Introduction to Recommendation Engine

Ching Lee

2018/09/07

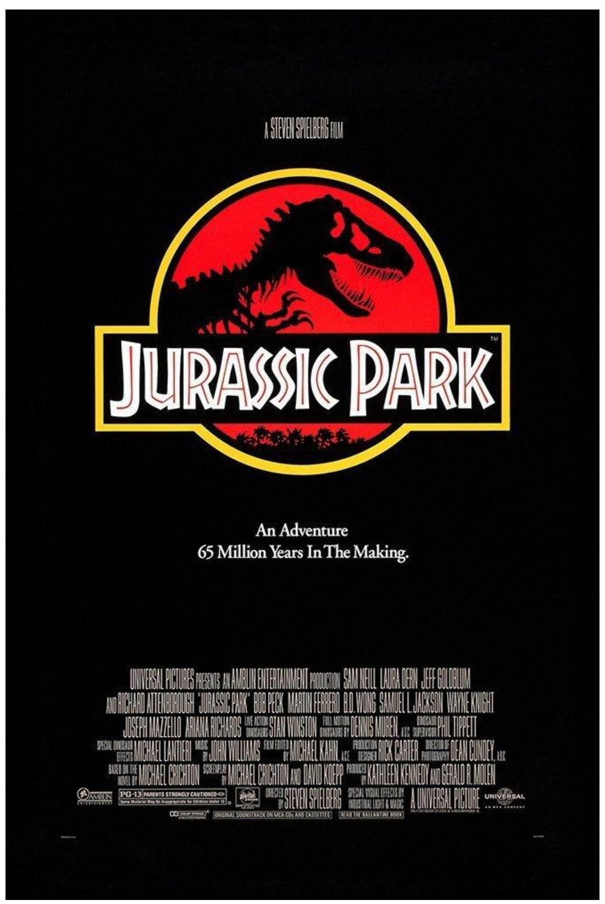
Prologue

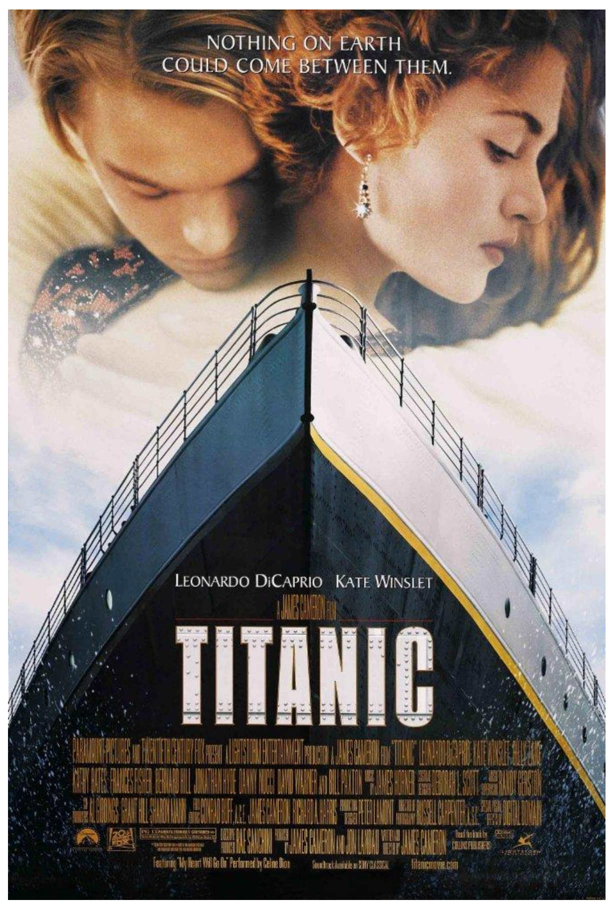
- People are increasing the reliance on conveniences such as e-commerce store or streaming entertainment.

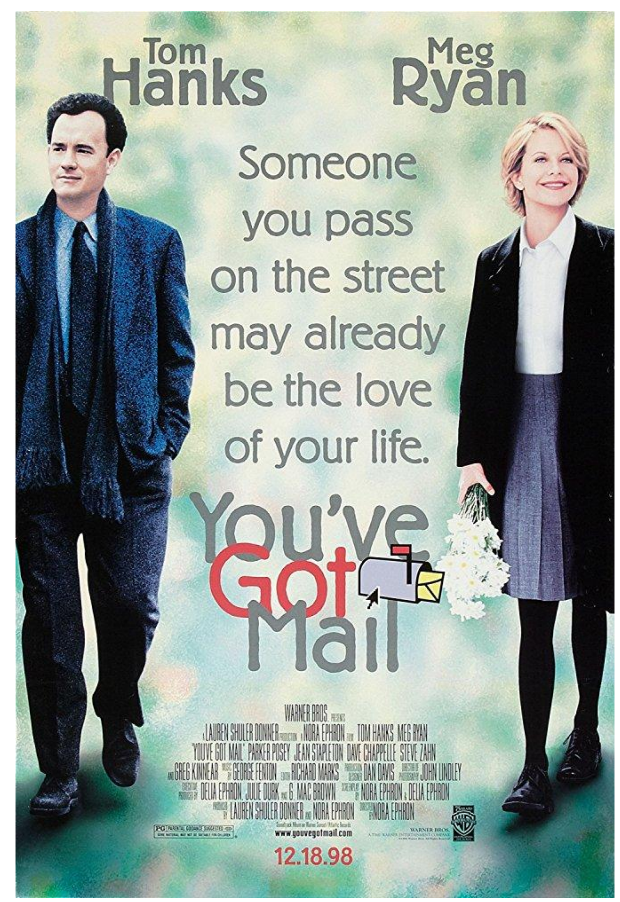
- "Guess" what the customers may like in advance, so to promote more things to sell and in turn generate more revenue.

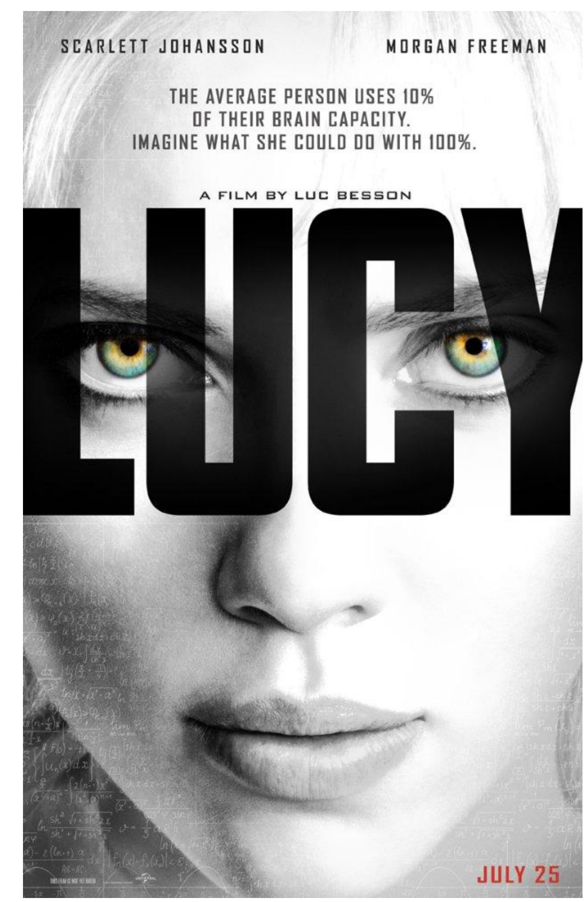
- A *recommendation engine (RE)* is any kind of model that can infer the relationship between users/items and make proper prediction for the users.

Gender	Male
Age	30
Prefer genre	Sci-fi, comedy, action
Prefer director	Steven Spielberg, Christopher Nolan, Michael Bay, James Cameron
Prefer actor/actress	Tom Hanks, Leonardo DiCaprio, Anne Hathaway, Scarlet Johansson

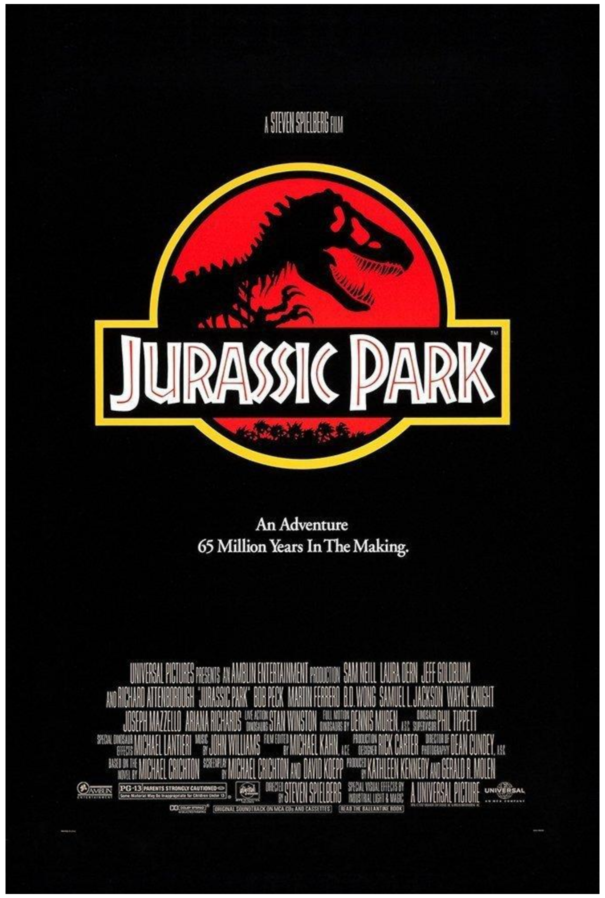


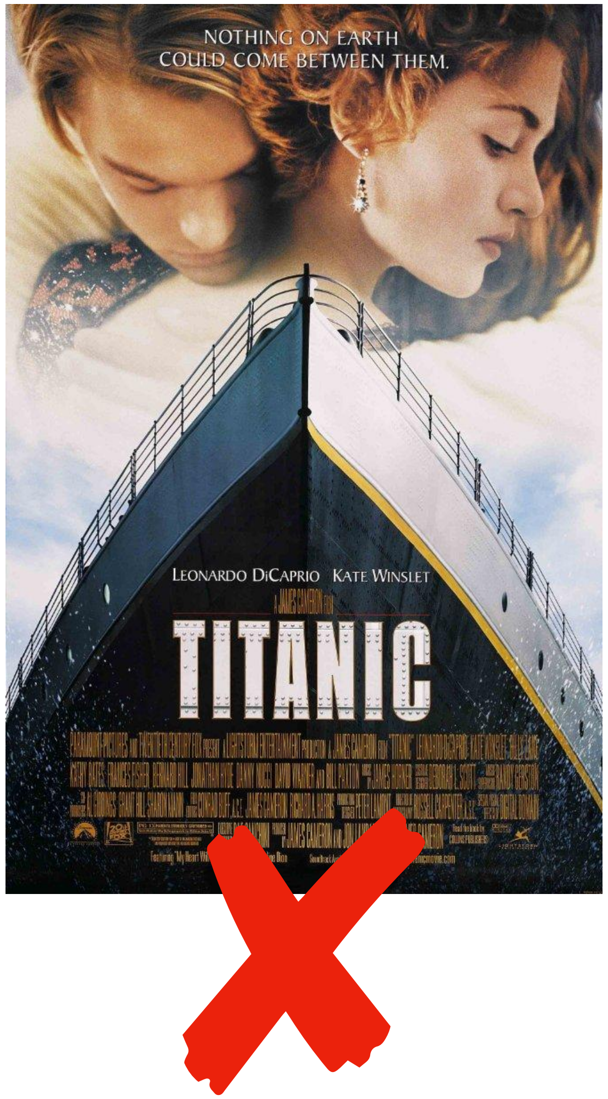


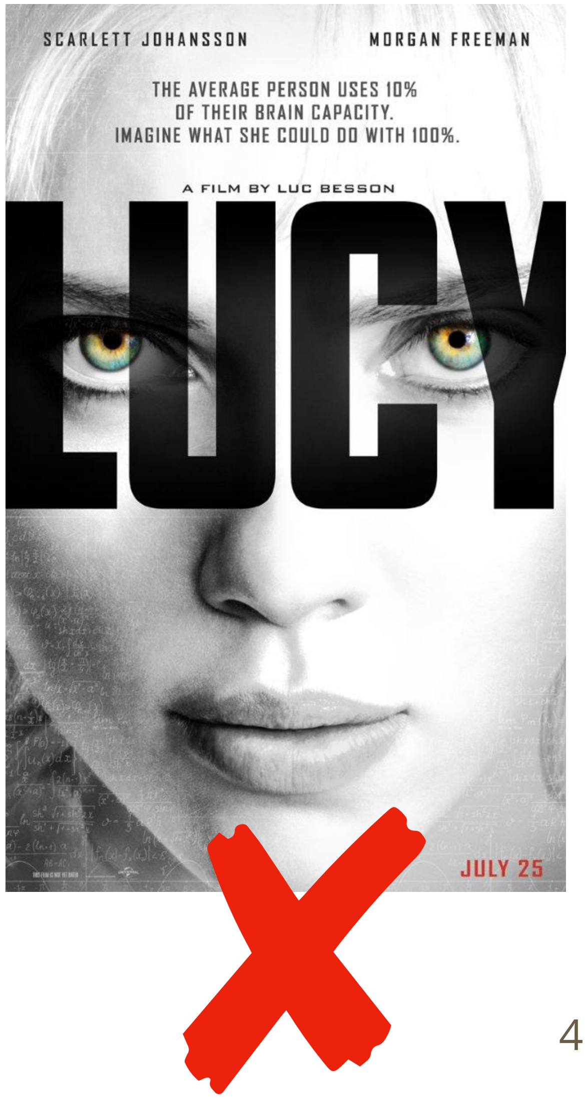




Gender	Male
Age	30
Prefer genre	Sci-fi, comedy, action
Prefer director	Steven Spielberg, Christopher Nolan, Michael Bay, James Cameron
Prefer actor/actress	Tom Hanks, Leonardo DiCaprio, Anne Hathaway, Scarlet Johansson







Content Filtering

- What we have just conducted is essentially one way of doing recommendation: *content filtering*.

- By **building profiles** for both the users and movies, we can provide recommendation by matching the **content** between the two groups, hence the name.

- Take a lot of efforts to build such profiles and many conditional settings to fit a person’s taste.

Strategies for Recommendation

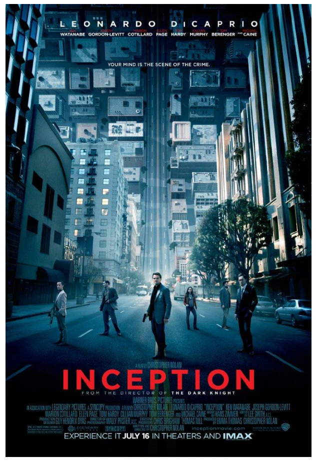
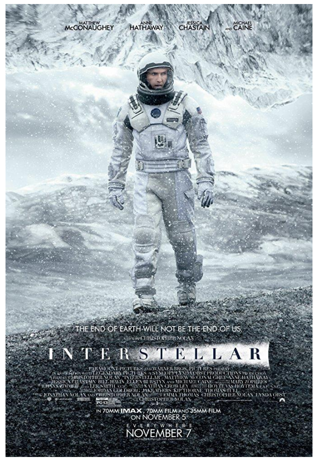
- Beside *content filtering* we had just mentioned, there is another method called *collaborative filtering (CF)*.

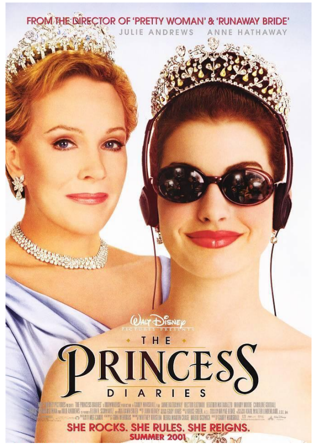
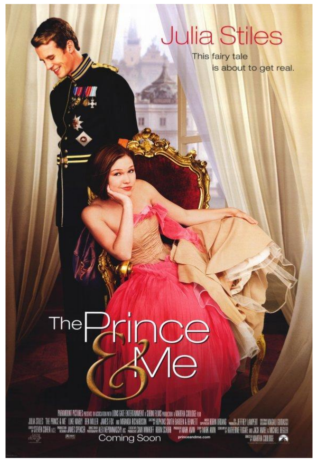
- *CF* relies on past user behavior among a group of users (hence *collaborative*), so we aren't required to create profiles explicitly.

- The two primary areas of *CF* are the *neighborhood methods* (*memory-based*) and *latent factor models*.



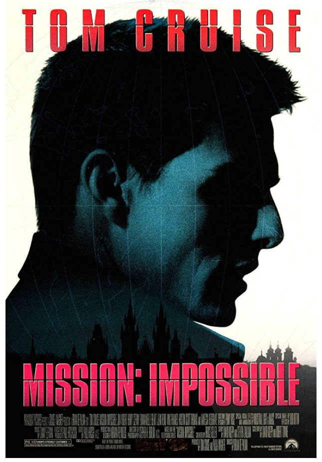
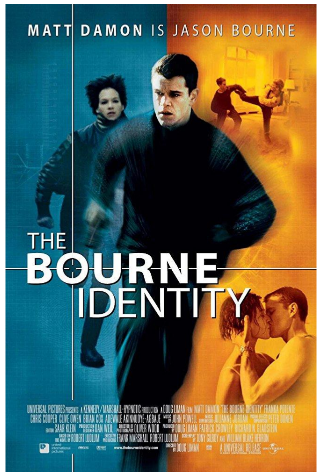
THE PRESTIGE
OCTOBER 20

THE DARK KNIGHT
JULY 18

LEONARDO DICAPRIO
INCEPTION

INTERSTELLAR
NOVEMBER 7


FROM THE DIRECTOR OF 'PRETTY WOMAN' & 'RUNAWAY BRIDE'
THE PRINCESS DIARIES
SHE ROCKS. SHE RULES. SHE REIGNS.

Julia Stiles
The Prince & Me

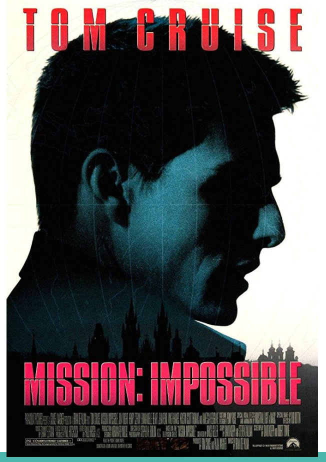
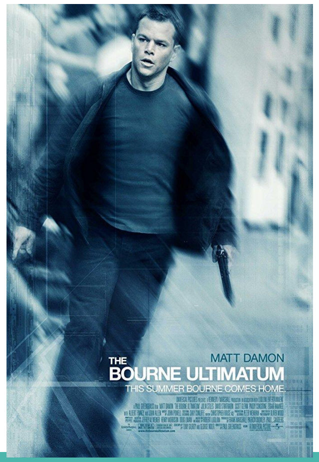
RICHARD GERE
JULIA ROBERTS
PRETTY WOMAN

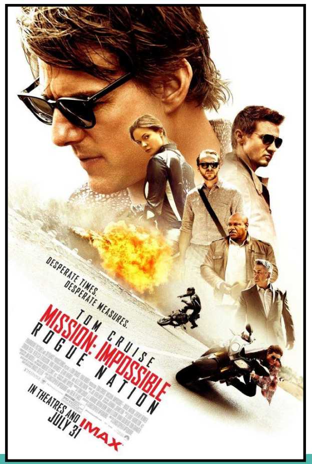
jennifer lopez
ralph fiennes
maid in manhattan
December 13...love checks in

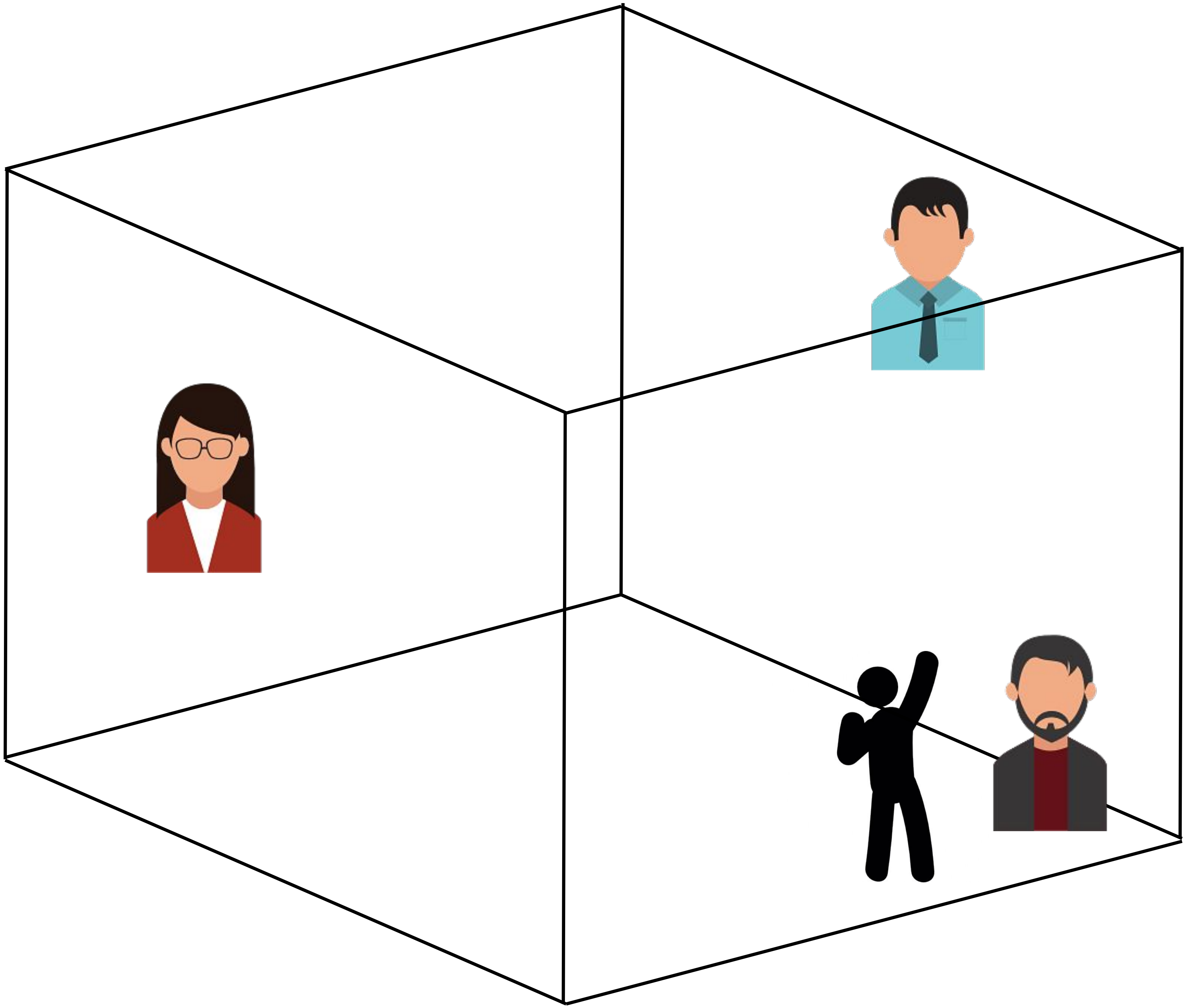

TOM CRUISE
MISSION: IMPOSSIBLE

MATT DAMON IS JASON BOURNE
THE BOURNE IDENTITY

CASINO ROYALE
NOVEMBER 17

Kingsman
THE SECRET SERVICE


TOM CRUISE
MISSION: IMPOSSIBLE

MATT DAMON
THE BOURNE ULTIMATUM

CASINO ROYALE
NOVEMBER 17

MISSION: IMPOSSIBLE
ROGUE NATION

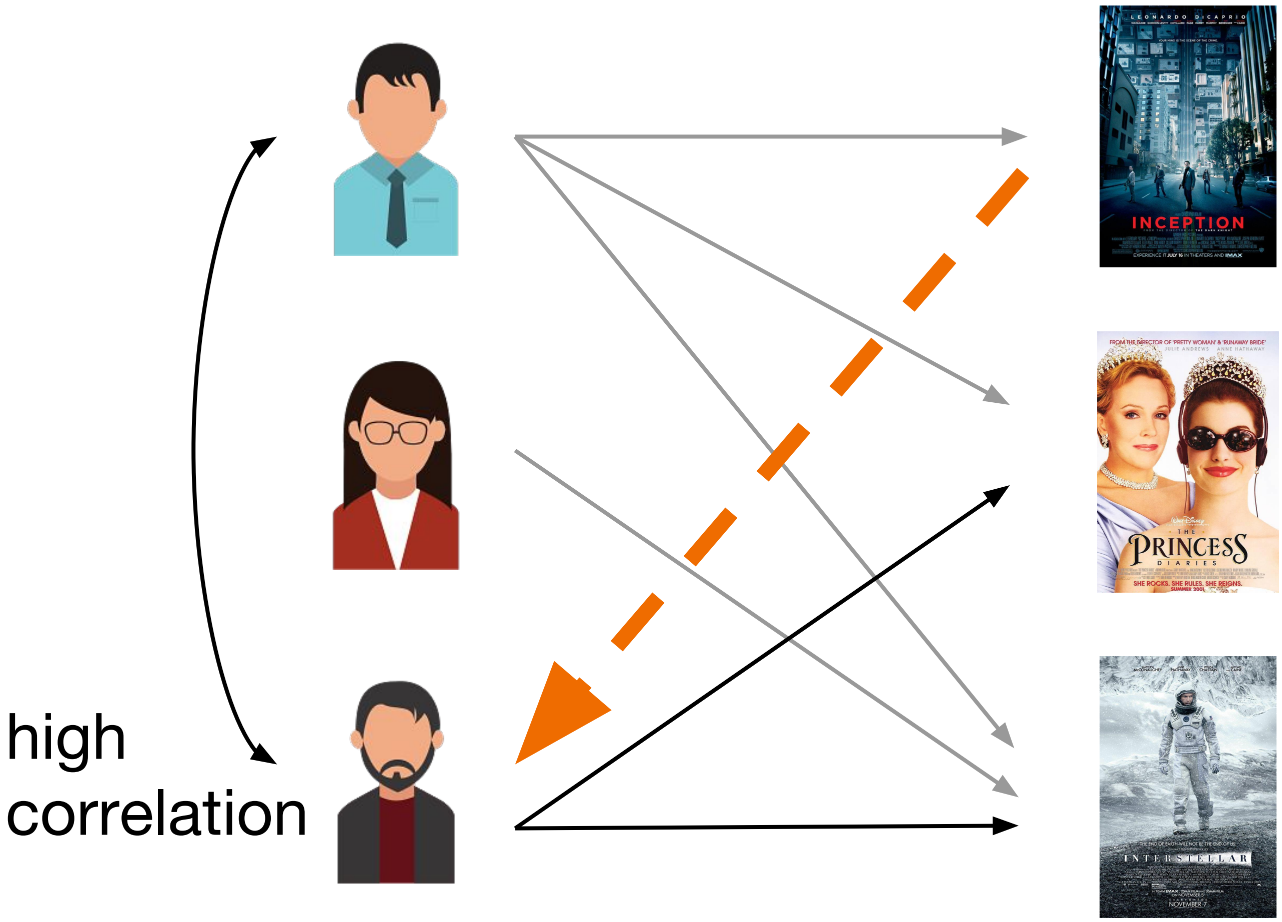


User-based and Item-based CF

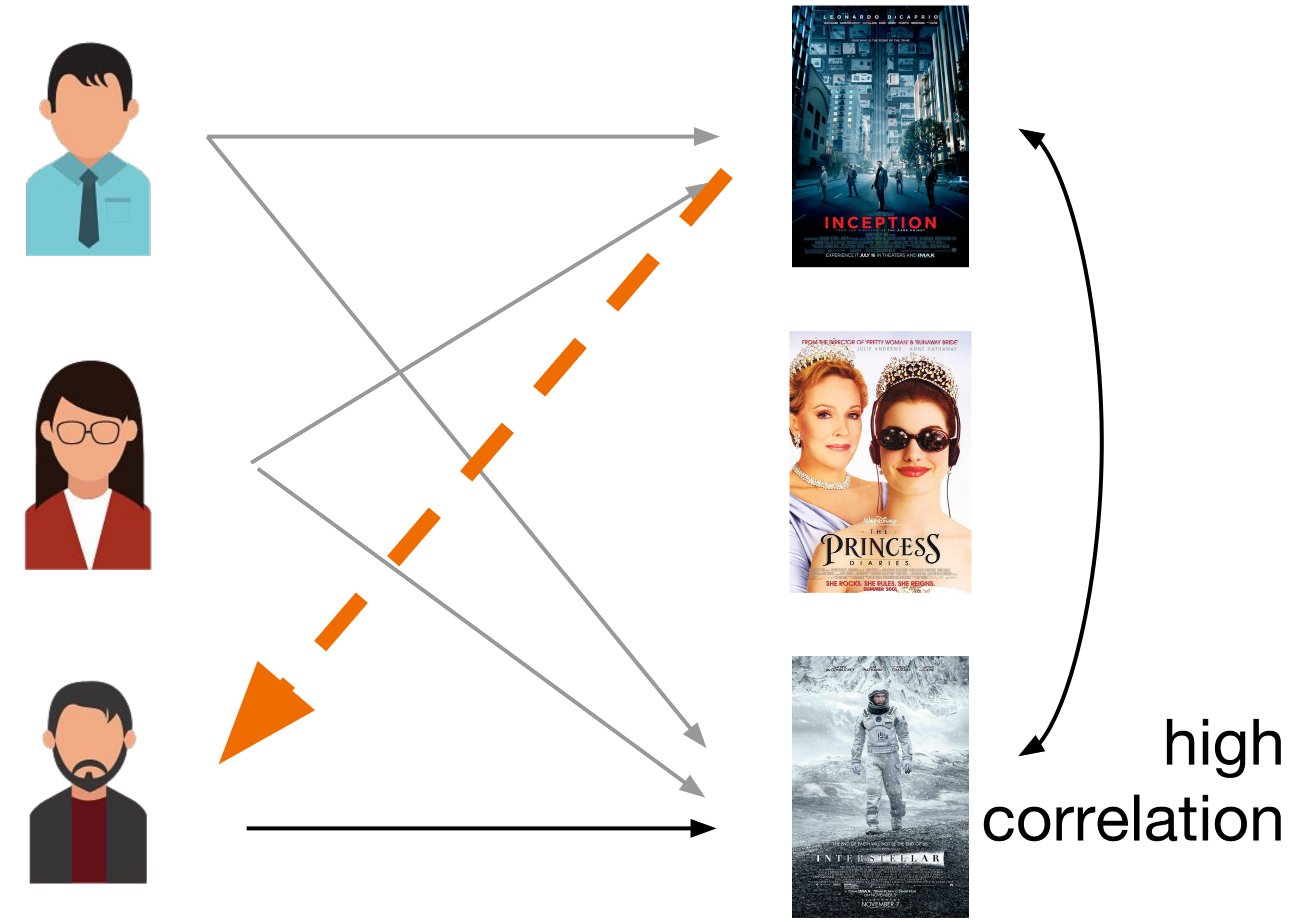
- For neighborhood methods, we have two primary types of algorithms: user-based and item-based.
- In *user-based CF*, we group together users who gave similar ratings to the same set of items, whereby we could later use the ratings of a specific users to predict those of his/her peers.
- For *item-based CF*, we group the items instead.

8.7	10	10	9	?	?	?	9	8	8.5
?	?	?	1	10	10	10	?	?	2
8	9	9	9	?	?	5	10	10	9.5

- - - User-based CF   - - - Item-based CF



User-based filtering



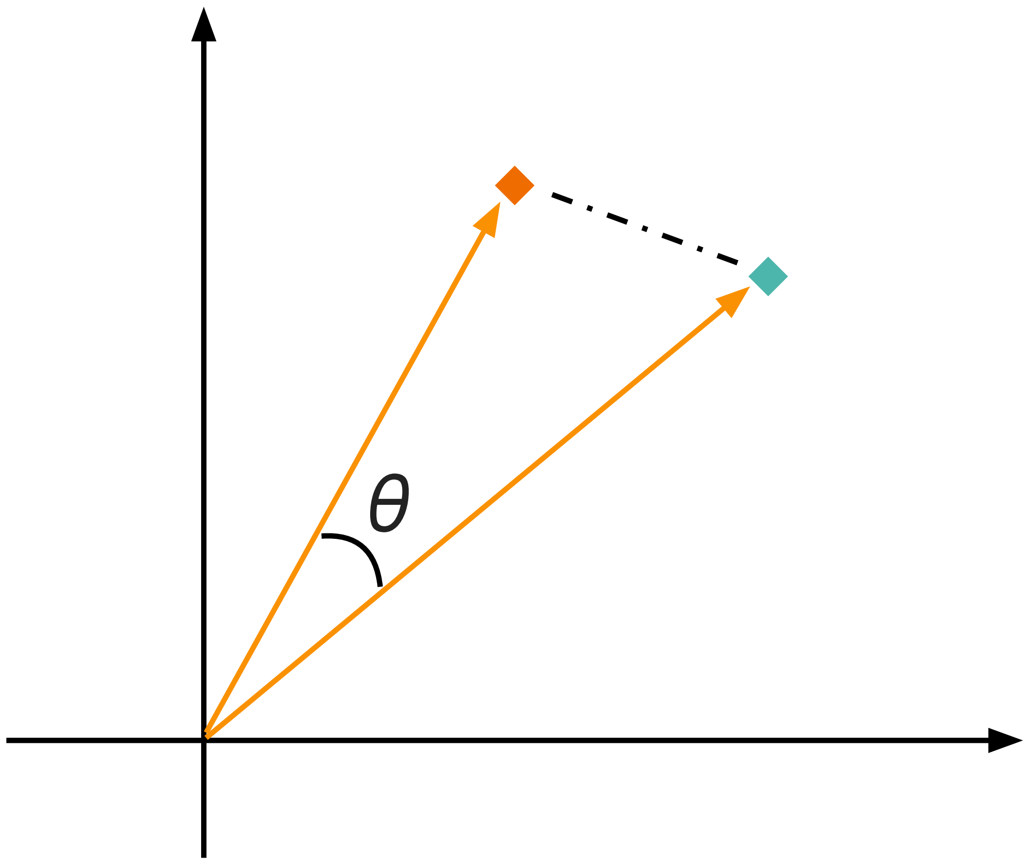
Item-based filtering

You may like it because your "friends" liked it.

You tend to like that item since you have liked those items.

Measuring Similarity

- To measure the similarity between users or items, we can use metrics like Euclidean distance or cosine similarity.



Euclidean distance: distance between points

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

Cosine similarity: angle between vectors

$$\cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|_2\|\mathbf{B}\|_2} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Example: User-Based CF

8.7	10	10	9	?	?	?	9	8	8.5	**23.96**
?	?	?	1	10	10	10	?	?	2	**17.46**
8	9	9	9	?	?	5	10	10	9.5	**24.94**

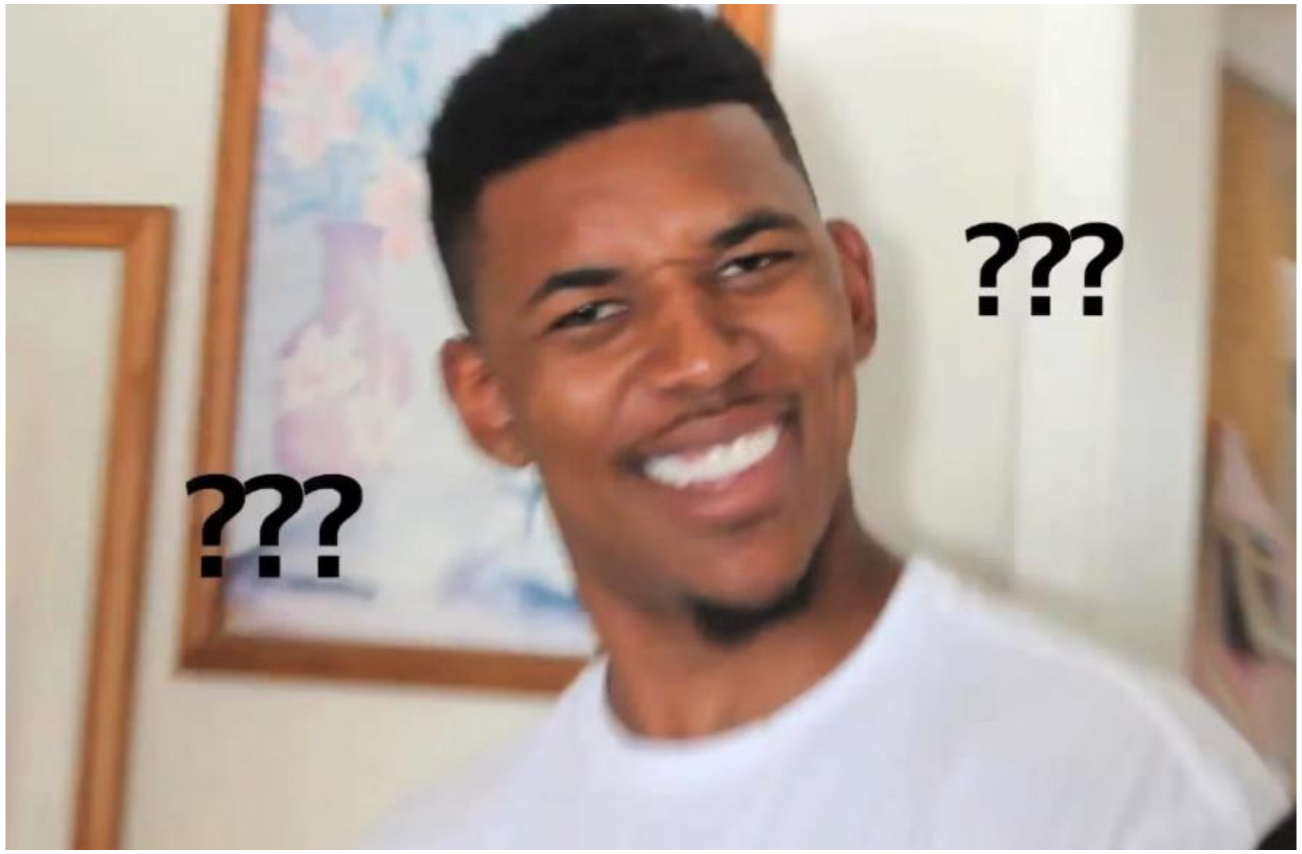
→ cosine similarity:

$$: \frac{9 * 1 + 8.5 * 2}{23.96 * 17.46} = 0.062$$

$$: \frac{8.7 * 8 + 10 * 9 + 10 * 9 + 9 * 9 + 9 * 10 + 8 * 10 + 8.5 * 9.5}{23.96 * 24.94} = 0.973$$

Latent Factor Models

- Explain the rating by characterizing both users and items on a number of *latent factors* inferred from rating patterns.



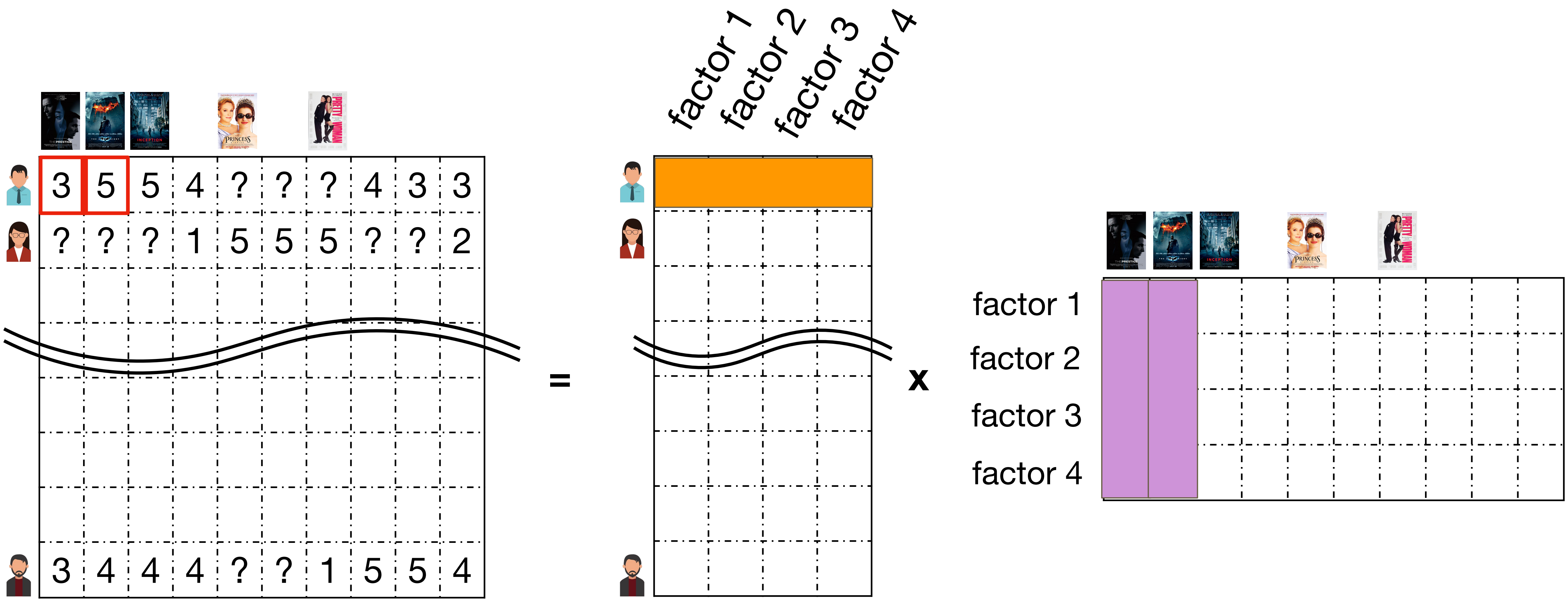
The factors are latent.

No one cares ...

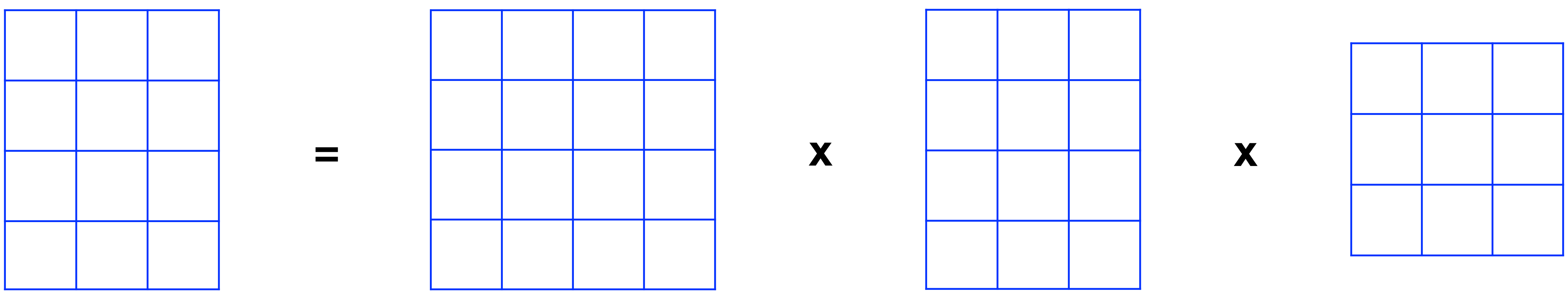
Not directly observable ...

	3	5	5	4	?	?	?	4	3	3
	?	?	?	1	5	5	5	?	?	2
	3	4	4	4	?	?	1	5	5	4

What is latent factor anyway?

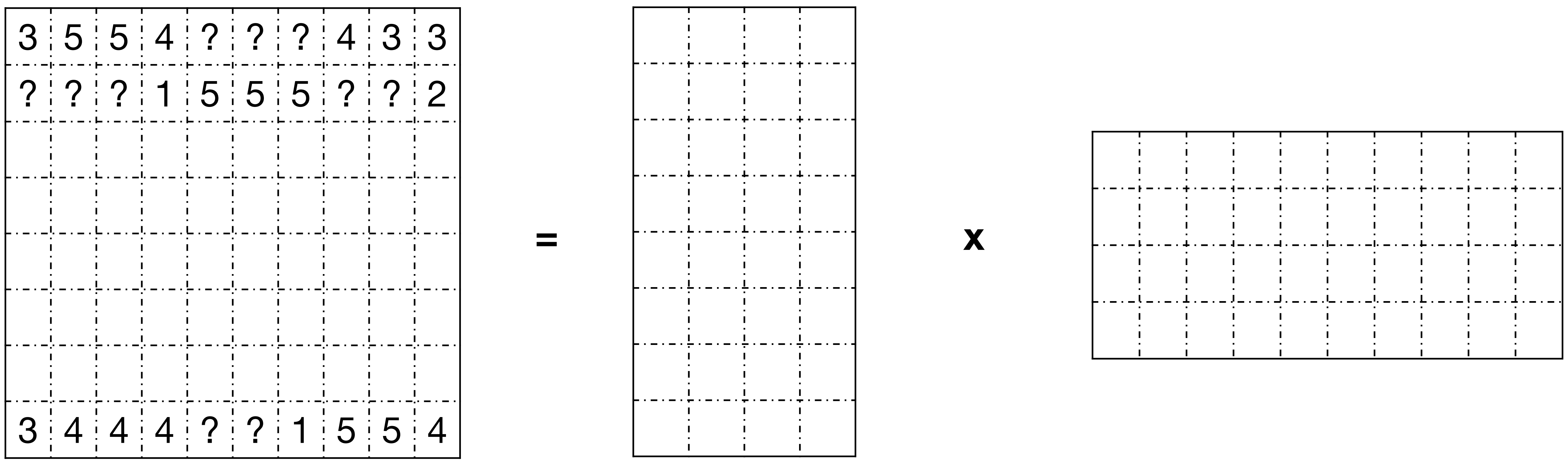


Singular Value Decomposition (SVD)

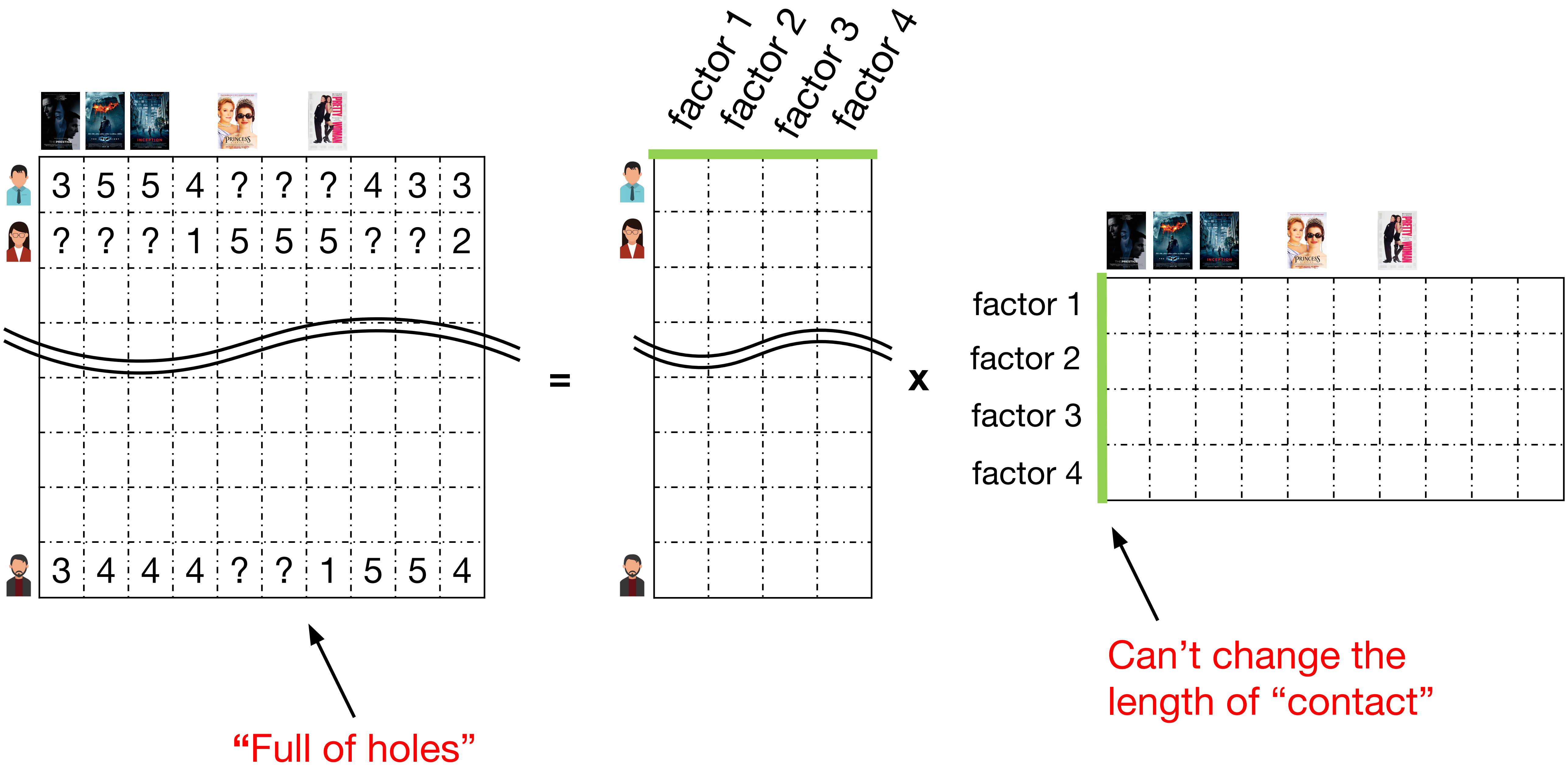


$A \qquad U \qquad \Sigma \qquad V^T$

That is (kind of) what SVD is



But in practice...



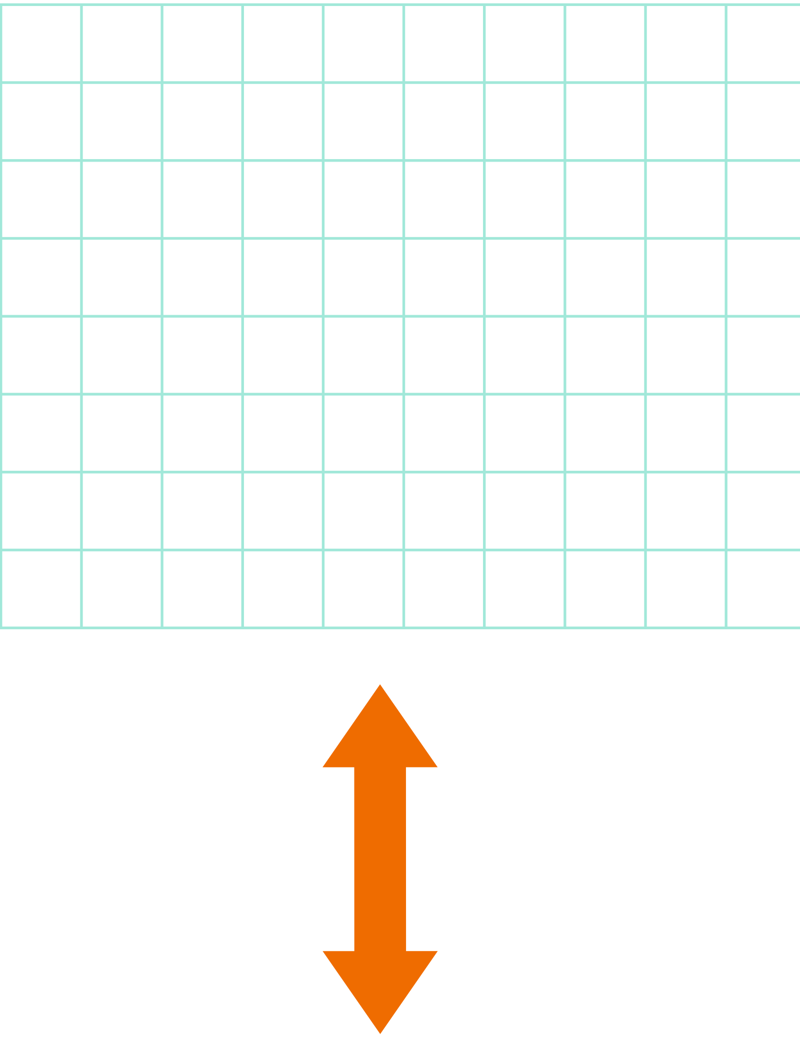
How to factor the matrix

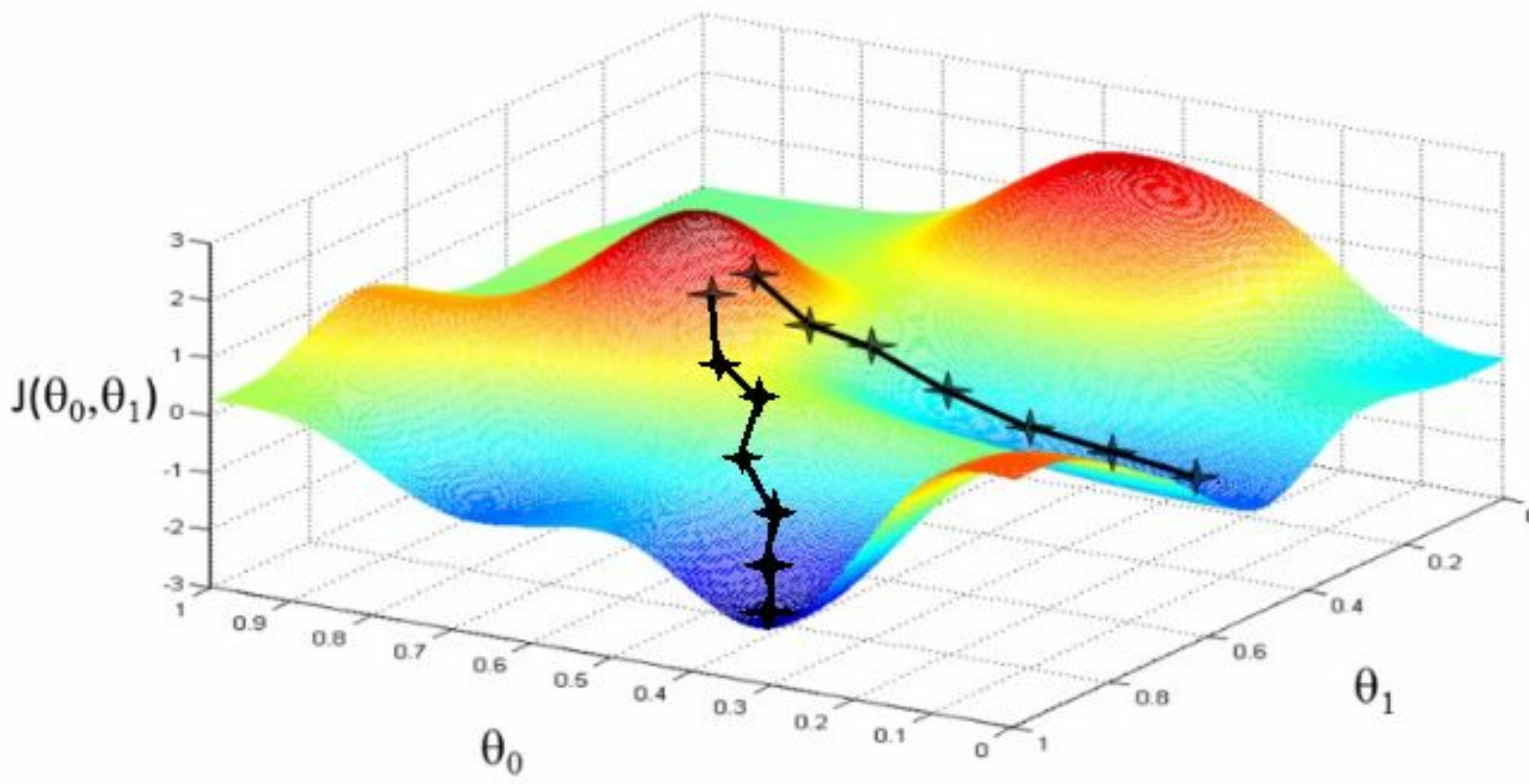
- To perform matrix factorization for large matrices, we learn the entries through optimization methods such as *stochastic gradient descent* (SGD).

- Methods like *alternating least square* (ALS) are also used when computation can be parallelized.

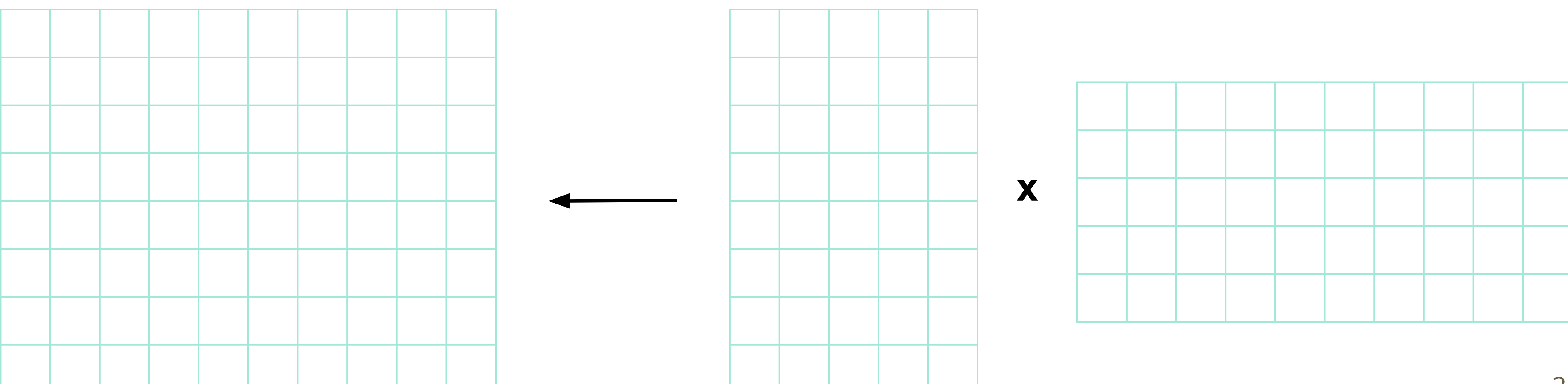
- We are going to briefly introduce SGD for its popularity.

Goal:

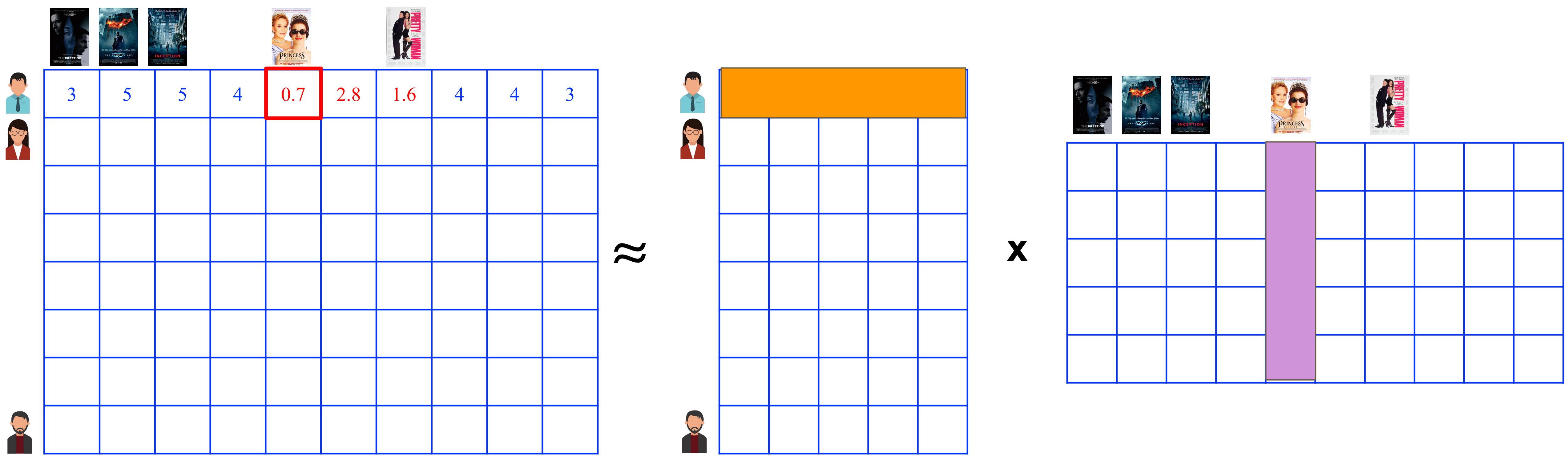




Initialize these and tune the numbers!



Ok, I’ve got the matrices. Then what?



→ Key points:

1. The approximate matrix will not be identical to the original.
2. The factor matrices will keep changing as long as there are users changing the rating (even if *you* stay inactive for a while).

Evaluating a model

1. Root Mean Square Error / Mean Absolute Error

$$\sqrt{\sum_i (Pred_i - True_i)^2} \qquad \sum_i |Pred_i - True_i|$$

2. Confusion Matrix (Precision and Recall)

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

3. Discounted Cumulative Gain (DCG)

Evaluating a model

- To evaluate how a machine learning model did, we use metrics such as *precision* and *recall*.

		Actual value	
		True	**False**
Predicted value	**True**	True positive (TP)	False positive (FP)
	False	False negative (FN)	True negative (TN)

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

Scenario 1:

Predicted value \ Actual value	Have cancer	Safe
Have cancer	45	5
Safe	190	1000

$$\text{Precision} = \frac{45}{45 + 5} = 90\%$$

$$\text{Recall} = \frac{45}{45 + 190} = 19.15\%$$

Scenario 2:

Predicted value \ Actual value	Have cancer	Safe
Have cancer	200	800
Safe	35	205

$$\text{Precision} = \frac{200}{200 + 800} = 20\%$$

$$\text{Recall} = \frac{200}{200 + 35} = 85.11\%$$

That's why we have F1-score

- F1-score (or F-score) is the harmonic mean between precision and recall:

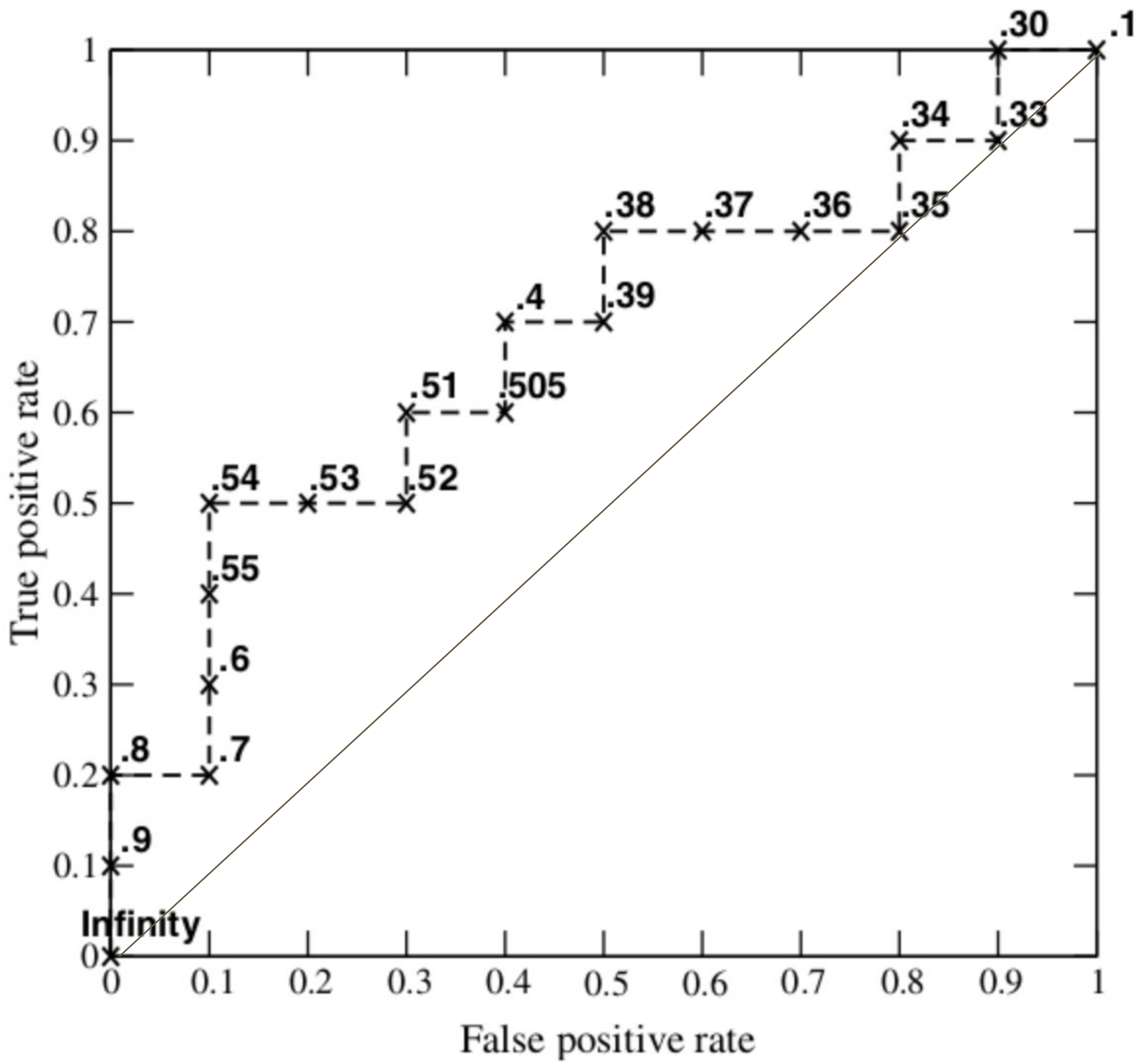
$$\mathbf{F1} = 2 * \frac{1}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Receiver Operating Characteristic (ROC) curve

		Actual value	
		True	False
Predicted value	True	True positive (TP)	False positive (FP)
	False	False negative (FN)	True negative (TN)

True positive rate: TPR=TP/(TP+FN)

False positive rate: FPR=FP/(FP+TN)



Let's evaluate the ranking as well

- A RE often delivers numerous outputs, while only a portion of them is most relevant to what the user really wants.

- We **rank our results** for the users, so the entries that the user would most likely selected would be near the top.

- How do we evaluate the *ranking* of results? 🤔

鍾欣怡

Search

（鍾欣桐）

（鍾欣凌）

1

2

3

4

Relevance (0-3 scale):

1

3

3

1

Cumulative Gain: 1 + 3 + 3 + 1 = 8
(@ rank 4)

Discounted Cumulative Gain (DCG):
(@ rank *p*)

$$\sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

$$\frac{1}{\log_2(1+1)} + \frac{3}{\log_2(2+1)} + \frac{3}{\log_2(3+1)} + \frac{1}{\log_2(4+1)} = 4.824$$

$$\frac{3}{\log_2(1+1)} + \frac{3}{\log_2(2+1)} + \frac{1}{\log_2(3+1)} + \frac{1}{\log_2(4+1)} = 5.824$$

IDCG :最理想化的分數

Normalized DCG:
(@ rank *p*)

$$\frac{DCG_p}{IDCG_p}$$