Recommendation Systems - 3

[ML-2]

Revisiting Some Ideas

→ item-item sim

→ user-user sim

→ regression

→ collaborative

Content Based Rec Sys

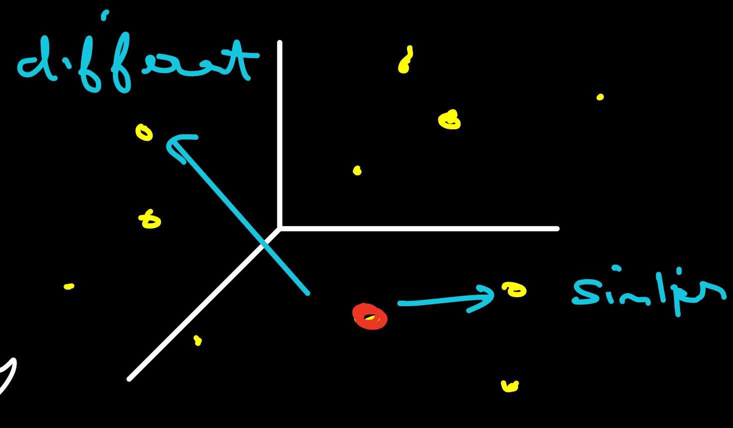
- → item - item sim
- → user - user sim

Represent entity as a vector and run a distance based similarity sort.

$$\text{item } 1 = [0, 1, 0, 0, 1.57, -1.25]$$
$$\text{item } 2 = [1, 0, 1, 0, -1.32, -1.25]$$

scaled

↳ $\text{sim}(1,2) \propto \dfrac{1}{\text{dist}(1,2)}$



→ Rec items similar to ones you liked

→ Rec items liked by users similar to you.

[extended]

Q: How do you accomodate user history into this model?

→ Take avg vector of all historical items,

→ weight of vector $\propto \dfrac{1}{\text{time since user liked that item}}$ [forget slowly]

Eg: days ago	1	15	30
user-1 item_id	A	B	C
rating	5	5	4

↓

vector $\sim \vec{A}(5) \cdot 1 + \vec{B}(5)\left(\frac{1}{15}\right)$

$+ \vec{C}(4)\frac{1}{30}$

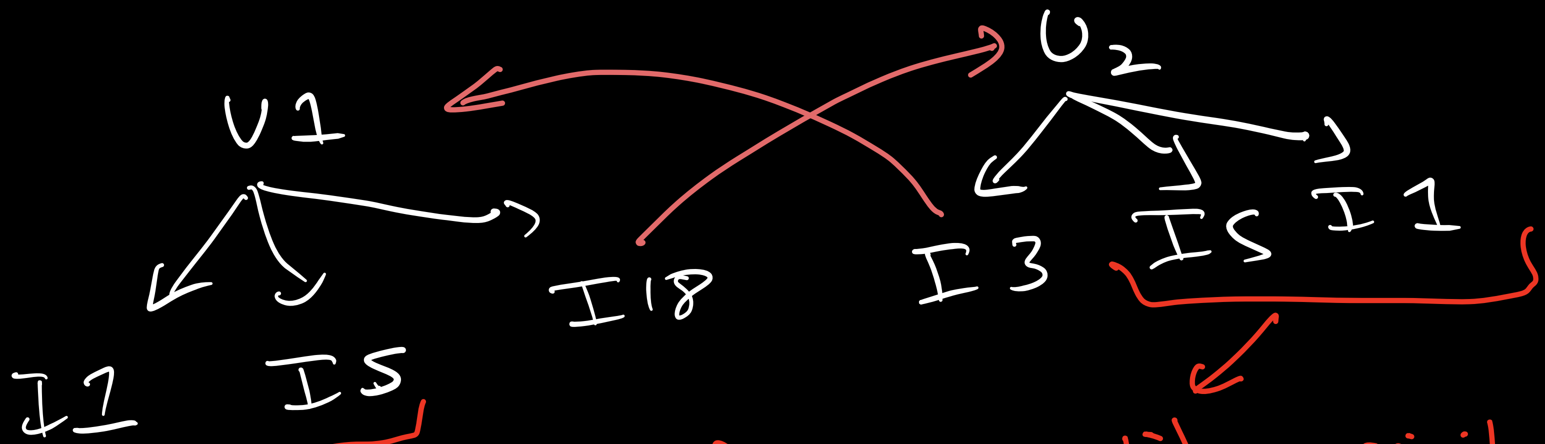
forget -slow → $1/\log(\text{days})$

↳ simple exp smoothing

Q: What if user / item features are not available or useful?

↳ **Collaborative Filtering**

[Cross recommend]



↳ u1 & u2 like similar content,

u1 is similar to u2 !!

Rec I3 → u1

I18 → u2

But, doing this for M of users and thousands of items can be very difficult

↓ Representation !!

Q: How do you represent a user using only their watch history?

$$
\begin{array}{cccccccc}
 & I1 & I2 & I3 & & \ldots & & In \\
\text{user\_1}: & [\,0, & 0, & 4.5, & 0, & 2, & 0, 0 \ldots\,] &
\end{array}
$$

0 ↓ did not watch this movie

4.5 ↓ like this

2 ↓ does not like this

u1 u2 u3 u4 - - - un

item_3 : [4.5, 0, 0, 0, 0, 5, 0..]

↓ U1 has given 4.5 to me

↓ U3 has not watched

. . .

Q: If there are 100 users, and 10 items

→ Length of I1 ? → 100

→ Length of u1 ? → 10

matrix →

$$
\begin{array}{c|cccc|}
 & I_1 & I_2 & \cdots & I_m \\
\hline
u_1 & & & & \\
u_2 & \cdots & 4.5 & & \cdots \\
\vdots & & & & 1 \\
 & 5 & & \cdots & \\
u_n & & & & \cdots \\
\hline
\end{array}
\quad n \times m
$$

|

Can you guess the sparsity for Netflix?

→ 5M users

→ 100K items

$5 \times 10^6 \times 100 \times 10^3$

$= 5 \times 10^{11}$ cells !!

on avg each user has seen ___ items? → 1000?

$$\frac{5 \times 10^6 \times 1000}{5 \times 10^{11}} = \underline{0.01}$$

↓

99% empty!!

To make recommendatn I need to estimate ratings for the 99% empty space!

Matrix Factorisatn

factor of a number:

$$12 = 3 \times 4$$

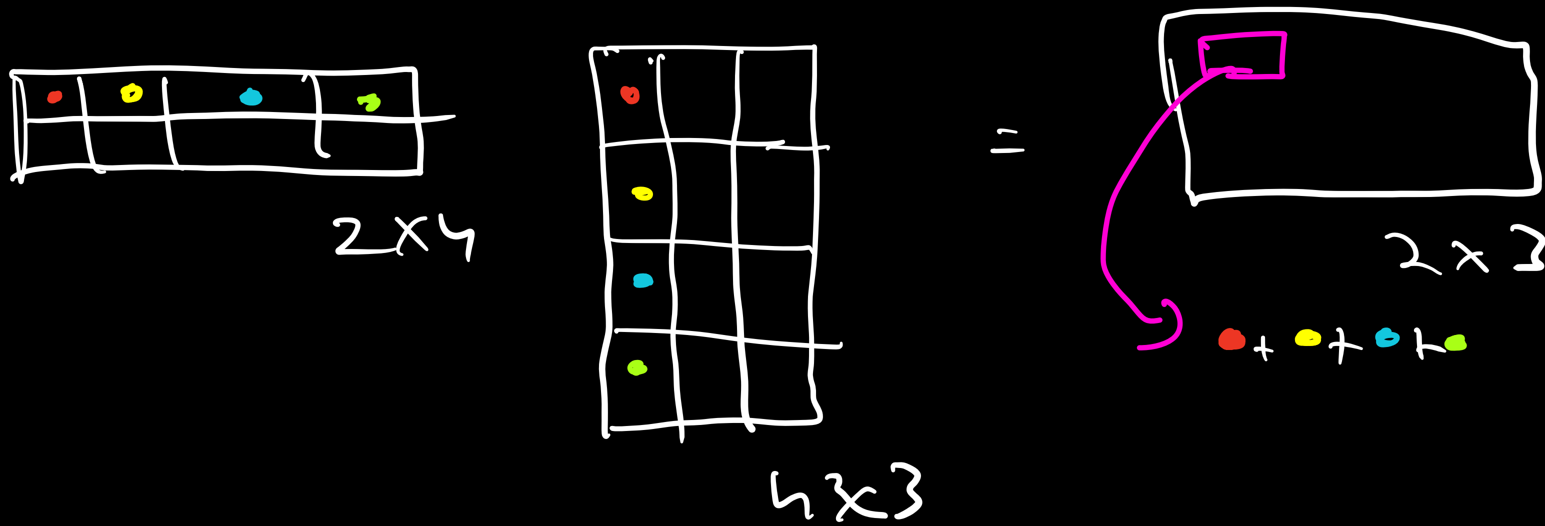
$$36 = 6 \times 6, \quad 9 \times 4, \quad 12 \times 3$$

↓

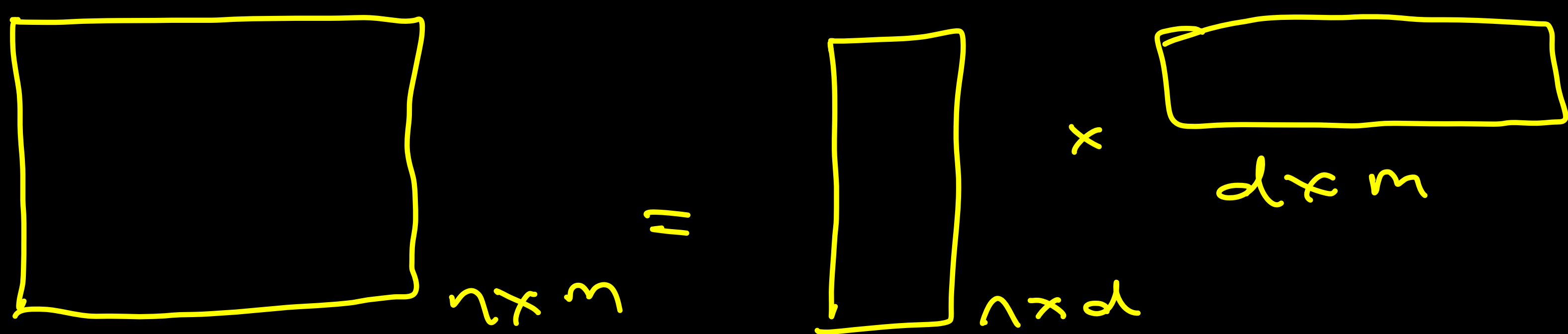
break down large num into a product of 2 smalle numbers.

Basics of matrix mult.

$$A_{2\times 4} \times B_{4\times 3} = C_{2\times 3}$$



$\therefore$



$$10^6 \times 10^5 = 10^6 \times 3 \,,\; 3 \times 10^5$$

Illustration of Matrix Factorization

R

Rating Matrix

User \ Item	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

=

P

User Matrix

A	1.2	0.8
B	1.4	0.9
C	1.5	1.0
D	1.2	0.8

X

Q

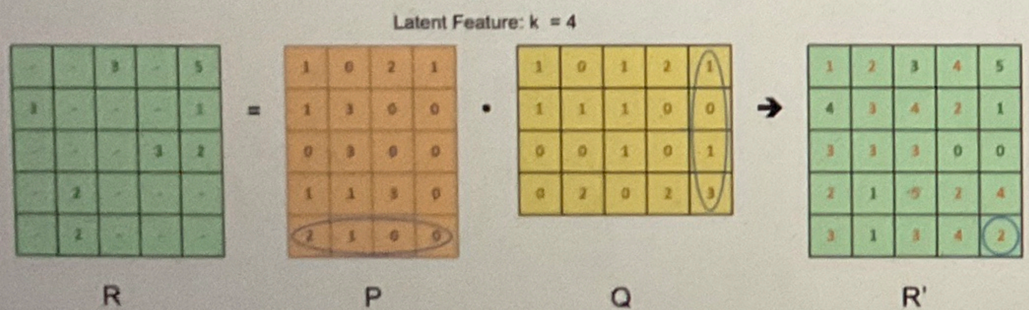
Item Matrix

W	X	Y	Z
1.5	1.2	1.0	0.8
1.7	0.6	1.1	0.4

Recommendations: Dot product of embeddings of user & item to be recommended

$$\textit{Predicted Ratings}: \; r'_{ui} = p_u^T q_i$$

Illustration showing sample dot product for a predicted rating



$$\hat{r}_{ui} = p_u^T q_i$$

Q: How can I learn user and Item condensed embeddings?

$$\min_{U, I} \sum_{i} \sum_{j} \left(R_{ij} - p_u^T q_i \right)^2$$

$R_{ij} \neq 0$

Gradient Descent

sum of squared error !!

However, some movies will always get good ratings and some users will always give good ratings, etc. These biases can affect our model.

it is best to model them explicitly.

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

Q: What value of 'd'?

$$n \times m = \underbrace{n \times d \quad \times \quad d \times m}_{\text{Hyperparam}}$$

High d → less compression

Low d → more compression

↳ loss of informatⁿ

the different rating values. However, much of the observed variation in rating values is due to effects associated with either users or items, known as *biases* or *intercepts*, independent of any interactions. For example, typical collaborative filtering data exhibits large systematic tendencies for some users to give higher ratings than others, and for some items to receive higher ratings than others. After all, some products are widely perceived as better (or worse) than others.

Thus, it would be unwise to explain the full rating value by an interaction of the form $q_i^T p_u$. Instead, the system tries to identify the portion of these values that individual user or item biases can explain, subjecting only the true interaction portion of the data to factor modeling. A first-order approximation of the bias involved in rating r_{ui} is as follows:

$$b_{ui} = \mu + b_i + b_u \tag{3}$$

The bias involved in rating r_{ui} is denoted by b_{ui} and accounts for the user and item effects. The overall average rating is denoted by μ; the parameters b_u and b_i indicate the observed deviations of user u and item i, respectively, from the average. For example, suppose that you want a first-order estimate for user Joe's rating of the movie *Titanic*. Now, say that the average rating over all movies, μ, is 3.7 stars. Furthermore, *Titanic* is better than an average movie, so it tends to be rated 0.5 stars above the average. On the other hand, Joe is a critical user, who tends to rate 0.3 stars lower than the average. Thus, the estimate for *Titanic*'s rating by Joe would be 3.9 stars (3.7 + 0.5 – 0.3). Biases extend Equation 1 as follows:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \tag{4}$$

Here, the observed rating is broken down into its four components: global average, item bias, user bias, and user-item interaction. This allows each component to explain only the part of a signal relevant to it. The system learns by minimizing the squared error function:[4,5]

MSE

$$\min_{p*,q*,b*} \sum_{(u,i)\in K} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda$$

regularisation

$$(\| p_u \|^2 + \| q_i \|^2 + b_u^2 + b_i^2) \tag{5}$$

Cold Start

As you can see, new movies or users cannot recieve rec based on collaborative fitering. This issue is called cold start!!

Some remedies are to use a diff algo till the user/item is "new".

→ Popolarity / reccency based

→ Content based

→ Random, etc.