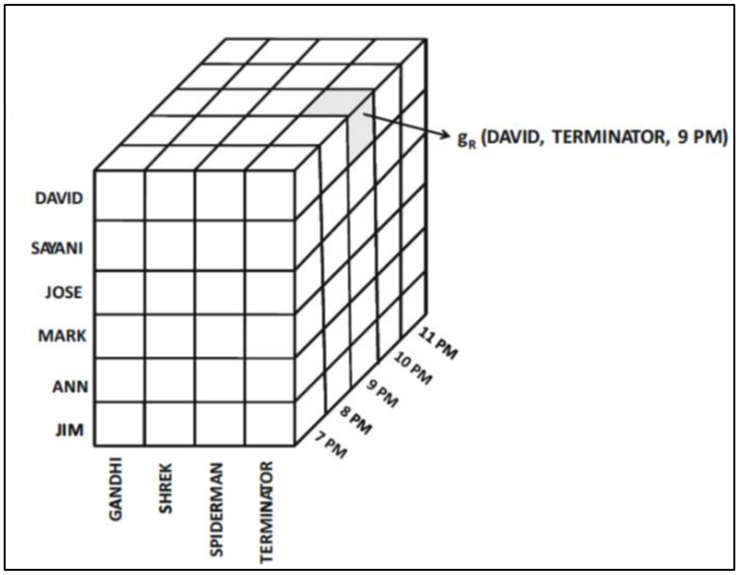
Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 08
Lecture - 38

Lecture 38: Context-Sensitive recommender systems

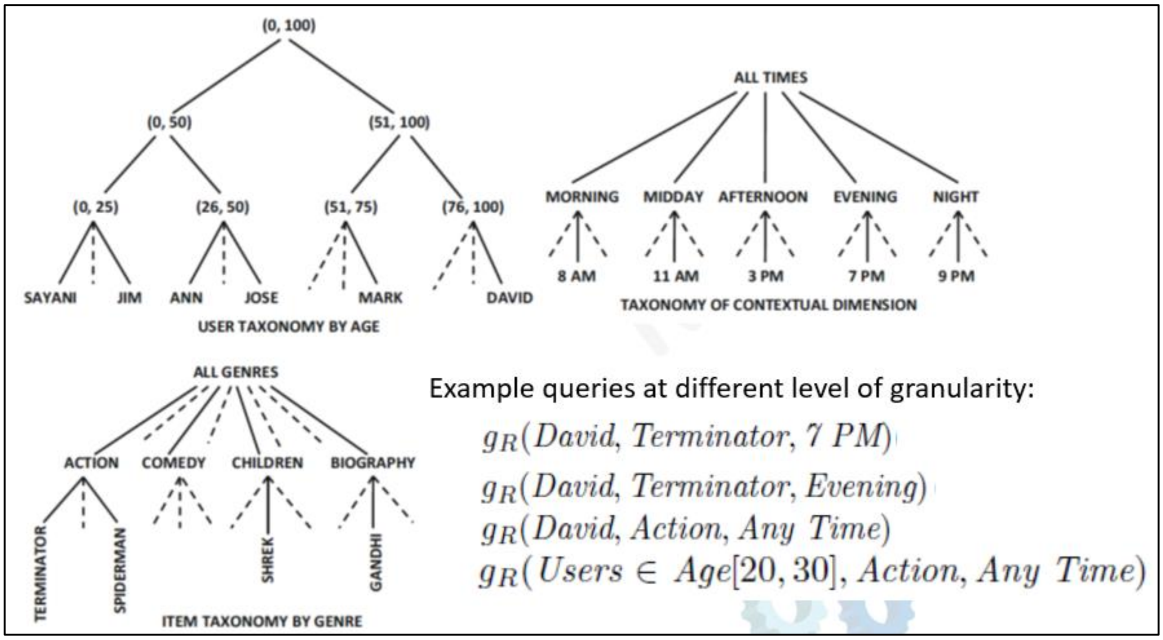
Hello everyone. Welcome to the 38th lecture of this series and we are talking about other types of recommender system. In this context, we have already seen how to make hybrid recommender systems and we have started talking about how to use different other sources of information. And we saw how to use additional knowledge which can be represented in terms of rules or in terms of cases. So today we are going to talk about a new type of recommender system which is called context sensitive recommender system. So to start with, let us try to understand what a context is.

A typical example of context can be time, location or social information. For example, if you are visiting certain place and you are searching for eatery, you will be shown, you will be recommended with the eateries in the nearby location, not from a far off place. So the recommendation that you got is actually context aware with respect to location. Similarly, with respect to time, may be if you are looking searching for clothings, you will be looking at if in winter if you are looking at the clothings, you will be getting something which you will not be getting in summer.



So what this context sensitive recommender systems do, they tailor their recommendations with additional informations that makes sense in a particular situation. So let us look at how this kind of system is different from the kind of system we have talked so far. So far we have discussed about content based and collaborative filtering. Content based and collaborative filtering. And what was the information source there? We were talking about a rating matrix and we had users, we had items.

So that is how we got users here, items here and it was a 2D matrix. Now along with the users here and items here, in case of a movie recommender system these are few users, these are few items. We have also collected the data about the time and what are these individual boxes? They provide the ratings. So as usual in rating matrix we used to have very sparse data. So which means users will not be, all the users will not be rating all the items.



Example queries at different level of granularity:

$g_R(David, Terminator, 7\ PM)$

$g_R(David, Terminator, Evening)$

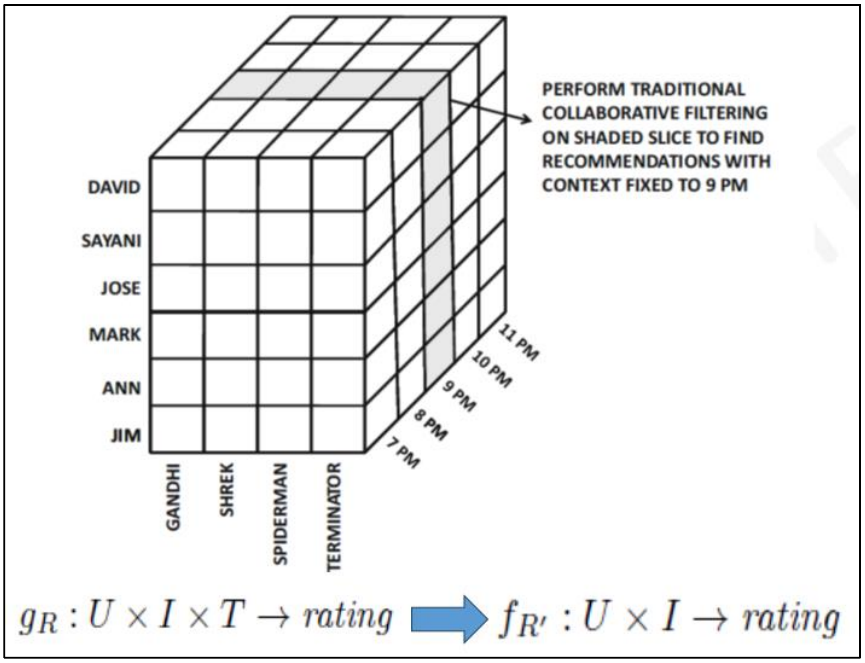
$g_R(David, Action, Any\ Time)$

$g_R(Users \in Age[20, 30], Action, Any\ Time)$

Now the situation is even worse. Worse in the sense, now this rating which we would have without this we would have merged together without this time dimension would have merged together. Now this is expanded along time. So as a result this is a more sparse situation. However, we are, we will be with this we will be able to give the predictions, the recommendations based on certain additional context.

So these are users, these are users, items and some additional context and this case it is time. So with time dimension now instead of matrix now what we have got is a 3 dimensional matrix which otherwise called be a tensor. And if we increase the dimension further let us say along with the time we also put, we also put place, location then this will become a 3 dimensional, 4 dimensional object. And if we additionally put something else let us say social information it will become a 5 dimensional object and so on. So now you will be mapping, initially you were mapping this users and items to rating that is how you were getting a matrix.

Now all this w dimensions you are going to map to rating. So you have now a sparse yet a more context aware data set. If you look at this, this system actually is a generalized view of the online analytical processing data queue. Now what is this online analytical processing data queue? If you study about data mining and data warehousing, you must be aware of such kind of systems. Now people who are not aware let me tell you in case of all of you at least know the database management system, the relational database management system where you have the tables.

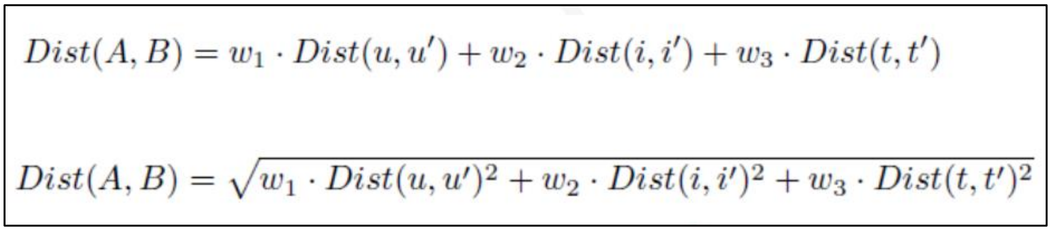

PERFORM TRADITIONAL COLLABORATIVE FILTERING ON SHADED SLICE TO FIND RECOMMENDATIONS WITH CONTEXT FIXED TO 9 PM
DAVID
SAYANI
JOSE
MARK
ANN
JIM
GANDHI
SHREK
SPIDERMAN
TERMINATOR
7 PM
8 PM
9 PM
10 PM
11 PM
$g_R : U \times I \times T \to rating$
$f_{R'} : U \times I \to rating$

The tables will have the rows which indicate the entities and you will have the columns which shows the attributes. So entities and their attributes together make a table. Now if we add more we can represent it in a multidimensional setting in the sense let us say we are collecting the sales data. We can collect the sales data for products per sales person for different locations over different financial years and so on. And you can do in depth analysis with that.

You can think of digging down the data, you can think of compressing the data and rolling up and dig down and do many other processing which otherwise would have been difficult in case of relational database management system. But anyway that is beyond the scope of our discussion we are not going to talk about them. So, there are many operations slicing, dicing and so on. So, we will not be talking about them. But this is quite the conceptually this is quite similar.

Now when we talk about the context sometimes it is possible that we can make a hierarchy of contextual dimensions and even if this hierarchy we do not have to be put as a part of the data cube that just now we saw. We can otherwise use this hierarchy to provide to answer the queries at different level of granularity. For example, here the time is represented let us say as morning, midday, afternoon, evening and night. And the time ranges are different with here the mean is 8, 8 AM here mean is 9 AM and so on. Suppose you have collected the data at these levels.

Now morning you have collected the data at each hour suppose for morning you say this is from let us say 6 o'clock, 6, 7, 8 then let us say 9 up to 10 you say that is your morning hour. So, when you try giving the recommendation for example, here suppose you are trying to give the see 7, 8, 9 let us say up to 7, 8, 9 these 3 considers morning. And the query is on debit that particular user particular movie and 7 PM. Now if we have to provide the recommendation for this particular time and we know this contextual hierarchy then if sufficient data is not available around 7 then maybe 6 look here 7, 8, 9 we can use the data given in 8 as well or maybe if we have 6 PM, 6 PM as well to make the recommendation. So, which means whatever spans evening which is up in the hierarchy up in the hierarchy here up in the hierarchy here we can use that as our for increasing our data.



$$Dist(A,B) = w_1 \cdot Dist(u,u') + w_2 \cdot Dist(i,i') + w_3 \cdot Dist(t,t')$$

$$Dist(A,B) = \sqrt{w_1 \cdot Dist(u,u')^2 + w_2 \cdot Dist(i,i')^2 + w_3 \cdot Dist(t,t')^2}$$

We can even make it denser in the sense we can talk about action movies action movies terminator is one kind of action movie and any time. So, which means you are actually considering the entire span of time. Similarly, you can also squeeze your search to a higher level and search for the users at a particular age group. Now, this multilevel multidimensional rating estimation problem where we have the data rating data represented in the form of some higher dimensional data cube. We can solve this using 3 approaches contextual pre filtering, contextual post filtering and contextual modeling.

In case of contextual pre and post filtering they are basically the approach which is similar to our other approaches where we use to have a 2 dimensional rating matrix. They somehow make arrangements to make this higher dimensional rating matrix squeeze into a 2 dimensional matrix and based on that they take the decision. As the name indicates in case of pre filtering they take each slice of this each slice of this, this slice if we take slice of the data cube you get for the let us say for 9 p m you get only users and items. So, based on that based on that 2 dimensional thing they try first taking the decision then slowly incorporate higher dimensions. In case of post filtering they try to merge this together I mean they do not consider they remove all the context.

So, that all the ratings put together are squeezed in the form of one matrix and in that matrix with respect to that matrix which is just like of your ordinary utility matrix or rating matrix you take the you provide the recommendation. And when you after you provide the recommendation you give some kind of heuristic rules to now add context to this scenario. But contextual modeling is different it actually works on this high dimensional data. So, this is what I was talking about contextual pre filtering. So, what you do here? If you have 3 dimensions which maps to rating, you reduce it to 2 dimensions and map to rating.

And I was as I was telling you consider a slice you consider a slice of this. And this slice relates to only the it becomes actually only to the context which is 9 p m 9 p m context. So, basically you are not taking care of the context at all you are considering you it as your rating matrix 2 dimensional rating matrix. So, the traditional collaborative filtering system you can apply, but only problem here is your data will be very sparse as such data rating matrix is sparse. Now you are considering a rating matrix which belongs to only 9 p m.

So, which means it is even more sparse. So, now to reduce sparsity you can adopt few approaches. One which is called the first one is called local approach. In this local approach what you do? You look for the ratings in the nearby time interval let us say in 8 and 10 p m. So, these 3 slices together now you compress.

So, you have now a 2 dimensional data again where the time for the ratings for 8 p m 9 p m as well as 10 p m are merged together. So, because the sparsity is reduced as a result you may get more data and you can accurately predict the rating. But in case of global approach you actually compress this entire thing this whole thing you press together and get one rating matrix. So, if you get one rating matrix of course, it will be denser than the local ah approach, but what happens it is extremely generalized and it will not be providing any context related it will not bring any context related ah recommendation. In case of contextual post filtering actually we start with a global model.

Global model in the sense compress that entire thing to a 2 dimensional matrix. So, with this global model what you do? First you use conventional collaborative filtering or whatever to provide the recommendation as the aggregate user level, then you introduce context to adjust and filter the recommended list. So, how do we do this ah filtering or adjustment? For example, suppose you have found out irrespective of the time you have given certain recommendation let us say for clothes. Now, ah if in this list now tops with let us say some sweaters and heavy jackets in the context of summer. So, which means externally you know what time now ah is what kind of season now is.

So, if you know it is summer then you really do not have to now ah you have to purge out those sweaters and heavy jackets which are not relevant in this context. So, therefore, you have to adopt some kind of heuristics which depends on the context. Then the third approach in this regard is contextual modeling. Now as we know both pre filtering and post filtering we actually use a 2 dimensional ah approach and use traditional collaborative filtering algorithms. But in this case we really have to play with that bigger w dimensional data ok.

So, this contextual modeling again can be done in 3 ways neighborhood based method, pattern factor models and content based models. So, let us just have a look at them. Come to this contextual modeling with neighborhood based approach. Now consider 2 points in a 3 dimensional queue neighborhood. So, you have to you you you already know what a neighborhood is.

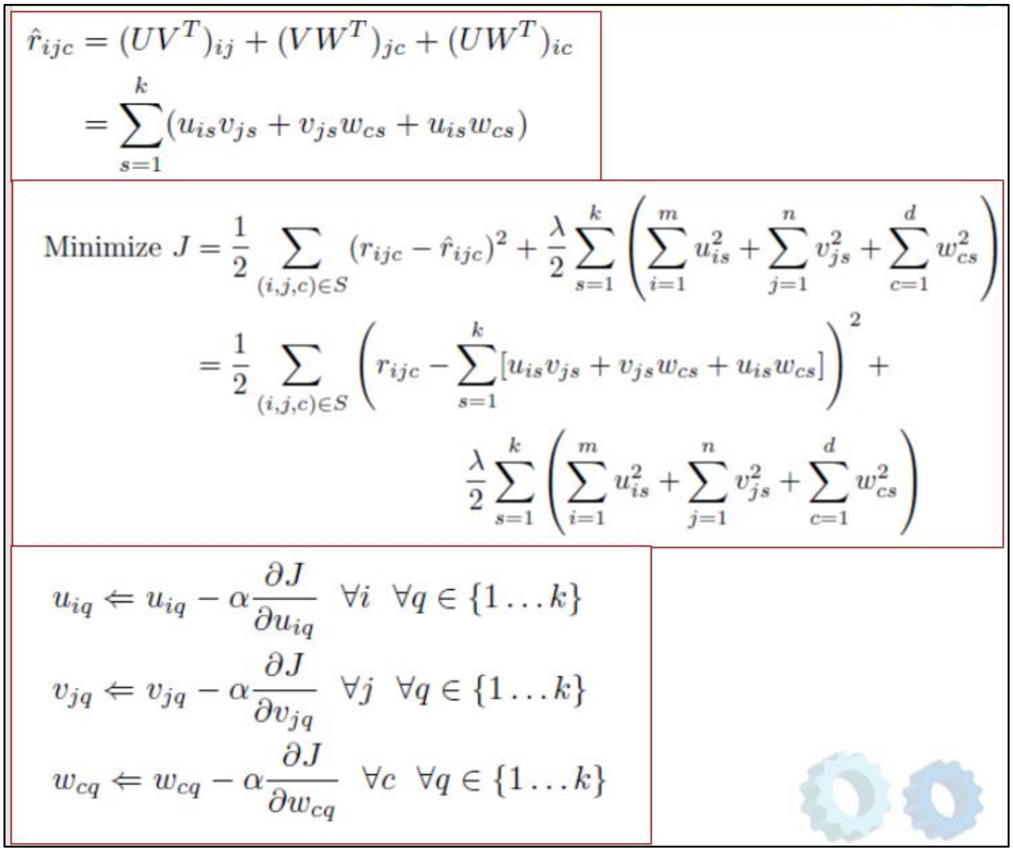
So, with respect to ah collaborative filtering you know what is a neighborhood based approach in case of KNN we we studied what is a neighborhood this is basically found out using the neighborhood is found out using the similarity or distance function ok. So, in case of similarity functions if you are using a similarity function to find out the neighborhood higher the similarity closer is the neighbor. If you are using the distance it is just opposite lower the distance closer is the neighbor ok. So, ah you can use some distance function which is actually weighted these are the weights. So, this weighted distance between individual dimensions ok.

So, based on dimension u user dimension based on item dimension based on distance time dimension you find out the distance and you take the distance between a and b as the weighted average of this. This can be another distance function the second one is another distance function. Now this distance may be determined using any of the following approach. You can use collaborative filtering kind of approach where we can extract 2 dimensional slices corresponding to user u and user u dash and compute. In case of content based approach we take the user or item features which are not part of this w dimensional data or 3 dimensional matrix in a in this case.

So, from there so that is one additional extra source from which you are finding out the similarity. Now, for a given cell of 3 dimensional matrix closest r observed ratings are determined by using this distance function. Weighted average of these ratings are called are are used as the predicted rating. Now the weighting used in this similarity ah is the similarity between a and b in this a and b in order to ah in order to perform the recommendation for a given user u and the context t one would need to keep the process for each item and then report the top k items as the recommendation.

Then we have latent factor models. In case of latent factor model with rating matrix with 2 dimensions we use certain matrix factorization. Now, the structure that we are dealing with a multidimensional matrix which is otherwise called a tensor. Now with respect to this tensor we can also have equivalent factorization models. So consequently, so in a factorization model let us say SVD what you have? You have u matrix and v matrix which where you were saying that u represented users features and v represented item features. Now you have to have something called context feature.

So as a result those u v and w which were 2 dimensional themselves now will become 3 dimensional. So as a result you are again going to land up in a very complex model. So therefore, you can simplify it by considering only pair wise interaction between different dimensions. So as a result you will land up in a matrix form. So u is the user factor, v is the item factor, w is the context factor.



$$\hat{r}_{ijc} = (UV^T)_{ij} + (VW^T)_{jc} + (UW^T)_{ic}$$
$$= \sum_{s=1}^{k} (u_{is}v_{js} + v_{js}w_{cs} + u_{is}w_{cs})$$

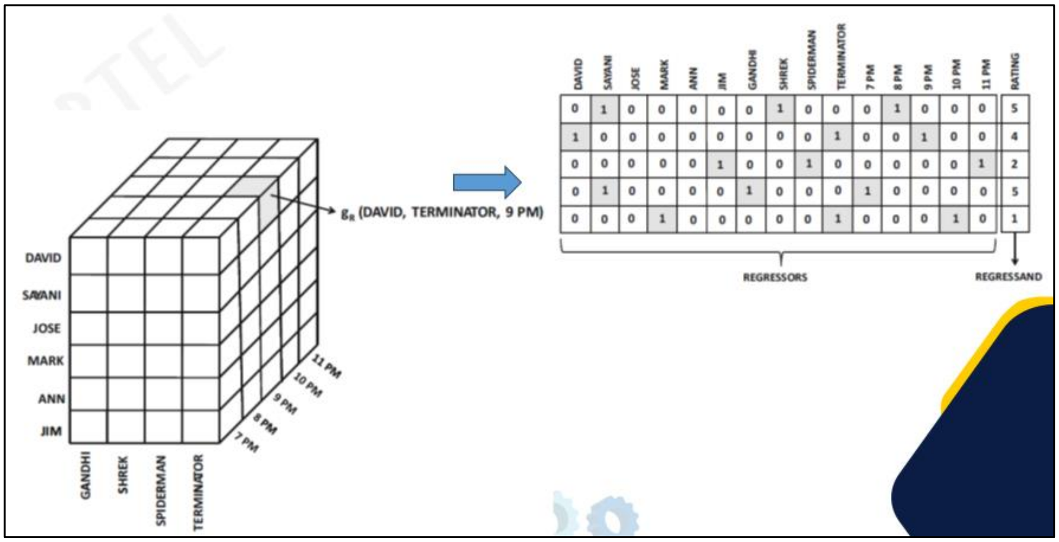
$$\text{Minimize } J = \frac{1}{2} \sum_{(i,j,c)\in S} (r_{ijc} - \hat{r}_{ijc})^2 + \frac{\lambda}{2} \sum_{s=1}^{k} \left(\sum_{i=1}^{m} u_{is}^2 + \sum_{j=1}^{n} v_{js}^2 + \sum_{c=1}^{d} w_{cs}^2 \right)$$
$$= \frac{1}{2} \sum_{(i,j,c)\in S} \left(r_{ijc} - \sum_{s=1}^{k} [u_{is}v_{js} + v_{js}w_{cs} + u_{is}w_{cs}] \right)^2 +$$
$$\frac{\lambda}{2} \sum_{s=1}^{k} \left(\sum_{i=1}^{m} u_{is}^2 + \sum_{j=1}^{n} v_{js}^2 + \sum_{c=1}^{d} w_{cs}^2 \right)$$

$$u_{iq} \Leftarrow u_{iq} - \alpha \frac{\partial J}{\partial u_{iq}} \quad \forall i \quad \forall q \in \{1 \dots k\}$$
$$v_{jq} \Leftarrow v_{jq} - \alpha \frac{\partial J}{\partial v_{jq}} \quad \forall j \quad \forall q \in \{1 \dots k\}$$
$$w_{cq} \Leftarrow w_{cq} - \alpha \frac{\partial J}{\partial w_{cq}} \quad \forall c \quad \forall q \in \{1 \dots k\}$$

So user item context and user context taken together. So you make your rating model in this manner. Here what you are supposed to discover? You are supposed to discover this u i, v, u, v and w individual elements from the known rating. So because if you remember the latent factor model this SVD, SVD plus plus etcetera what we did? Few ratings were known, we build a model and we tried training with respect to those available rating and we were predicting and build a model and predicted the rating for the cells where the rating values were missing. So here also we will be now subtracting this add regularization terms over here and it can be maybe it can be you can simplify it if you check you can simplify it further in this manner and you develop using taking the partial derivative etcetera that we have discussed already in earlier lectures that the process is same.

You develop some equations to update the parameters. So you have k attributes, k attributes sorry k dimensions in each case. You now start with certain random thing, random values and keep on updating using this update using this kind of update equation. So now with respect to S the set of all observed ratings in R that is this S all observed ratings you build the model. Once you decide the model parameters you can predict ratings with respect to any of the values.

The second approach here is factorization ratings factorization machines. So try to remember what are we trying to do we tried talking about contextual modeling. So there were three approaches prefiltering, post filtering and contextual modeling for dealing with context aware system. And when we are talking about contextual modeling under contextual modeling we had three methods neighborhood based method which we already looked at. Latent factor models we looked at now we have to consider the another version of this latent factor model which we call as factorization machines.



DAVID	SAYANI	JOSE	MARK	ANN	JIM	GANDHI	SHREK	SPIDERMAN	TERMINATOR	7 PM	8 PM	9 PM	10 PM	11 PM	RATING
0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	5
1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	4
0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	2
0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	5
0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1

REGRESSORS

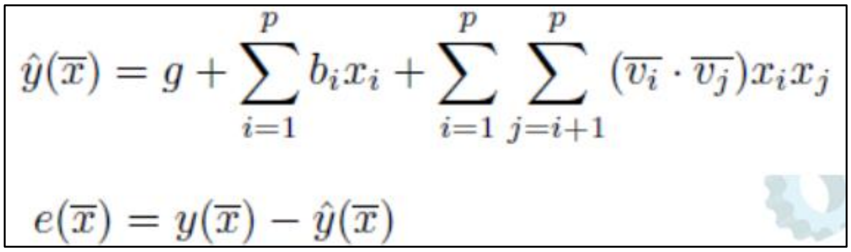
REGRESSAND

So in case of factorization machines the idea is to build a linear model. So this is also a model based approach, but we build a linear model instead of considering those factors sorry I told that is factor based. So it is not a latent factor model, but here we build some kind of learning model. So how do we make this learning model? What we do? We make a two dimensional regression problem out of this multiple dimensional problem. So what do we do? Here what is known here what we are supposed to predict is the rating.

And rating is available in only few places. Those few places are characterized by user item and time. Now this is the set of users, this is the set of items, this is the level time level in the context time context the various times on during which the observation is taken. Now just like we do one hot encoding suppose some rating 5 appears with respect to this person, this movie, this person, this movie and this time. So only in three places we have 1 in rest of the places we have 0. In the second case in three places we have 1 which represents the context rest of the places it is 0.

So that is how in a two dimensional form we bring the multidimensional matrix. And because of this is a quite straightforward approach for representing using some kind of one hot kind of encoding. And now based on the once our data set is ready we can use any kind of classifier or prediction model any kind of learning model supervised learning model. So they are basically multiple linear regression models, but because there are relationships between the contextual terms we provide some kind of additional additional factors. So you remember linear regression there is to be some coefficient y cap used to be some coefficient plus beta 1 x 1, beta 2 x 2 and so on.

So here up to this it is that and here you have interaction terms. Now using this interaction see when we limit here we have limited our interaction to between two factors, but it may so happen that we can have higher order interaction as well with multiple factors. But for simplicity after all it is a model. So for simplicity we can keep it up to two factors. So now what is the problem? The problem is to determine all this g bi is v i bar and v j bar.



$$\hat{y}(\overline{x}) = g + \sum_{i=1}^{p} b_i x_i + \sum_{i=1}^{p} \sum_{j=i+1}^{p} (\overline{v_i} \cdot \overline{v_j}) x_i x_j$$

$$e(\overline{x}) = y(\overline{x}) - \hat{y}(\overline{x})$$

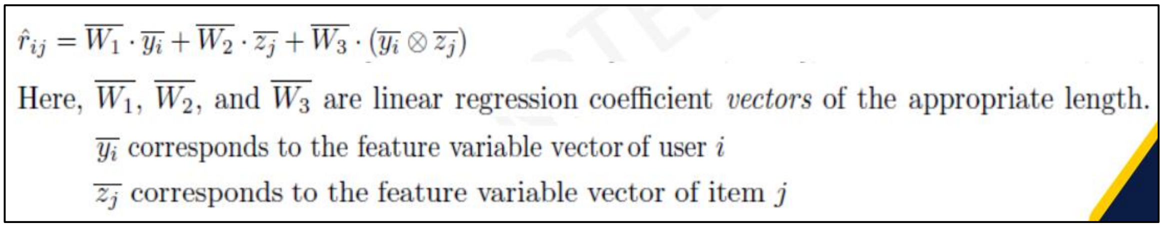
And how do we get these model parameters with respect to available rating? And here these are the available rating and we are supposed to trying to build a model using all these regressors. So this is the error function and we try minimizing this error function. So when we try minimizing this error function we have to we are leaving the steps you can see this whole thing in this whole week the last module everything I have taken from the book from Charu Agarwal you can refer the detailed derivation is also there. But because it is similar with whatever we did earlier so therefore we will not be repeating it. We will be taking the partial derivative and all and few more simplification steps.

So with respect to each parameter theta what is this theta? This theta can be g any of the bi's then v i v i bar v j bar. So basically both are same only v i bars and so on. So this g v i and v i bar. So with respect this theta represents that only it can be either any one of this. So now this how did we arrive at this? Following the same approach minimize that function take the partial derivative with respect to individual parameters come up with a come up with an equation like this one.

$$\hat{y}(\overline{x}) = g + \sum_{i=1}^{p} b_i x_i + \sum_{i=1}^{p} \sum_{j=i+1}^{p} (\overline{v_i} \cdot \overline{v_j}) x_i x_j \qquad e(\overline{x}) = y(\overline{x}) - \hat{y}(\overline{x})$$

$$\theta \Leftarrow \theta(1 - \alpha \cdot \lambda) + \alpha \cdot e(\overline{x}) \frac{\partial \hat{y}(\overline{x})}{\partial \theta}$$

The last one is your content based model. In case of content based model what are we trying to do? We are trying to use the a similar approach like that of your regression model. In a very simple case when we do not have any complex we can make a content based model in this way. R i j hat which is the predicted rating can be represented in this form where y i j represents corresponds to item feature variable vector for user i, z j corresponds to the feature vector variable for item j this is user this is item and w 1 w 2 are corresponding weights. Now here is a catch this particular term is the is called the kronecker cross product. So these are the interaction terms between user and item and this is typically represented through this matrix in this form this.



$\hat{r}_{ij} = \overline{W_1} \cdot \overline{y_i} + \overline{W_2} \cdot \overline{z_j} + \overline{W_3} \cdot (\overline{y_i} \otimes \overline{z_j})$

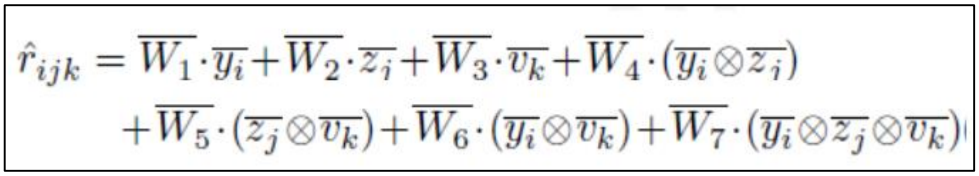
Here, $\overline{W_1}$, $\overline{W_2}$, and $\overline{W_3}$ are linear regression coefficient *vectors* of the appropriate length.

$\overline{y_i}$ corresponds to the feature variable vector of user i

$\overline{z_j}$ corresponds to the feature variable vector of item j

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

So all these terms here will appear with this weight. So now once we have this what is to be learnt? So we are supposed to learn these three things because these values will be available for known ratings. So once we know these three coefficients w 1 w 2 and w 3 bar then for unknown ratings we can find out. Now in case there is context we have let us say we have only one context that context also has to go already this part was there and this part was there. Now in addition this has come this part has this part has come and this are here. So interaction between user and context item and context and user item and context this is again going up to the interaction with all the three variables taken together.



$$\begin{aligned}\hat{r}_{ijk} &= \overline{W_1}\cdot\overline{y_i}+\overline{W_2}\cdot\overline{z_j}+\overline{W_3}\cdot\overline{v_k}+\overline{W_4}\cdot(\overline{y_i}\otimes\overline{z_j})\\ &\quad+\overline{W_5}\cdot(\overline{z_j}\otimes\overline{v_k})+\overline{W_6}\cdot(\overline{y_i}\otimes\overline{v_k})+\overline{W_7}\cdot(\overline{y_i}\otimes\overline{z_j}\otimes\overline{v_k})\end{aligned}$$

Even if you think to simplify you may think of dropping this one but once we have this what next? Take everywhere r i j k minus r i j k cap this makes your error term. Now minimize error to minimize the error again follow the same approach to determine these values and use again use stochastic gradient descent etcetera we do not have to repeat them those things. So references are this mostly the error it is taken from Charvagarwal completely taken from Charvagarwal. So these are our conclusions context sensitive recommender systems tailor their recommendations to additional information that defines the specific situations under which recommendations are made. Multi level multi dimensional rating estimation problem can be solved using three approaches contextual pre filtering contextual post filtering or contextual modeling.

Now both pre and post filtering approach are boiled down to traditional collaborative problem with two dimensional setting. Whereas in case of the last approach that is contextual modeling you can have many ways to solve it neighborhood based method latent factor based method here of course did not write we have this factorization machines and we have content based model. Thank you.