Context-Sensitive Recommender Systems

IT492: Recommendation Systems

DA-IICT

April 21, 2022

Context-Sensitive Recommender Systems (CSRS)

Context: Context are those variables which may change when a same activity is performed again and again.

- Watching a movie: time, location ,companion,etc
- Listening to music: time, location,emotion, occasions,etc
- Restaurant: time, location, ocassion,etc
- Travels: time, location, weather, transportation conditions,etc

Context-Sensitive RecSys

- Traditional RecSys: Users X Items $\rightarrow$ Ratings
- Contextual RecSys: Users X Items X Contexts $\rightarrow$ Ratings

User	Item	Rating	Time	Location	Companion
U1	T1	3	Weekend	Home	Kids
U1	T2	5	Weekday	Home	Partner
U2	T2	2	Weekend	Cinema	Partner
U2	T3	3	Weekday	Cinema	Family
U1	T3	?	Weekend	Cinema	Kids

Multi-dimensional Context-Sensitive Dataset

User	Item	Rating	Time	Location	Companion
U1	T1	3	Weekend	Home	Kids
U1	T2	5	Weekday	Home	Partner
U2	T2	2	Weekend	Cinema	Partner
U2	T3	3	Weekday	Cinema	Family
U1	T3	?	Weekend	Cinema	Kids

- Context Dimension: time, location, companion
- Context Condition: Weekend/Weekday, Home/Cinema
- Context Situation: weekends, home, kids

The Multidimensional Approach

Traditional problem of recommendations

- $f_R : U * I -> rating$

The Multidimensional Approach

- $g_R : D_1 * D_2 * .... * D_w -> rating$

The Multidimensional Approach

- $g_R : D_1 * D_2 * .... * D_w - > rating$
- Two of these dimensions will always be user and Item.
- Note that the context can be a property of the user, a property of the item, a property of the user-item combination, or a completely independent property.
- It is not important who the context is related to, because it is treated as a completely independent entity from the user or the item.

The Multidimensional Approach

More Generic Approach

- Use of two disjoint subsets from D1 ...Dw
- The selected subsets of dimensions in D1 ...Dw are either "what" dimensions, or they are "for whom" dimensions.
- In traditional recommender systems, what = Item, for Whom = User

Definition 8.2.1 (Multidimensional Recommendations) *Given the recommendation space* $D_1 \times D_2 \times \dots D_w$ *and the rating function* $g_R : D_1 \times D_2 \dots \times D_w \rightarrow \text{rating}$, *the recommendation problem is defined by selecting certain "what" dimensions* $D_{i_1} \dots D_{i_p}$ *and certain "for whom" dimensions* $D_{j_1} \dots D_{j_q}$ *that do not overlap, and recommending for a query tuple* $(d_{j_1} \dots d_{j_q}) \in D_{j_1} \times \dots \times D_{j_q}$ *the top-k tuples* $(d_{i_1} \dots d_{i_p}) \in D_{i_1} \times \dots \times D_{i_p}$ *with the maximum predicted value of the rating* $g_R(d_1, d_2, \dots, d_w)$.

The Importance of Hierarchies

- Sales at the level of state, region, or the country can be different.
- One can combine the location dimension with the time dimension by aggregating the sales in a particular region over a particular period of time
- Dimensions such as time can be represented in various granular levels of hierarchy, such as hours, days, weeks, months, and so on.
- The location dimension can have a hierarchy corresponding to city, state, region, country, and so on
- The location dimension can have a hierarchy corresponding to city, state, region, country, and so on
- The user needs to make careful choices up front about the hierarchy to use.
- The user needs to make careful choices up front about the hierarchy to use.

The Importance of Hierarchies

Context: Categories of contextual recommendation

- Contextual pre-filtering: In these methods, a segment of the ratings is pre-filtered corresponding to the relevant context.
- Contextual post-filtering: In these methods, the recommendations are first performed on the entire global set of ratings. Subsequently, the ranked recommendation lists are filtered or adjusted as a post-processing step with the use of temporal context.
- Contextual modeling: In this case, the contextual information is incorporated directly into the prediction function, rather than as a pre-filtering or post-filtering step.

Contextual Pre-filtering: A Reduction-Based Approach

Contextual pre-filtering is also referred to as *reduction* [6]. In the reduction-based approach, the idea is to reduce the *w*-dimensional estimation problem to a set of 2-dimensional estimations.

The 2-dimensional estimation problem is equivalent to that in traditional collaborative filtering systems.

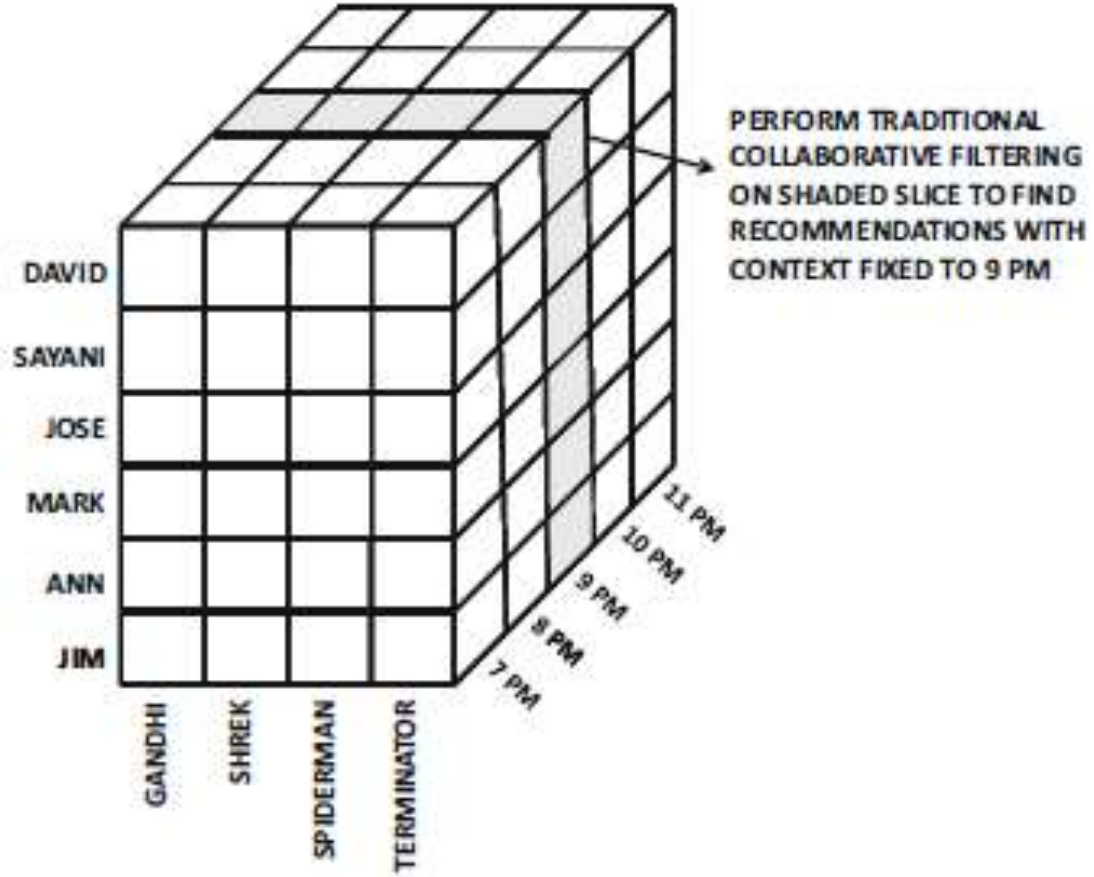
$$gR : U \times I \times T \rightarrow rating$$

$$fR' : U \times I \rightarrow rating$$

At any queried time t, this is achieved by deriving a 2-dimensional ratings matrix $R(t)$ from R with a pair of standard database operations:

$$R'(t) = \text{Project}_{U,I}(\text{Select}_{T=t}(R))$$

$$= \pi_{U,I}\,(\sigma_{T=t}(R))$$



8.3: Extracting a 2-dimensional slice by fixing the context in reduction methods

- The 2-dimensional *slice* of the data cube in which the time is fixed to *t* corresponds to $R(t)$.
- Note that this 2-dimensional slice creates a user-item matrix, which can be used with traditional collaborative filtering algorithms.
- In general, the 3-dimensional ratings estimation can be systematically reduced to 2-dimensional ratings estimation on this slice with the following relationship between the 3-dimensional function $\boldsymbol{g_R}$ and the traditional 2-dimensional collaborative filtering function $\boldsymbol{f_{R'(t)}}$:

$$\forall (u, i, t) \in U \times I \times T,\ \boldsymbol{g_R}(u, i, t) = \boldsymbol{f_{R'(t)}}(u, i)$$

- This approach can easily be generalized to the case where there are $w > 3$ dimensions $D1 \ldots Dw$ by fixing the remaining $w-2$ dimensions. The two dimensions, which are not fixed, are referred to as the *main* dimensions, whereas the other dimensions are the *contextual* dimensions. In typical applications, the users and items are the main dimensions

- A small subset of the ratings are used in a given slice, one may sometimes not have sufficient ratings to perform an accurate recommendation. In such cases, one may aggregate the rating at *t* with other adjacent time slices to create more accurate recommendations.
- Averaging over adjacent slices allows retention of a limited amount of local relevance, while reducing sparsity
- The main advantage of the approach is that it performs the collaborative filtering only over the *relevant* ratings in which the ratings have been selected with the use of the context.
- This can lead to improved accuracy in many cases, although the trade-off is that fewer ratings are now being used for prediction.
- When fewer ratings are available overfitting becomes more likely.

- Global relevance is the extreme generalization of the aggregation process where context is not considered and all the slices are aggregated to reduce sparsity
- The accuracy between the two alternatives depends on the nature of the trade-off between relevance and data sparsity.

Ensemble-Based Improvements

- The goal of the ensemble-based method is to use the best of both worlds in the prediction process. In other words, either the local or the global matrix may be used, depending on the part of the ratings matrix that one is looking at.

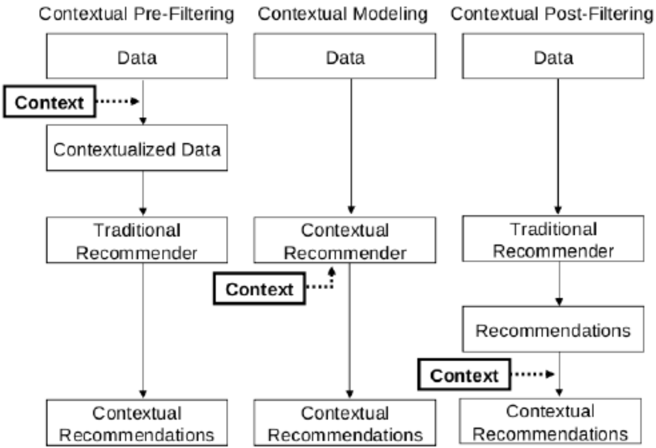
For example, if the contextual variables are location and time, examples of generalizations of (*9 PM, Boston*) might be (*night, Boston*), (*9 PM,Massachusetts*), (*night,Massachusetts*),(*9 PM,* $*$), ($*$*, Boston*), (*night,* $*$), ($*$*,Massachusetts*), and ($*$, $*$)

- One problem with this approach is that it can be very expensive when the number of contextual possibilities is large.

POST-FILTERING METHOD

- Context is ignored in the first step
- Recommendations are generated on all the data by applying a conventional collaborative filtering model on an aggregated user-item matrix.
- Context is then used to adjust or filter the recommended list.

Difference Between Pre- and Post Filtering



Example

- If the context corresponds to summer, winter
- A clothing merchant might want to filter out sweaters and heavy jackets in the summer context.
- Even if they are high in the list of recommended items. Such items can be detected with the use of attribute information.
- Attribute like “wool” for a clothing item may be relevant to the context of the season attribute.

Final Predicted Value

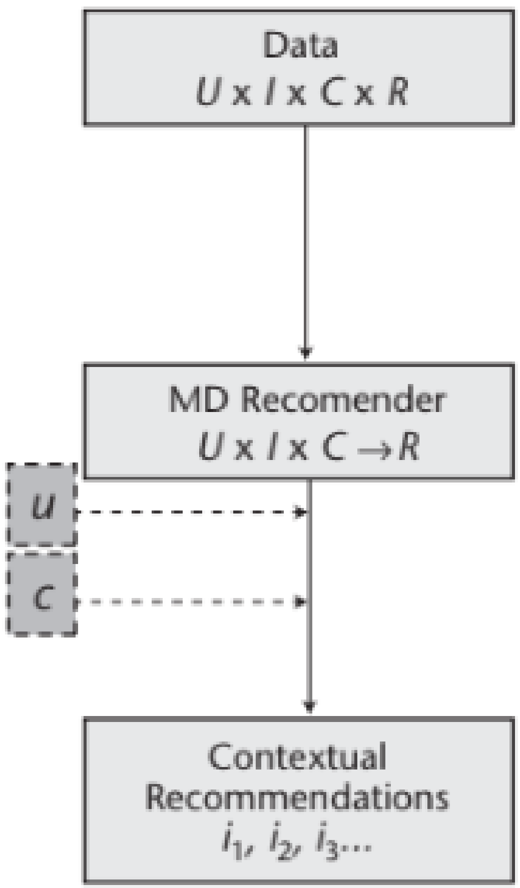
- Predictive model that uses the attributes to estimate the probability of relevance of the item to the context
- context-based probability $P(u, j, C)$
- instead of using the pre-filtered prediction directly as the final result, it is normalized to the range (0, 1)
- adjusted value of the prediction after post-filtering is $P(u, j, C)\hat{r}_{uj}$
- final predicted value of the rating of user i for item j after the post-filtering step is given by $P(\star, j, C)\hat{r}_{uj}$.

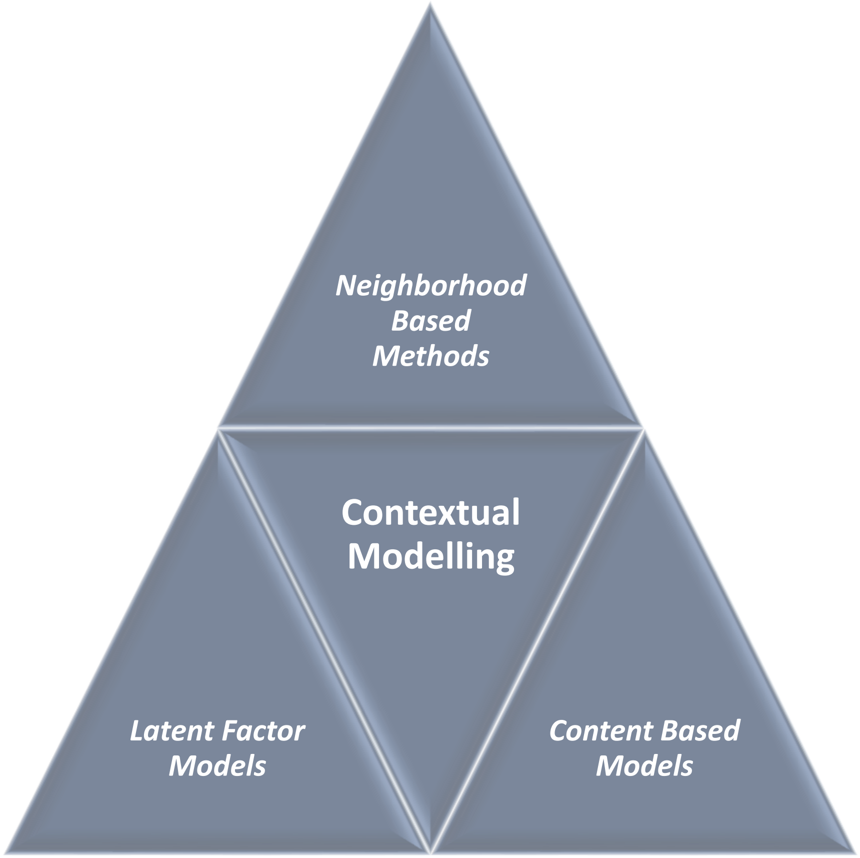
Need of contextual modeling

- Pre-filtering and post-filtering reduces collaborative filtering problem in multidimensional setting to 2-Dimensions, and context is used either in pre-processing or post-processing.
- Thus, pre-filtering and post-filtering have disadvantage that context is loosely integrated into recommender.
- Contextual modeling gives the possibility of **full usage of relationships between user-item combinations and contextual values**.

Contextual Modeling

Incorporate context directly into the recommendation process





Neighborhood Based Methods

- Use contextual dimensions in the similarity computation process.
- **Distance computation**

  Three dimensions corresponding to users, items, and context (e.g. time)

  Let A = (u, i, t) and B = (u’ , I’ , t’) be two data points in 3-dimensional cube.

  - Sum of the weighted distances between the individual dimensions

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

  - Weighted Euclidean metric

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

❖ *How to calculate* $Dist(u,u')$, $Dist(i,i')$, and $Dist(t,t')$’?

Neighborhood Based Methods

- Predicted rating for cell C is weighted average of the closest r (observed) ratings are determined by distance metric.

$$r'_C = \frac{1}{r}\sum_{i=1}^{r} w(C, i) * r_i$$

- Similarity between A and B, is used as weights.

$$w(A, B) = sim(A, B) = \frac{1}{Dist(A,B)}$$

- To give recommendations for user u and context t, rating is predicted for each item.
- Top-K items are recommended.

Content Based Methods

- Generalized content based models as **attributes get associated with any of the three dimensions** (user, item or context)
- Use of **machine learning models**, like SVM and linear regression.
- Example – Consider restaurant recommendation system along with contextual dimensions
  - Liked and disliked items are separated by SVM.
  - Each item-context combination is represented as feature vector.

Content Based Methods

Linear Regression with contextual information

- Estimate regression coefficient vectors $\overline{W_q}$
- Predicted rating is given by

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

where

$\overline{y}_i$ *corresponds to feature variable vector of user i*

$\overline{z}_j$ *corresponds to feature variable vector of item j*

$\overline{v_k}$ *corresponds to feature variable vector of kth dimension of context*

$\overline{z}_j \bigotimes \overline{v}_k$ *represent Kronecker product*

Latent Factor Models

-Ambuj Mishra
(202116003)

Introduction

- The traditional context-sensitive representation is indeed an w-dimensional cube, and therefore it is particularly well suited to tensor factorization.
- Tensor factorization methods can be considered contextual generalizations of conventional matrix factorization methods in recommender systems.
- Tensor Factorization methods are not feasible if the underlying data cube is very large.
- One simplified pairwise interaction approach that can be used is : ***Pairwise Interaction Tensor Factorization (PITF)***

Predicted Ratings and Cost Function

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

➡ $$S = \{(i, j, c) : r_{ijc} \text{ is observed}\}$$

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

Error Optimization using Gradient Descent

$$u_{iq} \Leftarrow u_{iq} - \alpha \frac{\partial J}{\partial u_{iq}} \quad \forall i \quad \forall q \in \{1 \dots k\}$$

$$v_{jq} \Leftarrow v_{jq} - \alpha \frac{\partial J}{\partial v_{jq}} \quad \forall j \quad \forall q \in \{1 \dots k\}$$

$$w_{cq} \Leftarrow w_{cq} - \alpha \frac{\partial J}{\partial w_{cq}} \quad \forall c \quad \forall q \in \{1 \dots k\}$$

$$u_{iq} \Leftarrow u_{iq} + \alpha \left(\sum_{j,c:(i,j,c) \in S} e_{ijc} \cdot (v_{jq} + w_{cq}) - \lambda \cdot u_{iq} \right) \quad \forall i \quad \forall q \in \{1 \dots k\}$$; Where $e_{ijc} = r_{ijc} - \hat{r}_{ijc}$

$$v_{jq} \Leftarrow v_{jq} + \alpha \left(\sum_{i,c:(i,j,c) \in S} e_{ijc} \cdot (u_{iq} + w_{cq}) - \lambda \cdot v_{jq} \right) \quad \forall j \quad \forall q \in \{1 \dots k\}$$

$$w_{cq} \Leftarrow w_{cq} + \alpha \left(\sum_{i,j:(i,j,c) \in S} e_{ijc} \cdot (u_{iq} + v_{jq}) - \lambda \cdot w_{cq} \right) \quad \forall c \quad \forall q \in \{1 \dots k\}$$

Error Optimization using Stochastic Gradient Descent

$$u_{iq} \Leftarrow u_{iq} - \alpha \left[\frac{\partial J}{\partial u_{iq}}\right]_{\text{Contributed by } (i,j,c)} \quad \forall q \in \{1 \dots k\}$$

$$v_{jq} \Leftarrow v_{jq} - \alpha \left[\frac{\partial J}{\partial v_{jq}}\right]_{\text{Contributed by } (i,j,c)} \quad \forall q \in \{1 \dots k\}$$

$$w_{cq} \Leftarrow w_{cq} - \alpha \left[\frac{\partial J}{\partial w_{cq}}\right]_{\text{Contributed by } (i,j,c)} \quad \forall q \in \{1 \dots k\}$$

$$u_{iq} \Leftarrow u_{iq} + \alpha \left(e_{ijc} \cdot (v_{jq} + w_{cq}) - \frac{\lambda \cdot u_{iq}}{n_i^{user}}\right) \quad \forall q \in \{1 \dots k\}$$; Where $e_{ijc} = r_{ijc} - \hat{r}_{ijc}$

$$v_{jq} \Leftarrow v_{jq} + \alpha \left(e_{ijc} \cdot (u_{iq} + w_{cq}) - \frac{\lambda \cdot v_{jq}}{n_j^{item}}\right) \quad \forall q \in \{1 \dots k\}$$

$$w_{cq} \Leftarrow w_{cq} + \alpha \left(e_{ijc} \cdot (u_{iq} + v_{jq}) - \frac{\lambda \cdot w_{cq}}{n_c^{context}}\right) \quad \forall q \in \{1 \dots k\}$$

Factorization Machines

- Originally used in data mining.
- The w-dimensional cube can be flattened to be used as a input in FM model.
- The columns are divided in groups containing users , items and contextual dimensions.
- The rating column is added at the end of the table.
- In the basic scenario , the groups are one hot encoded(every interaction can be defined in form of a unique (user, item, context triplet)).

FM Continued.

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

$$\tilde{r}^{(k)} = \omega_0 + \sum_{i=1}^{n} \omega_i \cdot x_i^{(k)} + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \omega_{i,j} \cdot x_i^{(k)} \cdot x_j^{(k)}$$

$$\omega_{i,j} = \boldsymbol{a}_i \cdot \boldsymbol{b}_j = \sum_{h=0}^{k} a_{i,h} \cdot b_{h,j}$$

$$\tilde{r}^{(k)} = \omega_0 + \sum_{i=1}^{n} \omega_i \cdot x_i^{(k)} + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{h=0}^{k} a_{i,h} \cdot b_{h,j} \cdot x_i^{(k)} \cdot x_j^{(k)}$$

- FM are more powerful than MF as more than one user, item and context can be used per interaction.
- The expressiveness of the model can be modified by changing xi's.
- The main task before apply FM is feature engineering.
- LibFM is a library which provides FM implementations.

THE END