior in the absence of such external implicit feedback you can get it from the user rating. So, it can be derived from user rating. So, if you do that the model turns out to be like this. Item feature remains same, but users model users feature model latent feature model gets enhanced by this was the original P U and this is the additional term this is the additional term ok.

$$\min_{b_*, q_*, p_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda_4 (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2)$$

$e_{ui} = (r_{ui} - \hat{r}_{ui})$ Suppose bi is incremented by a small amount

$$E = \left(r_{ui} - \mu - (b_i + \Delta b_i) - b_u - p_u q_i^T\right)^2 + \lambda_4 \left(b_u^2 + (b_i + \Delta b_i)^2 + \|p_u\|^2 + \|q_i\|^2\right)$$

$$= \left(r_{ui} - \hat{r}ui - \Delta b_i\right)^2 + \lambda_4 \left(b_u^2 + \|p_u\|^2 + \|q_i\|^2 + (\Delta b_i)^2 + 2 b_i \Delta b_i\right)$$

$$\frac{\partial E}{\partial \Delta b_i} = 0 \longrightarrow \Delta b_i = \frac{1}{(1-\lambda_4)} \left(e_{ui} - \lambda_4 b_i\right)$$

$$= \gamma \left(e_{ui} - \lambda_4 b_i\right)$$

So, here in this term y i is the factor vector rating for jth item user ah for the jth item from users perspective and R U is the set of items rated by user U and R this is the total number this is the set this is the number of elements in that set. Now the question is why

do we get it in this form. So, this you have to remember because in one of in few next lecture also we are going to encounter with this term. This is one example how SVD plus plus is derived is we can derive implicit feedback from the rating matrix. Suppose this is our rating matrix wherever there is question mark those places are blank rating is not available.

Now we can construct a feedback implicit feedback matrix from this. So, this feedback concept behind this feedback matrix F goes like this. If user has given some rating then let it be 1 at least he has considered giving rating and if the rating is not available then make it 0. So, that is how here 1 rating here it is minus 1 here it is 1 here it is 1 here it is 2 only this place rating was not available.

So, we make it 0. So, we repeat the same thing for all the rows. So, after we repeat all the rows then we try making the normalized feedback matrix. How do we normalize it? Take all the elements over here take the square take the root over and divide that. So, it is normalized. So, that L 2 norm of each row is 1 L 2 norm of each row remains 1.

- $b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda_4 \cdot b_u)$
- $b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda_4 \cdot b_i)$
- $q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda_4 \cdot q_i)$
- $p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda_4 \cdot p_u)$

Where

$$e_{ui} = r_{ui} - \hat{r}_{ui}$$

So, we divide this by this is 1 2 3 4 5. So, some square of all the ones square of all the ones is 1 again and take the root over that that makes it root 5. So, root 5 is divided everywhere. So, also the root 2 here and root 3 here and we want to implicit feedback from this. So, this is our y value and we multiply it with the feedback matrix with this variable y and we get this matrix. Now going back look this was the average value of each has to be with respect to each u we are supposed to do it.

So, you take this sum and take this sum divided by average take the sum divided by average and so on. So, the same manner in which we divide we derived the update equation for SVD. Moving ahead with this how many unknowns b i b u p i p u b i m this is n this is how many this is n cross k this is m cross k and this is also m cross k ok 2 vectors are getting added. So, element wise element they will be getting added ok. So, this many parameters and with respect to the way we did this one for each of the variables we are supposed to now find out the updates.

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j \right)$$

So, once you find out the updates you will be getting this and you will again run stochastic gradient descent to determine start with certain random value and keep on iterating till your final criteria stopping criteria is met. Now that was the feedback which was getting generated from that rating matrix and from rating matrix why it was called the feedback from the rating matrix because our observation we made was whenever the rating was given which means user is interested get give it 1 if the rating is not given give it 0 and continuing in the same manner if certain external rating matrix is external implicit feedback is available and treating it in a 1 0 manner you will be getting this in that that case it was a number of this value here. This value here it was getting normalized by number of rating available on with respect to that particular item by user u here in this context it is no more part of that no more derived from that rating matrix. So, this in number of it has no it has to be now depend it has to now depend on the number of observations which is available from user u this is not from the rating matrix that is why that was the actually the number of actual rating given by the user u this is actually observed from another source. These are the references as usual and all this content is derived from here with this we wind up these are our conclusions a typical collaborative filtering data exhibits large user and item biases.

$$\begin{bmatrix} 1 & -1 & 1 & ? & 1 & 2 \\ ? & ? & -2 & ? & -1 & ? \\ 0 & ? & ? & ? & ? & ? \\ -1 & 2 & -2 & ? & ? & ? \end{bmatrix}$$

The feedback matrix *R*

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The feedback matrix *F*

The feedback matrix *F* is subsequently normalized so that the L_2-norm of each row is 1

$$\begin{bmatrix} 1 & 1/\sqrt{5} & 1/\sqrt{5} & 0 & 1/\sqrt{5} & 1/\sqrt{5} \\ 0 & 0 & 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \\ y_{31} & y_{32} \\ y_{41} & y_{42} \\ y_{51} & y_{52} \\ y_{61} & y_{62} \end{bmatrix} = \begin{bmatrix} \sum_i y_{i1}/\sqrt{5} & \sum_i y_{i2}/\sqrt{5} \\ \sum_i y_{i1}/\sqrt{2} & \sum_i y_{i1}/\sqrt{2} \\ \sum_i y_{i1}/1 & \sum_i y_{i1}/1 \\ \sum_i y_{i1}/\sqrt{3} & \sum_i y_{i2}/\sqrt{3} \end{bmatrix}$$

- $b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda_5 \cdot b_u)$
- $b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda_5 \cdot b_i)$
- $q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot (p_u + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{R}(u)} y_j) - \lambda_6 \cdot q_i)$
- $p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda_6 \cdot p_u)$
- $\forall j \in \mathrm{R}(u)$:
  $y_j \leftarrow y_j + \gamma \cdot (e_{ui} \cdot |\mathrm{R}(u)|^{-\frac{1}{2}} \cdot q_i - \lambda_6 \cdot y_j)$

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left(p_u + |\mathrm{N}^1(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}^1(u)} y_j^{(1)} + |\mathrm{N}^2(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}^2(u)} y_j^{(2)} \right)$$

Baseline predictor models these biases these baseline predictors can be improved by adopting user and item latent features conceptualized from your singular value decomposition and this SVD based model enhances this baseline model and it can be further enhanced by considering implicit feedback. And this implicit feedback can be derived from the rating matrix if it is not available from the external sources and if it is available then it can be added to improve the model performance. Now, these model parameters can be learned using the stochastic gradient descent for this purpose we have to take the error function find out the partial derivatives with respect to each of the ah variable and develop certain equations. So, which equations can be used to iteratively update this value starting from certain random values. Thank you.