Recommendation Systems - 2

[ML-2]

→ Lift gives bi-dir recom..

Revisiting Some Ideas

- → item - item sim
- → user - user sim
- → regression
- → collaborative

} - history features?

-> extended discussion
  ↳ likes matrix, shared matrix, etc..

Content Based Rec Sys

→ item - item sim

→ user - user sim

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

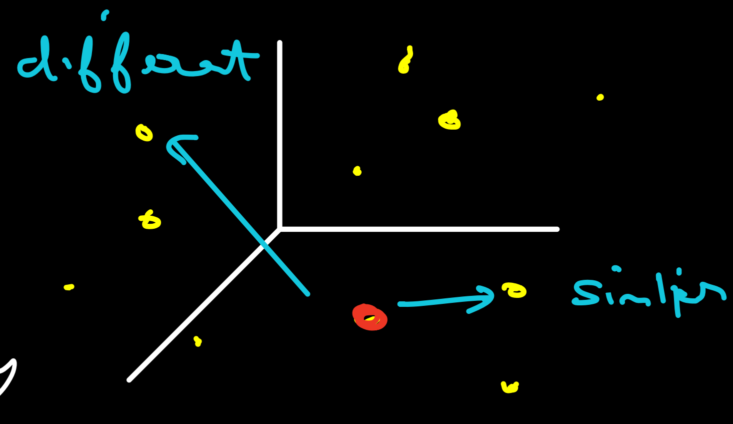
item 1 = [0, 1 0, 0, 1.57, -1.25]

item - 2 = [1, 0, 1, 0, -1.32, -1.25]

scaled

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

→ 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 historic 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

$$\text{vector} \sim \vec{A}(5) \cdot 1 + \vec{B}(5)\left(\frac{1}{15}\right) + \vec{C}(4)\frac{1}{30}$$

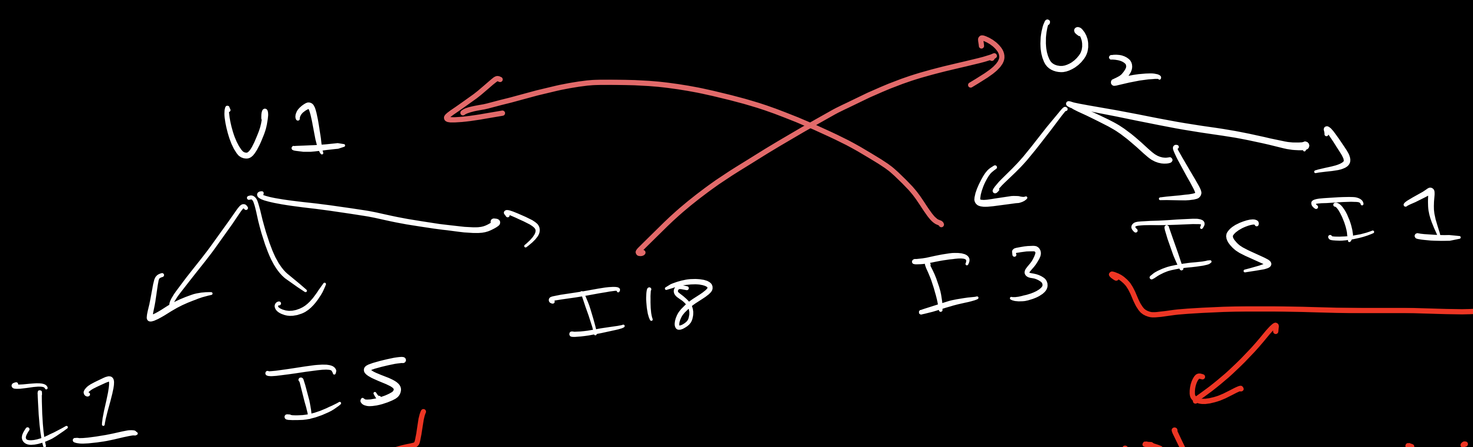
forget -slow → $1/\log(\text{days})$

↳ 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?

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

0 → did not watch this movie

4.5 → likes this

2 → does not like this

$$\begin{array}{cccccccc} U1 & U2 & U3 & U4 & - & - & - & U_n \end{array}$$

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 ->

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

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?

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

↓

99% empty !!

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

Matrix Factorisatn

factor of a number:

$$12 = 3 \times 4$$

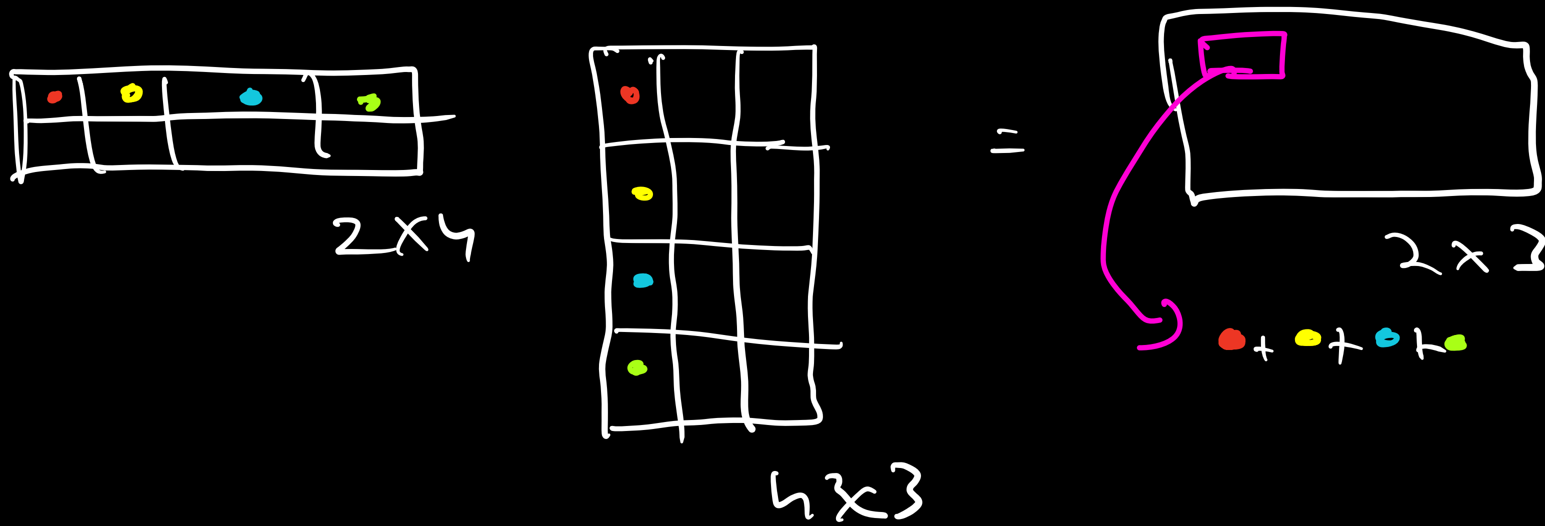
$$36 = 6 \times 6,\ 9 \times 4,\ 12 \times 3$$

↓

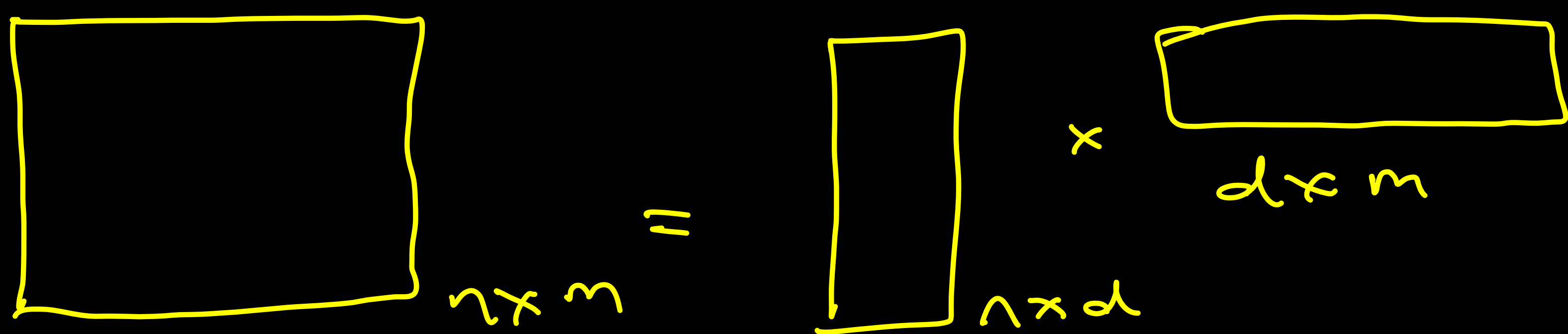
break down large num into a product of 2 smalle numbers.

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

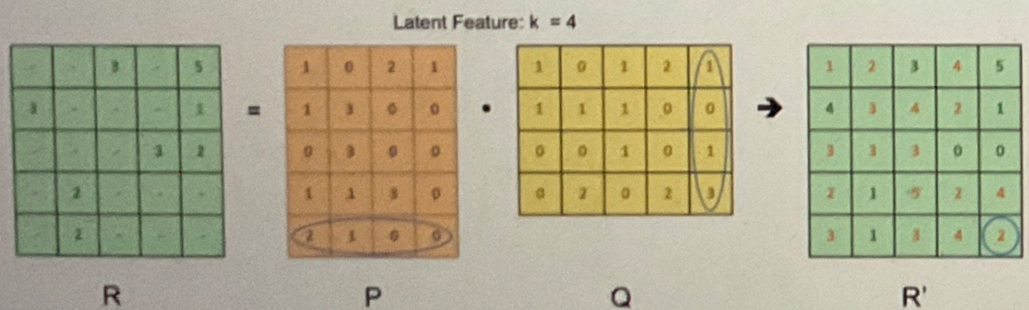
Q (Item Matrix)

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



$$R_{ij} = \vec{U}_i \cdot \vec{I}_j$$

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

$$\min_{U, I} \sum_{i} \sum_{j} (R_{ij} - U_i \cdot I_j)^2$$

$$R_{ij} \neq 0$$

Gradient Descent

sum of squared error!!

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 <u>information</u>