UNIVERSIDADE CATÓLICA PORTUGUESA

BRAGA

Behavior Analysis Technologies

Session 19

Introduction to Recommender Systems

Applied Data Science

2024/2025

From Search to Recommendation

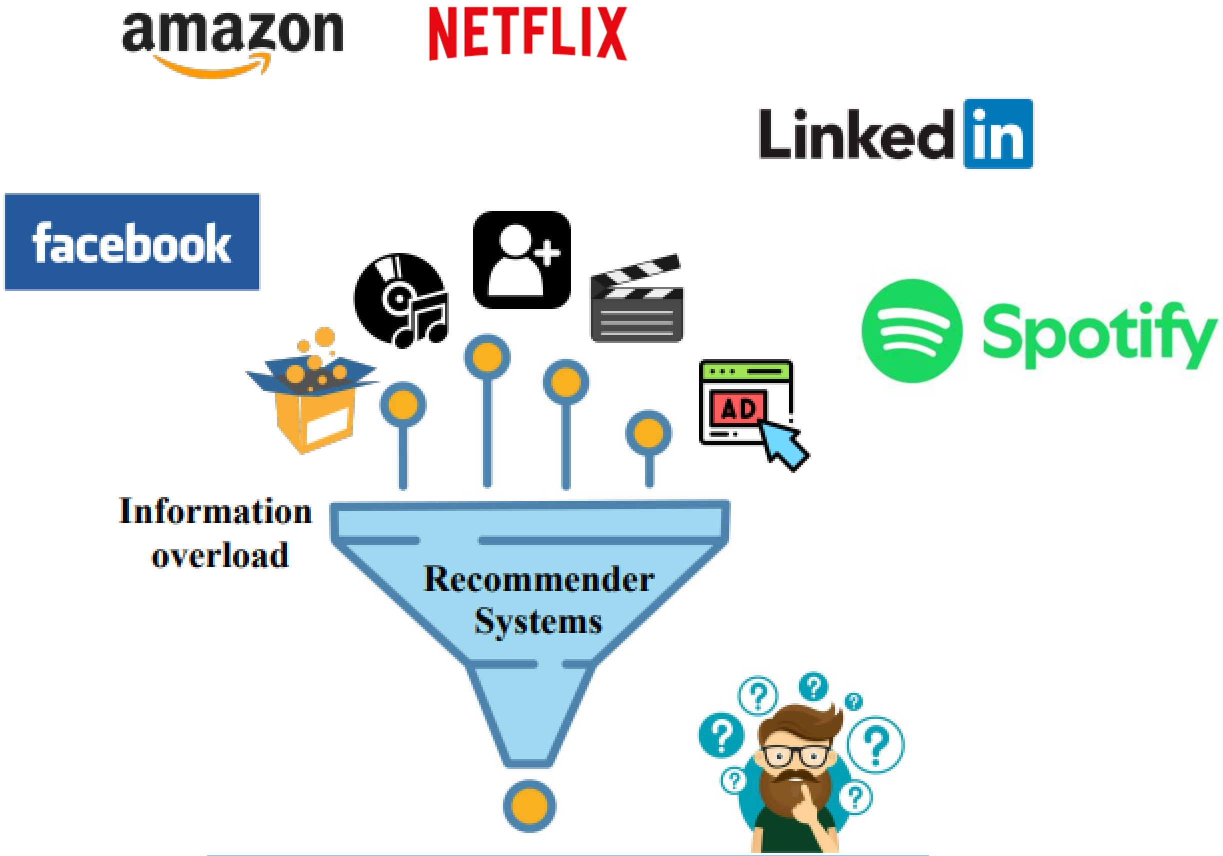
UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

"The Web is leaving the era of search and entering one of discovery. What's the difference?

Search is what you do when you're looking for something. **Discovery** is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you." – CNN Money, "The race to create a 'smart' Google

Recommender Systems

Age of Information Explosion



Recommend item X to user

Items can be: Products, Friends, News, Movies, Videos, etc.

Recommender Systems

- Recommendation has been widely applied in online services:
  - **E-commerce,** Content Sharing, Social Networking …

Product Recommendation

Amazon's recommendation algorithm drives **35%** of its sales [from McKinsey, 2012]

Recommender Systems

- Recommendation has been widely applied in online services:
  - E-commerce, **Content Sharing**, Social Networking ...

YouTube

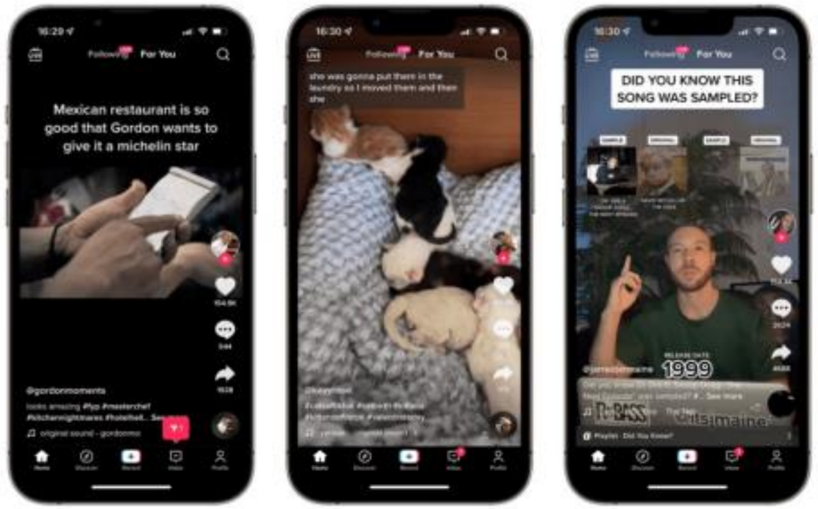
TikTok

Google News

News/Video/Image Recommendation

TikTok's recommendation algorithm
Top 10 Global Breakthrough Technologies in 2021

MIT Technology Review



The "Recommender Problem"

Estimate a **utility function**

to **predict** how

a user will **like** an item.

A Good Recommendation…

- Is relevant to the iser: personalized

A Good Recommendation...

- Is diverse:
  - It representes multiple possible interests of one user

A Good Recommendation

- Does not recommend items the user already knows.
- Expandas the user's taste into neighboring areas.

Top K Recommendations

- Users take into account only few suggestions.

- There is a need to do better on the top scoring recommended items.

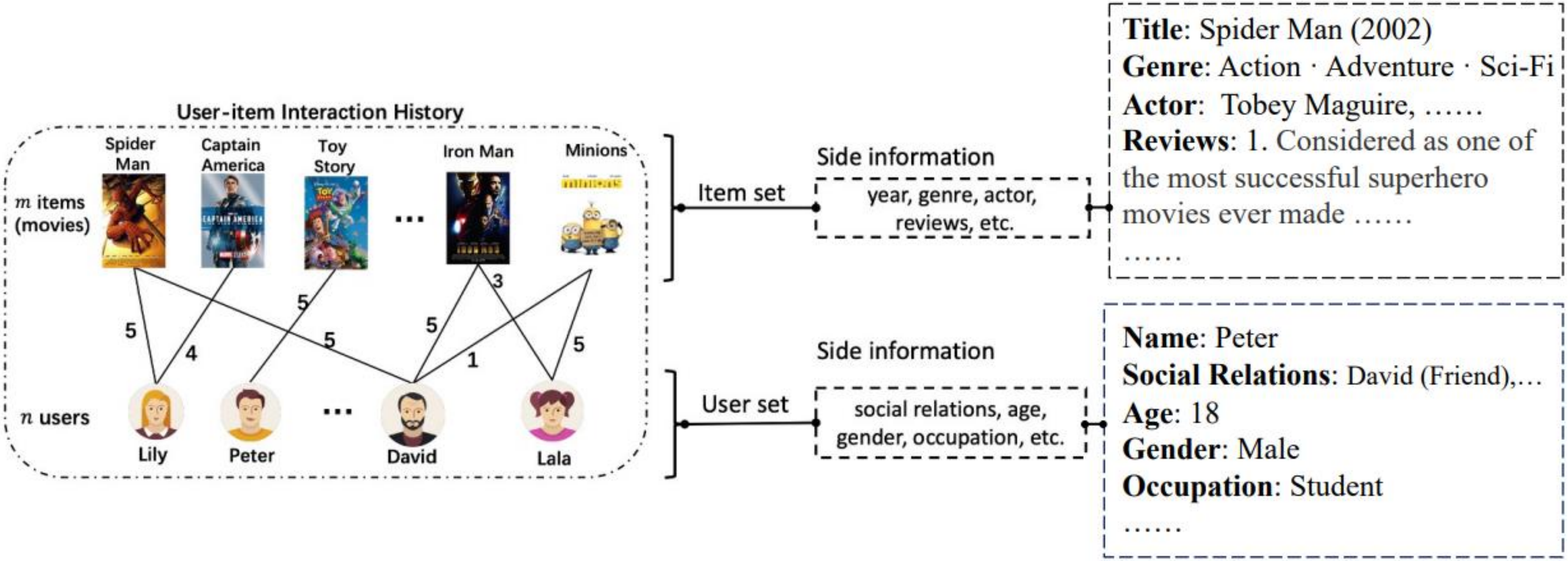
Recommender Systems

INPUT

Historical user-item interactions or additional side information (e.g., social relations, item's knowledge, etc.)

OUTPUT

Predict how likely a user would interact with a target Item (e.g., click, view, or purchase)



Recommender Systems Setup

- You have n users and m items in your system
  - Typically, n >> m. E.g., Youtube: 2.6B users, 800M videos

- Based on the content, we have a way of measuring user preference

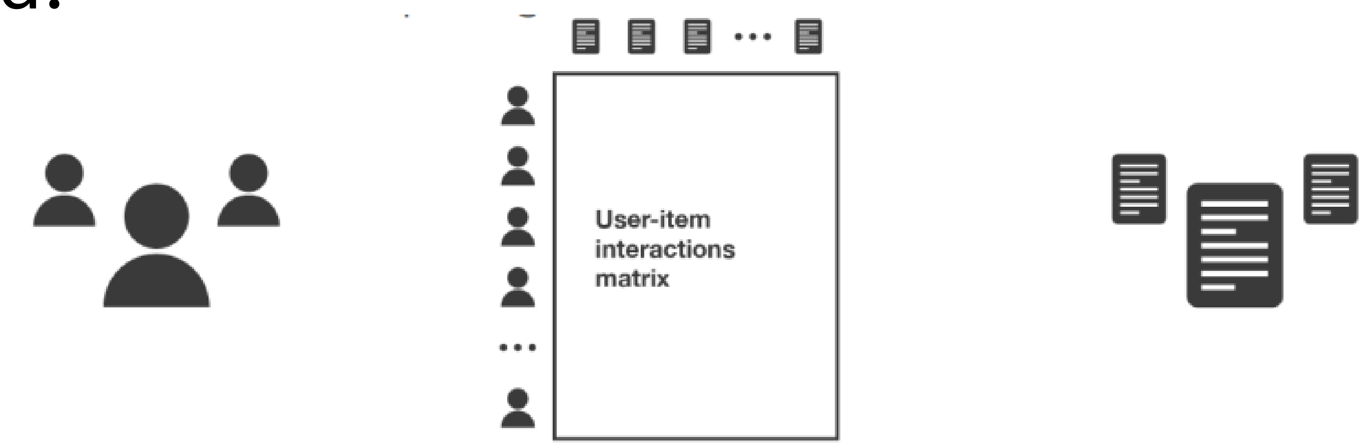
- This data is put together into a **user-item interaction matrix.**

- **Task**: Given a user u_i or item v_j, predict one or more items to recommend.

Recommender Systems Setup

- This data is put together into a **user-item interaction matrix.**

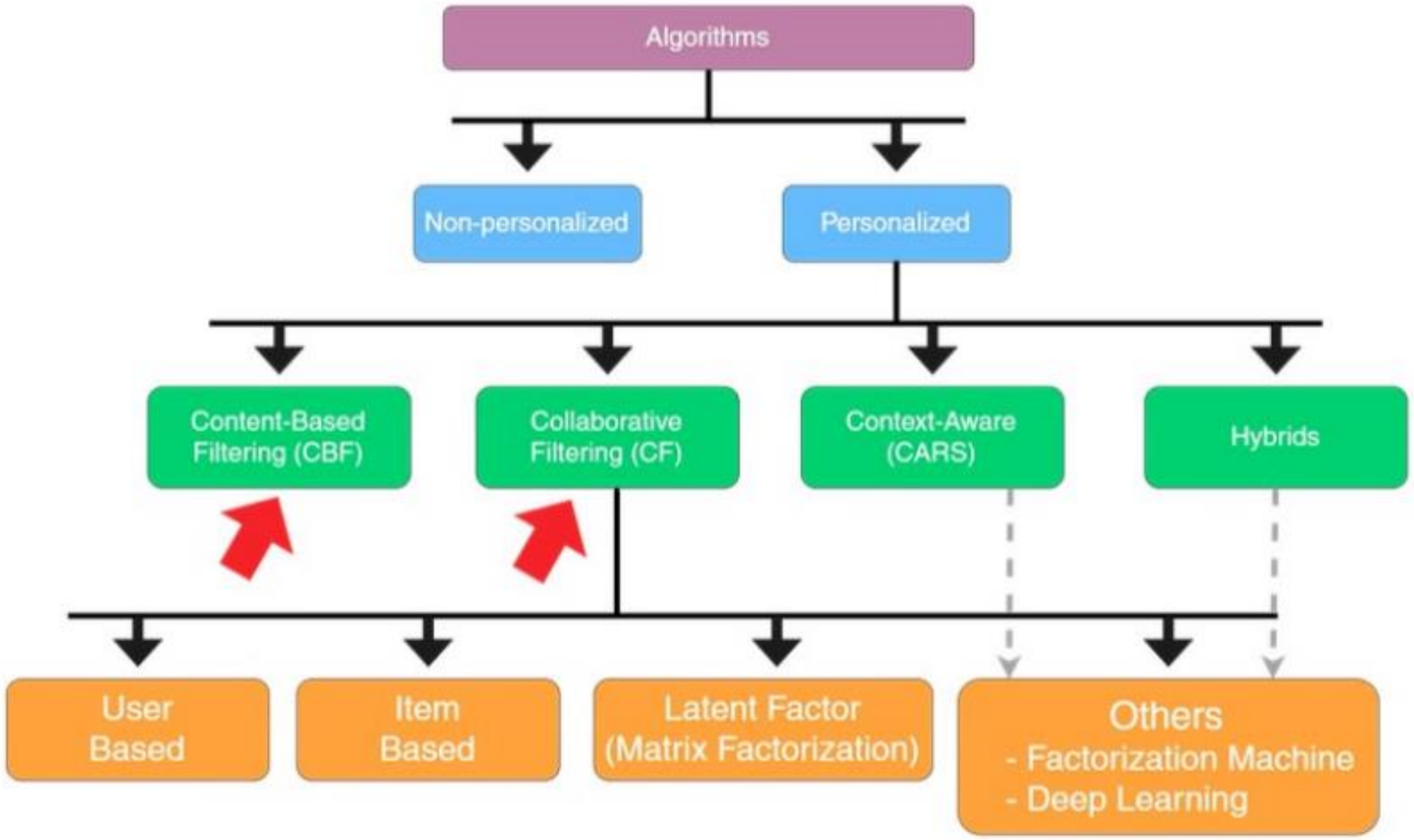
- **Task**: Given a user u_i or item v_j, predict one or more items to recommend.



Users	User-item interactions matrix	Items
suscribers	rating given by a user to a movie (integer)	movies
readers	time spent by a reader on an article (float)	articles
buyers	product clicked or not when suggested (boolean)	products

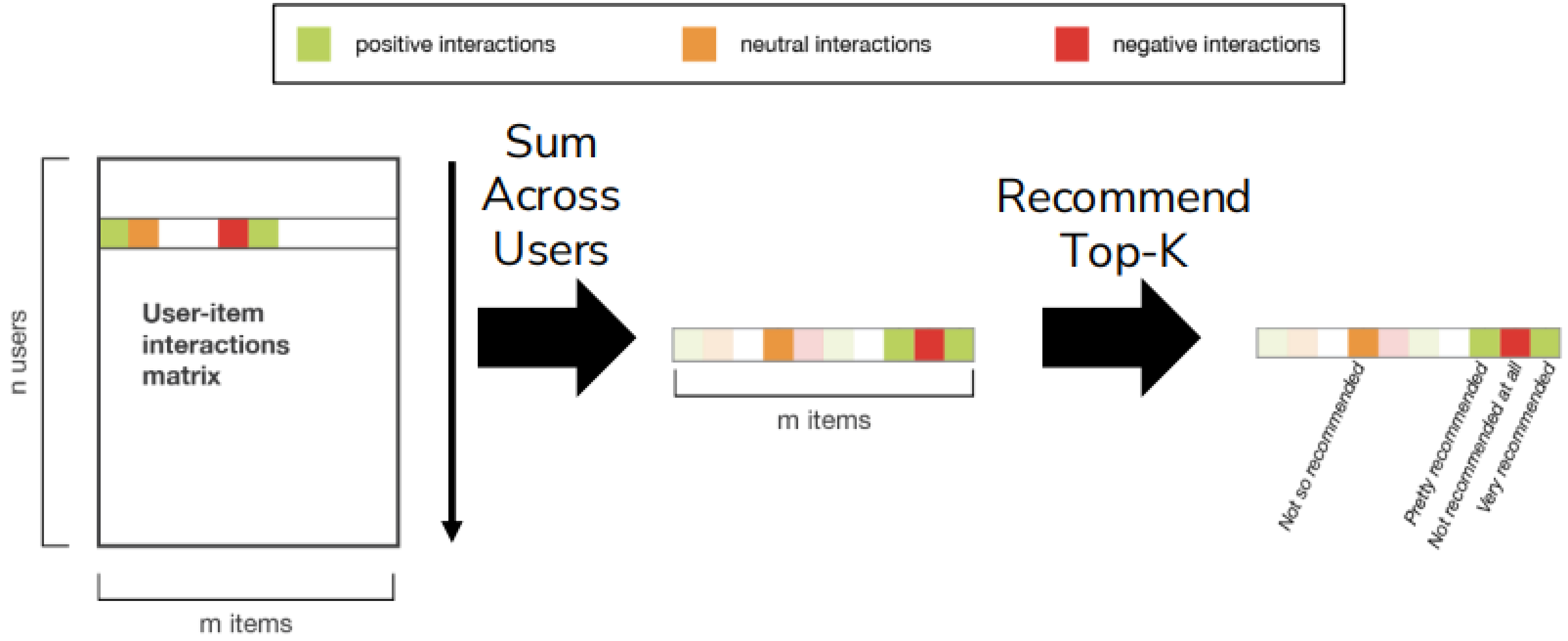
...

Recommender Systems Taxonomy



Non-Personalized Recommender Systems

- Simplest approach: **Just recommend whatever is popular**
  - Rank by global **popularity** (i.e. Avengers Endgame)



Non-Personalized Recommender Systems

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Pros:
  - Easy to Implement
- Cons:
  - No personalization
  - Feedback Loops
  - Top-K recommendations might be redundant
    - e.g., when a new Harry Potter is released, the others may also jump into top-k popularity

Content-Based Recommendations

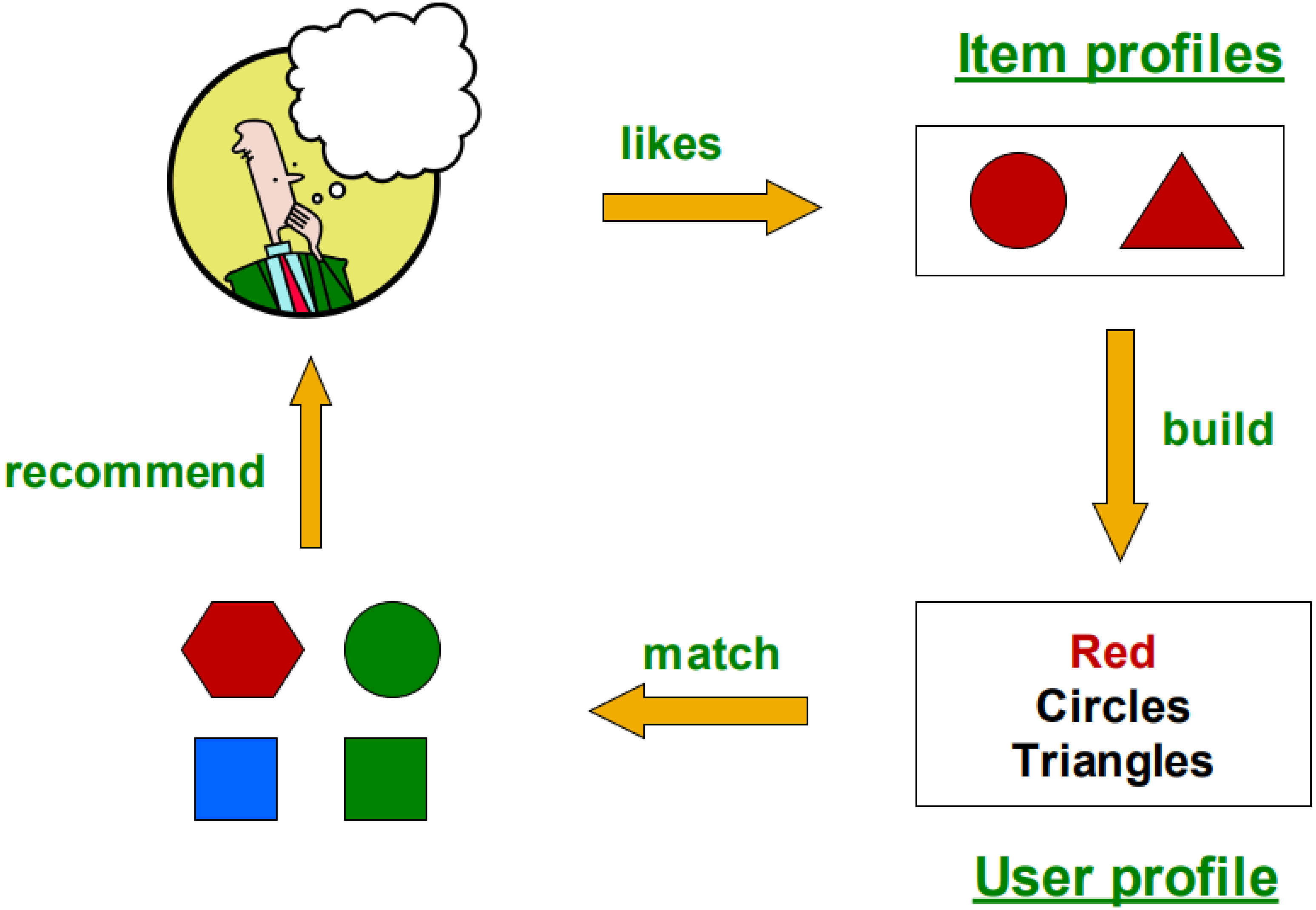
- Basic approach to recommendation

- Compare items based on attributes
  - Items have profiles:
    - Movies --> [genre, director, actors, plot, release year]
    - News --> [set of keywords]

- **User that liked an item is likely to like similar items.**

- Example:
  - Movie recommendations:
    - Recommend movies with same actor(s), director, genre, etc.
  - Websites, blogs, news:
    - Recommend other sites with "similar" content

Content-Based Recommendations



Content-Based Recommendations

- **User profile** possibilities:
  - Weighted average of rated item profiles
  - Variation: weight by difference from average rating for item

- Prediction heuristic: **Cosine similarity** of user and item profiles
  - Given user profile **x** and item profile **i**, estimate

$$u(\boldsymbol{x}, \boldsymbol{i}) = \cos(\boldsymbol{x}, \boldsymbol{i}) = \frac{\boldsymbol{x} \cdot \boldsymbol{i}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{i}||}$$

Content-Based Recommendations

- **Pros:**
  - **No need for data on other users**
    - No item cold-start problem, no sparsity problem
  - **Able to recommend to users with unique tastes**
  - **Able to recommend new and unpopular items**
    - No first-rater problem
  - **Able to provide explanations**
    - Can provide explanations of recommended items by listing content-features that caused na item to be recommended

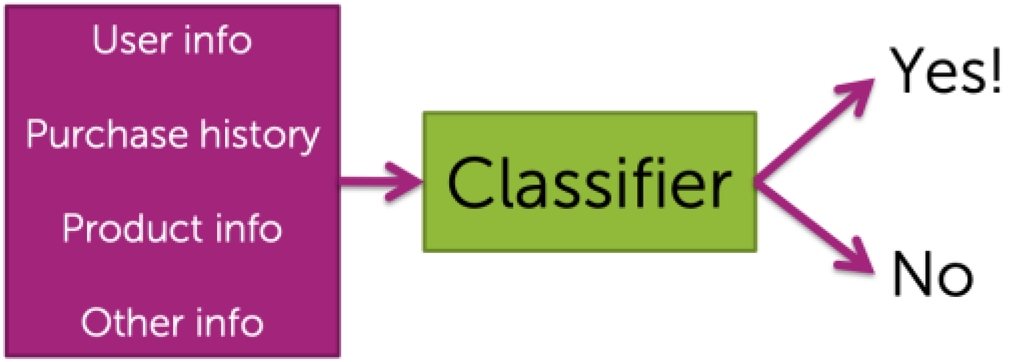
Content-Based Recommendations

UNIVERSIDADE CATÓLICA PORTUGUESA BRAGA

- **Cons:**
  - **Finding the appropriate features is hard**
    - E.g., images, movies, music
  - **Recommendations for new users**
    - How to build a user profile?
  - **Overspecialization**
    - Never recommends items outside user's content profile
    - People might have multiple interests
    - Unable to exploit quality judgments of other users

Content-Based Recommendations: Classification Model

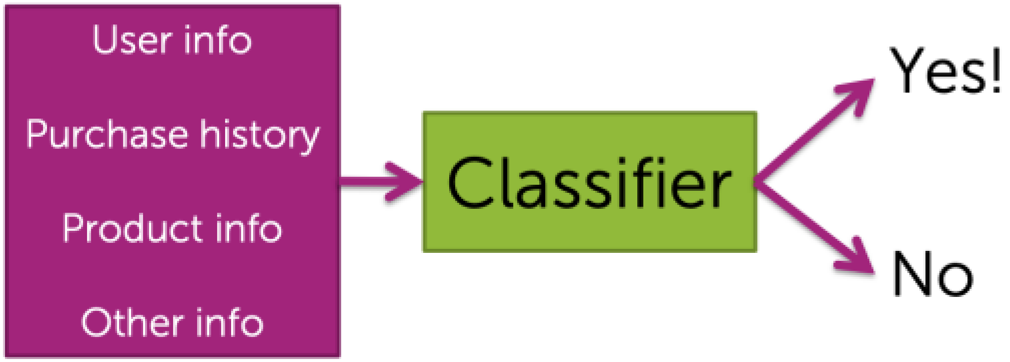
- Train a classifier to learn whether or not someone will like an item.



- Pros:
  - Personalized
  - Features can capture context (time of day, recent history, ...)
  - Can even handle limited user history (age of user, location, ...)

Content-Based Recommendations: Classification Model

- Train a classifier to learn whether or not someone will like an item.

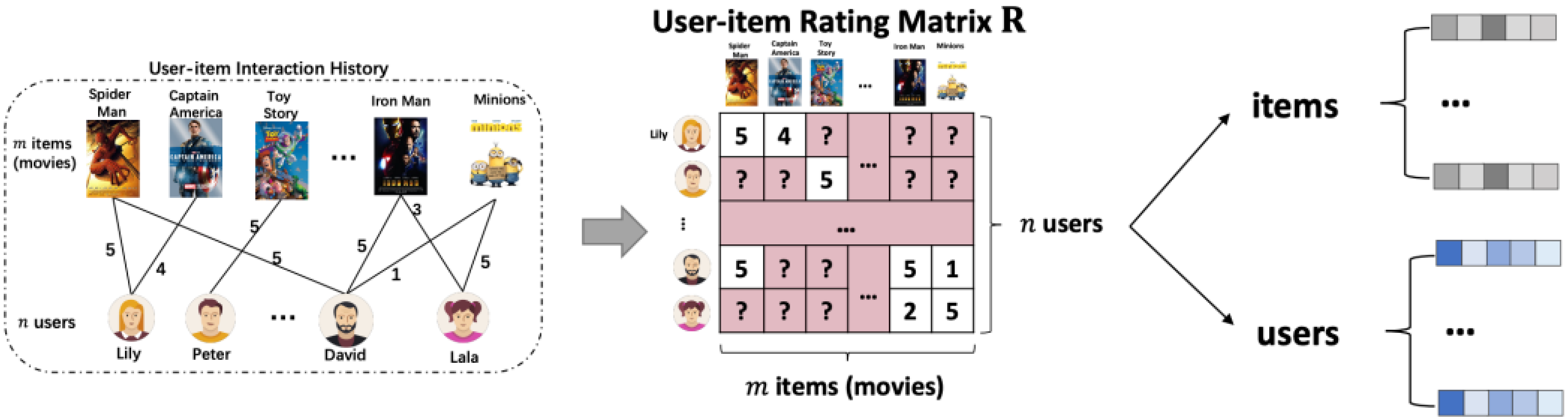


- Cons:
    - Features might not be available or hard to work with
    - Often doesn’t perform well in practice when compared to more advanced techniques like **collaborative filtering**
    - Can still lead to feedback loops

Collaborative Filtering

- The most well-known set of techniques for recommendation
  - Similar users (with respect to their historical interactions) have similar preferences.
  - Modelling user's preferences on items based on their past interactions (e.g., ratings and clicks).

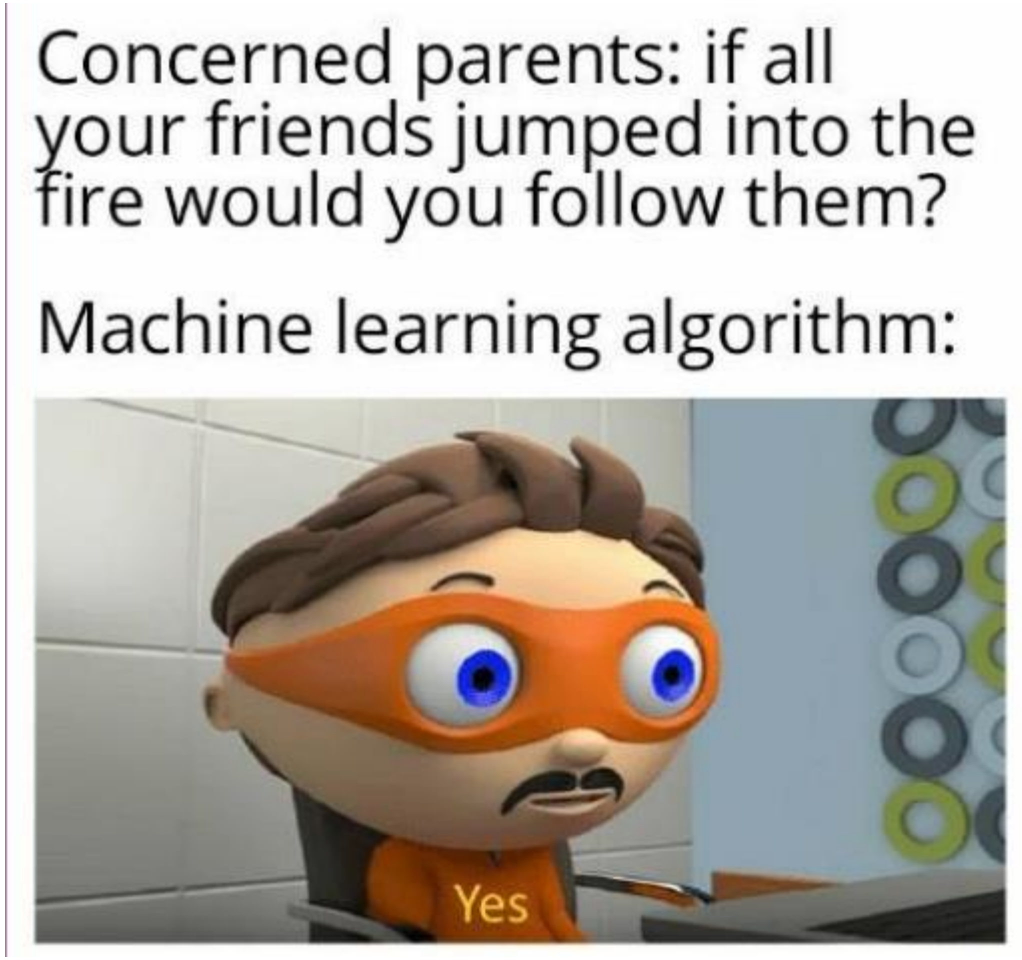
- Learning representations of users and items is the key



Task: predicting missing movie ratings in Netflix.

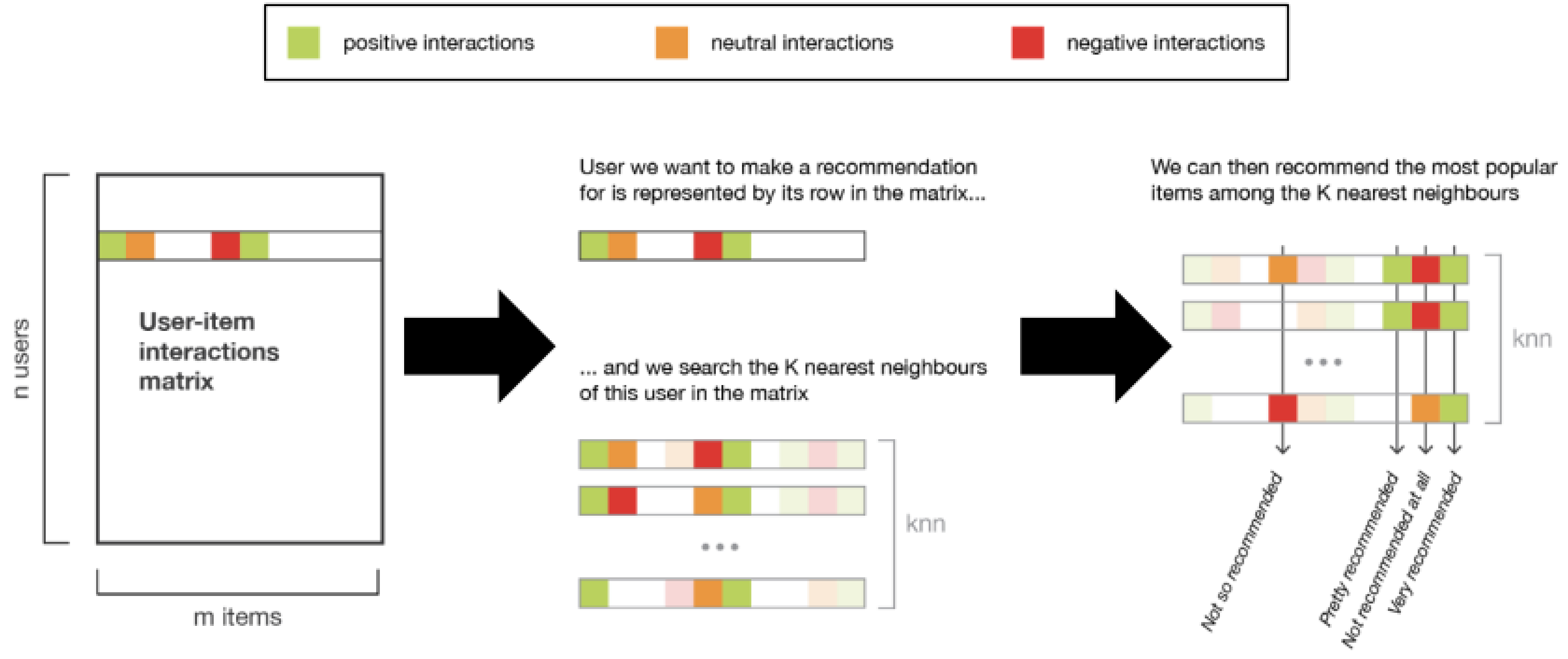
Collaborative Filtering: User-User

- **Nearest User**



Collaborative Filtering: Nearest-User

- **User-User Recommendation:**
  - Given a user u_i, compute their k nearest neighbors.
  - Recommend items that are most popular amongst the nearest neighbors.



Collaborative Filtering: User-User

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- **Pros:**
  - Personalized to the user

- **Cons**:
  - Nearest Neighbors might be too similar
    - This approach only works if the nearest neighbors have interacted with items that the user hasn't.
  - Feedback Loop (Echo Chambers)
  - Scalability
    - Must store and search through entire user-item matrix
  - Cold-Start Problem
    - What do you do about new users, with no data?

Collaborative Filtering: Item-Item

- People Who Bought This Also Bougth...
- Item-Item Recommendation:
  - Create a co-occurrence matrix of items that are bought together.

	Sunglasses	Baby Bottle	...	Diapers	Swim Trunks	Baby Formula
Sunglasses	500	15	...	9	130	20
Baby Bottle	15	45	...	6	10	10
...	...	...	...	...	...	...
Diapers	9	6	...	30	9	6
Swim Trunks	130	10	...	9	200	8
Baby Formula	20	10	...	6	8	50

Collaborative Filtering: Item-Item

- Problem: popular items drown out the rest!
- Solution: Normalizing using Jaccard Similarity.

$$S_{ij} = \frac{\#\text{ purchased } i \textbf{ and } j}{\#\text{ purchased } i \textbf{ or } j} = \frac{C_{ij}}{C_{ii} + C_{jj} - C_{ij}}$$

	Sunglasses	Baby Bottle	...	Diapers	Swim Trunks	Baby Formula
Sunglasses	1.00	0.03	...	0.02	0.23	0.04
Baby Bottle	0.03	1.00	...	0.09	0.04	0.12
...	...	...	...	...	...	...
Diapers	0.02	0.09	...	1.00	0.04	0.08
Swim Trunks	0.23	0.04	...	0.04	1.00	0.03
Baby Formula	0.04	0.12	...	0.08	0.03	1.00

Collaborative Filtering: Item-Item

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Incorporating Purchase History:
  - What if I know the user u has bought a baby bottle and formula?
  - **Idea**: Take the average similarity between items they have bought

$$Score(u, diapers) = \frac{s_{diapers,baby\ bottle} + s_{diapers,baby\ formula}}{2}$$

  - Could also weight them differently based on recency of purchase!
  - Then all we need to do is find the item with the highest average score!

Collaborative Filtering: Item-Item

- Pros:
  - Personalizes to item (incorporating purchase history also personalizes to the user)

- Cons:
  - Can still suffer from feedback loops
    - (As can all recommender systems – but in some cases it's worse than others)
  - Scalability (must store entire item-item matrix)
  - Cold-Start Problem
    - What do you do about new items, with no data?

Collaborative Filtering: Feature-Based

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- What if we know what factors lead users to like an item?

- **Idea**: Create a feature vector for each item. Learn a regression model!

Genre	**Year**	**Director**	**...**
Action	1994	Quentin Tarantino	...
Sci-Fi	1977	George Lucas	...

- Define weights on these features for all users (global) $w_G \in \mathbb{R}^d$

- Fit a linear model.

Collaborative Filtering: Feature-Based

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- What if we know what factors lead users to like an item?

- **Idea**: Create a feature vector for each item. Learn a regression model!

Genre	Year	Director	...
Action	1994	Quentin Tarantino	...
Sci-Fi	1977	George Lucas	...

- Define weights on these features for all users (global) $w_G \in \mathbb{R}^d$

$$\hat{r}_{uv} = w_G^T h(v) = \sum_i w_{G,i}\, h_i(v)$$

- Fit a linear model.

$$\hat{w}_G = argmin_w \frac{1}{\#\, ratings} \sum_{u,v: r_{uv} \neq ?} (w_G^T h(v) - r_{uv})^2 + \lambda \|w_G\|$$

Collaborative Filtering: Feature-Based

- Personalization:
  - Add user-specific features to the feature vector!

Genre	Year	Director	...	Gender	Age	...
Action	1994	Quentin Tarantino	...	F	25	...
Sci-Fi	1977	George Lucas	...	M	42	...

Collaborative Filtering: Feature-Based

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Pros:
  - No cold-start issue!
    - Even if a new user/item has no purchase history, you know features about them.
  - Personalizes to the user and item
  - Scalable (only need to store weights per feature)
  - Can add arbitrary features (e.g., time of day)

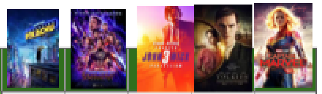
- Cons:
  - Hand-crafting features is very tedious and unscalable.

Matrix Factorization

- Want to recommend movies based on user ratings for movies.

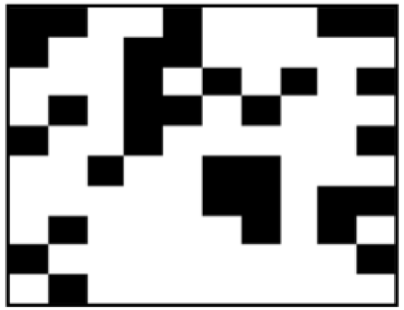
- **Challenge**: Users have rated relatively few of the entire catalog.

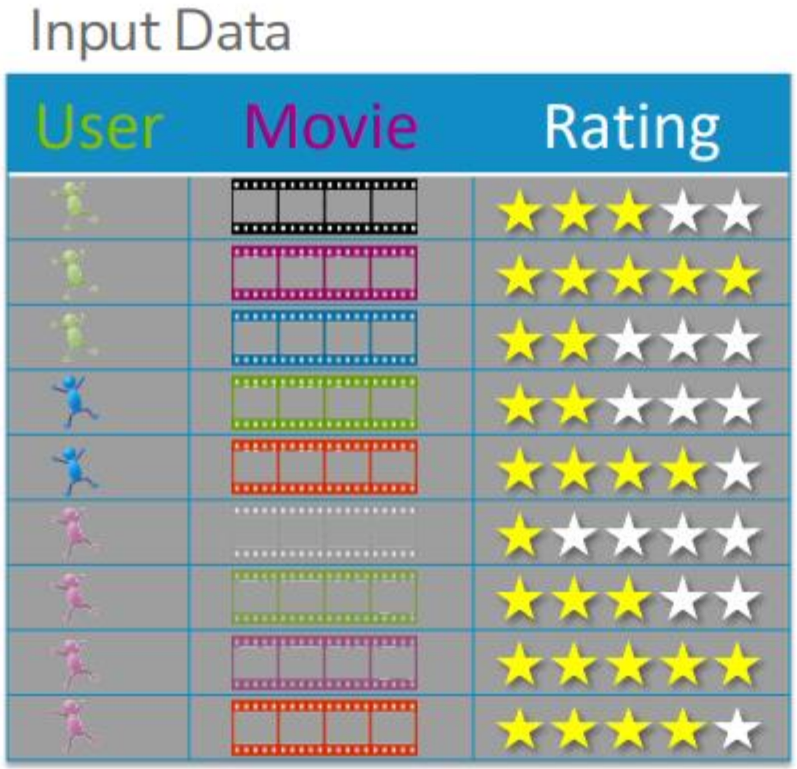
- Can think of this as a matrix of users and ratings with missing data!



User 1	5				3
User 2		2		4	
User 3			3		
User 4	1				
User 5			4		
User 6		5			2

=





Matrix Factorization

- Assumptions:
  - Assume that each item has k (unknown) features.
    - e.g., k possible genres of movies (action, romance, sci-fi, etc.)
  - Then, we can describe an item v with feature vector R_v
    - How much is the movie action, romance, sci-fi, ...
    - e.g., $R_v = [0.3, 0.01, 1.5, \ldots]$
  - We can also describe each user u with a feature vector L_u
    - How much they like action, romance, sci-fi, ...
    - Example: $L_u = [2.3, 0 , 0.7 , \ldots]$
  - Estimate rating for user u and movie v as:
    - $\text{Rating}(u, v) = L_u * R_v = 2.3 * 0.3 + 0*0.01 + 0.7*1.5 + \ldots$

Matrix Factorization

- Example:
  - Suppose we have learned the following user and movie features using 2 features

User ID	Feature
1	(2, 0)
2	(1, 1)
3	(0, 1)
4	(2, 1)

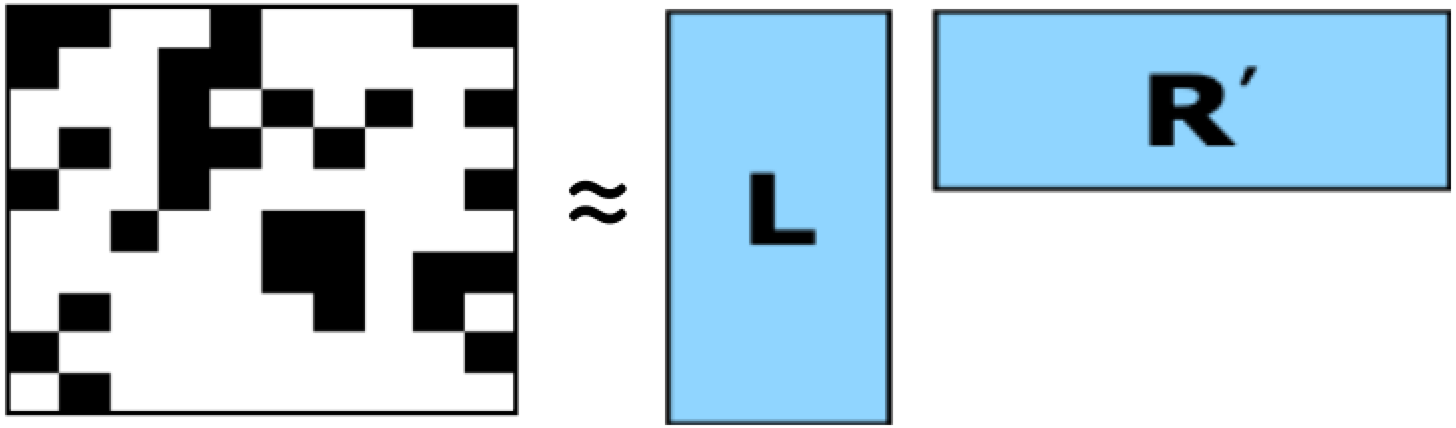
Movie ID	Feature vector
1	(3, 1)
2	(1, 2)
3	(2, 1)

  - Then we can predict what each user would rate each movie

$$\underset{L}{\begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 0 & 1 \\ 2 & 1 \end{bmatrix}} \underset{R^T}{\begin{bmatrix} 3 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}} = \begin{bmatrix} 6 & 2 & 4 \\ 4 & 3 & 3 \\ 1 & 2 & 1 \\ 7 & 4 & 5 \end{bmatrix}$$

Matrix Factorization

Rating =



- Goal: Find L_u and R_v that when multiplied, achieve predicted ratings that are close to the values that we have data for

- Our quality metric will be (could use others):

$$\hat{L},\hat{R} = \underset{L,R}{\operatorname{argmin}} \sum_{u,v:r_{uv}\neq ?} \left(L_u \cdot R_v - r_{uv}\right)^2$$

Matrix Factorization

- Is this problem well posed? Unfortunately, there is not a unique solution.

- For example, assume we had a solution:

$$\begin{bmatrix}6&2&4\\4&3&3\\1&2&1\\7&4&5\end{bmatrix} = \overset{L}{\begin{bmatrix}2&0\\1&1\\0&1\\2&1\end{bmatrix}}\ \overset{R^T}{\begin{bmatrix}3&1&2\\1&2&1\end{bmatrix}}$$

Matrix Factorization

- Is this problem well posed? Unfortunately, there is not a unique solution.

- Then doubling everything in L and halving everything in R is also a valid solution. The same is true for all constant multiples.

6	2	4
4	3	3
1	2	1
7	4	5

=

L

4	0
2	2
0	2
4	2

R^T

1.5	0.5	1.0
0.5	1.0	0.5

Collaborative Filtering

- **Pros**:
  - Works for any kind of item
    - No feature selction needed

- **Cons**:
  - Cold start:
    - Nedd enough users in the system to find a match
  - Sparsity:
    - The user/ratings matrix is sparse
    - Hard to find users that rated the same items
  - First rater:
    - Cannot tecommend na item that has not been previously rated
  - Popularity bias:
    - Cannot recommend items to someone with unique taste
    - Tends to recommend popular items

Hybrid Methods

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Implement two or more different recommenders and combine predictions
  - Perhaps using a linear model

- Add content-based methods to collaborative filtering
  - Item profiles for new item problem
  - Demographics to deal with new user problem

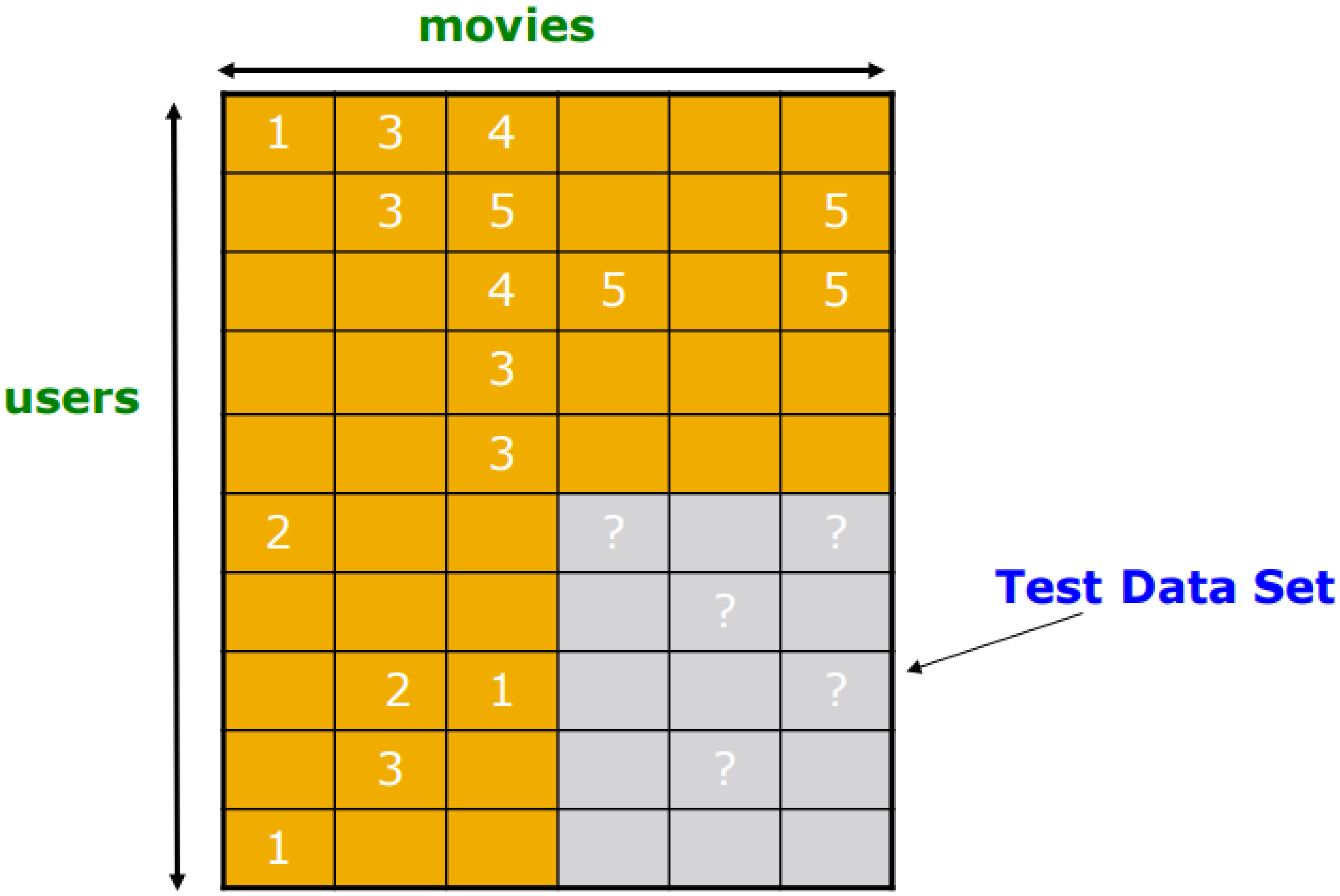
Evaluation

movies

users

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

Evaluation



Evaluating Predictions

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Compare predictions with known ratings
  - Root Mean Squared Error

    $\sqrt{\frac{1}{N}\sum_{xi}\left(r_{xi} - r_{xi}^{*}\right)^2}$ where $\boldsymbol{r_{xi}}$ is predicted, $\boldsymbol{r_{xi}^{*}}$ is the true rating of $\boldsymbol{x}$ on $\boldsymbol{i}$

    - ***N is the number of points we are making comparisons on***
  - Precision at top X:
    - % of relevant items in top X
  - Accuracy (if predicting like/dislike or click/ignore)

- Another approach: 0/1 model
  - Coverage: number of items/users for wich the system can make predictions
  - Precision: Accuracy of predictions
  - Receiver operating characteristic: Tradeoff curve between false positives and false negatives

Problems with Error Measures

- Narrow focus on accuracy sometimes misses the point
  - Prediction Diversity
  - Prediction Context
  - Order of predictions

- In practice, we care only to predict high ratings:
  - RMSE might penalize a method that does well for high ratings and badly for others

Cold-Start Problem

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- When a new user comes in, we don’t know what items they like! When a new item comes into our system, we don’t know who likes it! This is called the **cold start** problem.

- Addressing the cold-start problem (for new users):
  - Give random predictions to a new user.
  - Give the globally popular recommendations to a new user.
  - Require users to rate items before using the service.
  - Use a feature-based model (or a hybrid between feature-based and matrix factorization) for new users.

Top-K vs Diverse Recommendations

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Top-k recommendations might be very redundant:
  - Someone who likes Rocky I also will likely enjoy Rocky II, Rocky III, Rocky IV, Rocky V
- Diverse recommendations:
  - Users are multi-faceted & we want to hedge our bets
  - Maybe recommend: Rocky II, Always Sunny in Philadelphia, Robin Hood
- Solution: Maximal Marginal Relevance
  - Pick recommendations one-at-a-time.
  - Select the item that the user is most likely to like and that is most dissimilar from existing recommendations.

Feedback Loops / Echo Chambers

- Users always get recommended similar content and are unable to discover new content they might like.
  - Exploration-Exploitation Dilemma
    - Common problem in “online learning” settings
  - Pure Exploration: show users random content
    - Users can discover new interests, but will likely be unsatisfied
  - Pure Exploitation: show users content they’re likely to like
    - Users can’t discover new interests

  - Find tradeoff between exploration and exploitation.