Text Analytics & Business Application

Recommendation Systems

Qinglai He

Department of Operations and Information Management

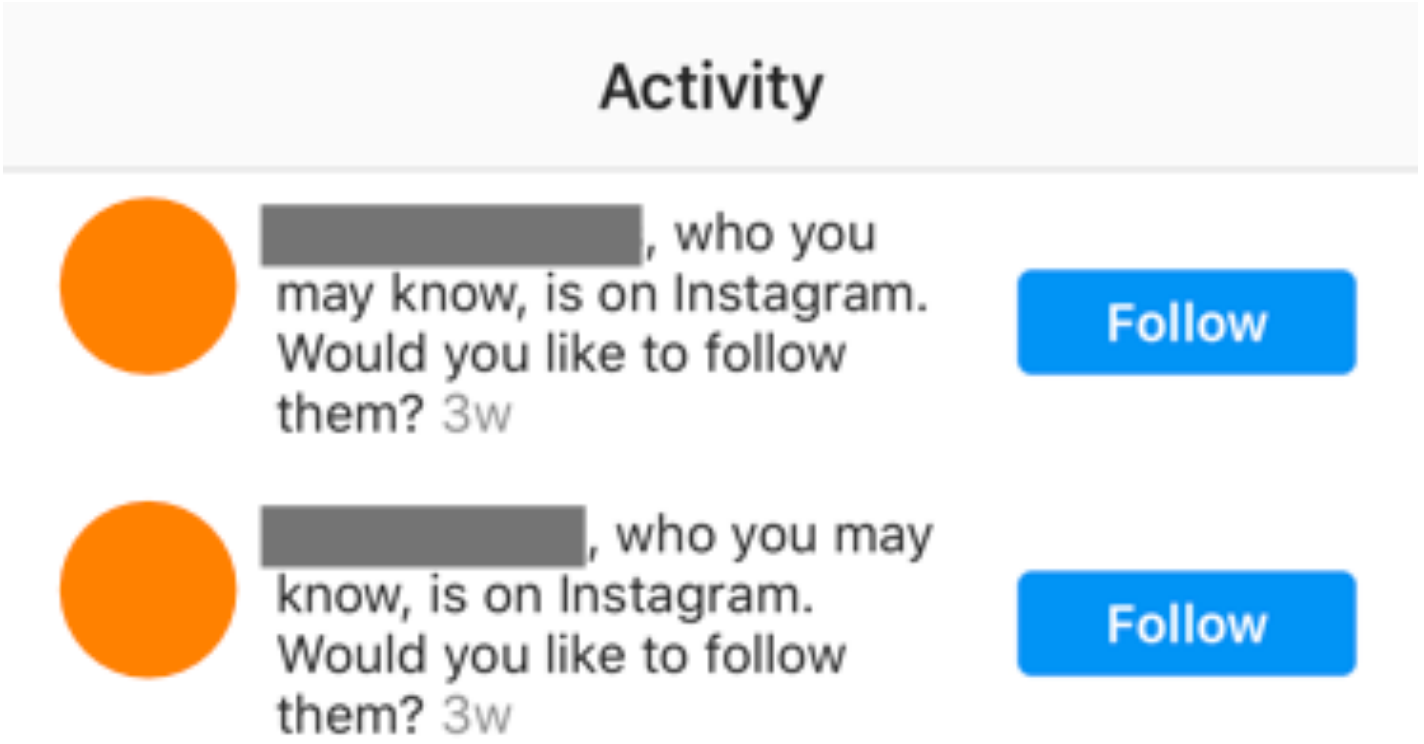
Wisconsin School of Business

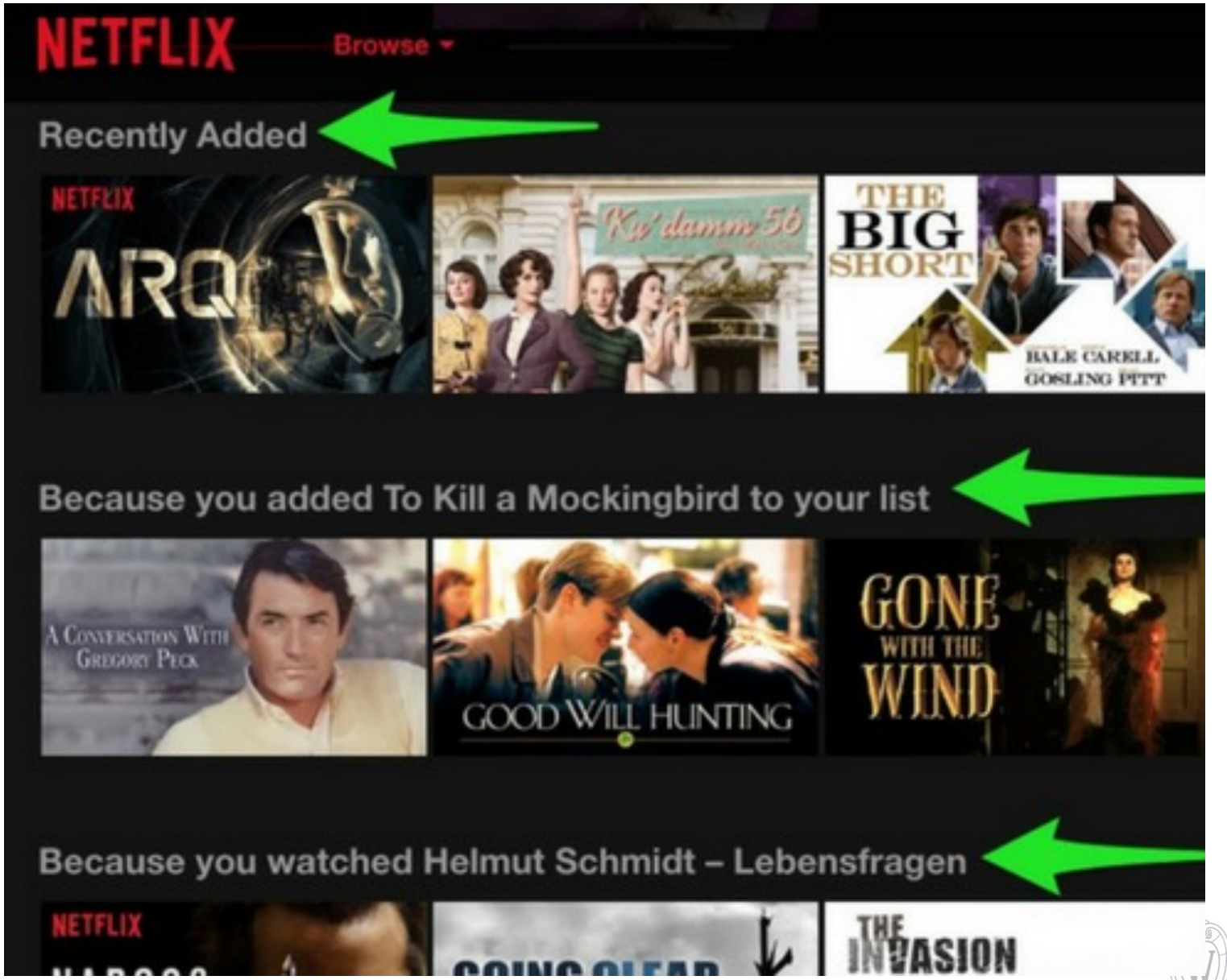
WISCONSIN SCHOOL OF BUSINESS

Outline of Today's Class

- Intro to Recommendation System (Rec Sys)
- Recap of Similarity Measures
- Two Types of Rec Sys
  - Content-based Recommendation
  - Collaborative Filtering (CF) Recommendation
    - User-based CF
    - Item-based CF


Instagram


Activity
[redacted], who you may know, is on Instagram. Would you like to follow them? 3w
Follow
[redacted], who you may know, is on Instagram. Would you like to follow them? 3w
Follow


NETFLIX
Browse
Recently Added
NETFLIX
ARQ
Kid dammy 56
THE BIG SHORT
BALE CARELL
GOSLING PITT
Because you added To Kill a Mockingbird to your list
A Conversation With Gregory Peck
GOOD WILL HUNTING
GONE WITH THE WIND
Because you watched Helmut Schmidt – Lebensfragen
NETFLIX
THE INVASION

Customers who searched for "headphones" ultimately bought

Apple EarPods with Lightning Connector - White

4.5 out of 5 stars 201,446

Amazon's Choice in Earbud & In-Ear Headphones

$18.53

Get it as soon as **Sunday, Apr 17**

FREE Shipping on orders over $25 shipped by Amazon

Wired Earbuds with Microphone 5 Pack, in-Ear Headphones with Heavy Bass, High Sound Quality Earphones...

4.5 out of 5 stars 1,240

$12.99

Get it as soon as **Sunday, Apr 17**

FREE Shipping on orders over $25 shipped by Amazon

Sony ZX Series Wired On-Ear Headphones, Black MDR-ZX110

4.5 out of 5 stars 80,188

Amazon's Choice in On-Ear Headphones

$9.99

Get it as soon as **Sunday, Apr 17**

FREE Shipping on orders over $25 shipped by Amazon

OneOdio Wired Over Ear Headphones Studio Monitor & Mixing DJ Stereo Headsets with...

4.5 out of 5 stars 31,702

#1 Best Seller in DJ Headphones

$31.99

Get it as soon as **Sunday, Apr 17**

FREE Shipping on orders over $25 shipped by Amazon

RORSOU R10 On-Ear Headphones with Microphone, Lightweight Folding Stereo Bass Headphones with 1.5...

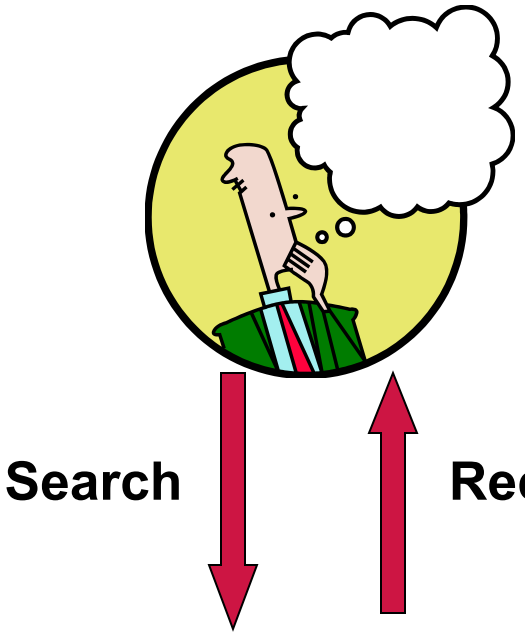
4.5 out of 5 stars 1,609

$16.99

Get it as soon as **Sunday, Apr 17**

FREE Shipping on orders over $25 shipped by Amazon

Recommendations



Products, web sites, blogs, news items, …

Examples:

amazon.com
PANDORA
StumbleUpon
del.icio.us
NETFLIX
movielens
helping you find the *right* movies
last·fm
the social music revolution
Google News
YouTube
XBOX LIVE

From Scarcity to Abundance

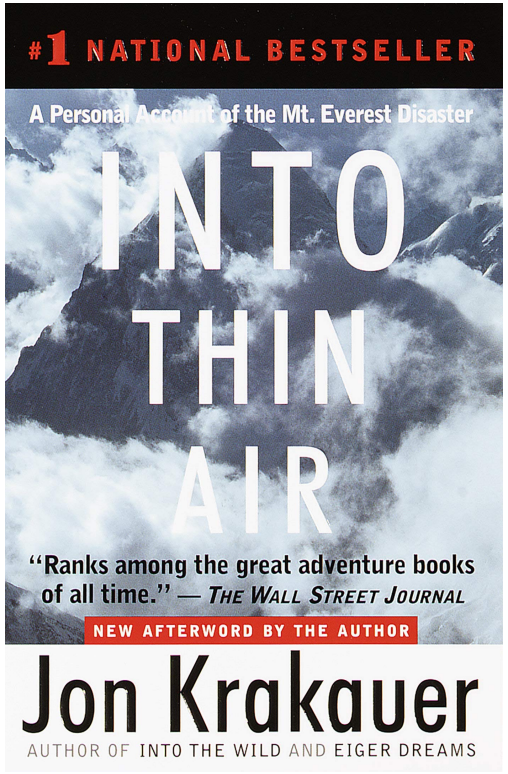
- Shelf space is a scarce commodity for traditional retailers
  - Also: TV networks, movie theaters,…
- Web enables near-zero-cost dissemination of information about products
  - From scarcity to abundance
- More choice necessitates better filters
  - Recommendation engines
  - How **Into Thin Air** made **Touching the Void**

  a bestseller:

  http://www.wired.com/wired/archive/12.10/tail.html

How *Into Thin Air* made *Touching the Void*

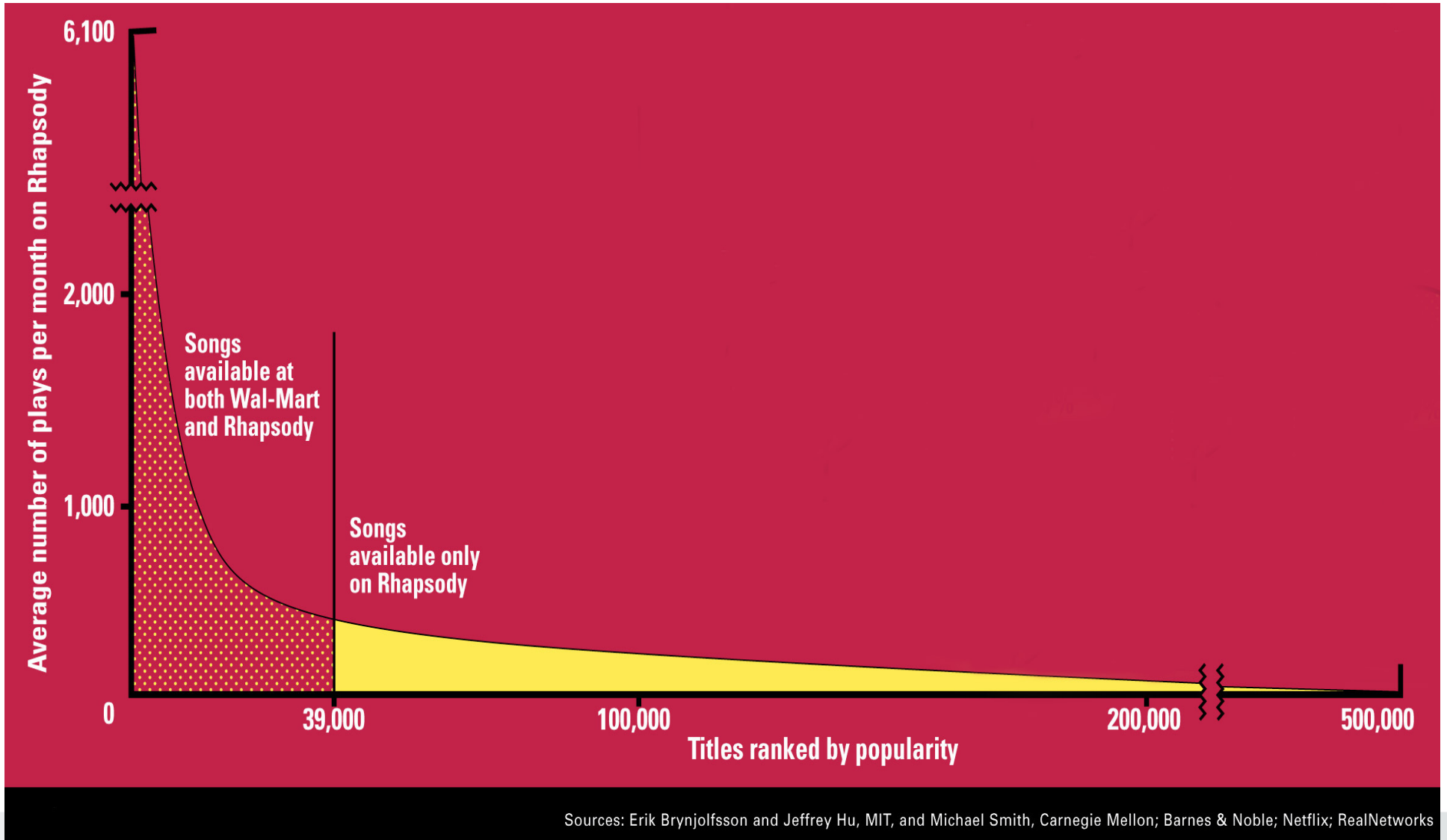
- In 1988, a British mountain climber named Joe Simpson wrote a book called *Touching the Void*, a harrowing account of near death in the Peruvian Andes.
- A decade later, a strange thing happened.
  - In 1997, Jon Krakauer wrote *Into Thin Air*, another book about a mountain-climbing tragedy, which became a publishing sensation.
  - Suddenly *Touching the Void* started to sell again.
- A revised paperback edition of *Touching the Void*, which came out in January the next year, spent 14 weeks on the *New York Times* bestseller list.
  - Now *Touching the Void* outsells *Into Thin Air* more than two to one.



What Happened?

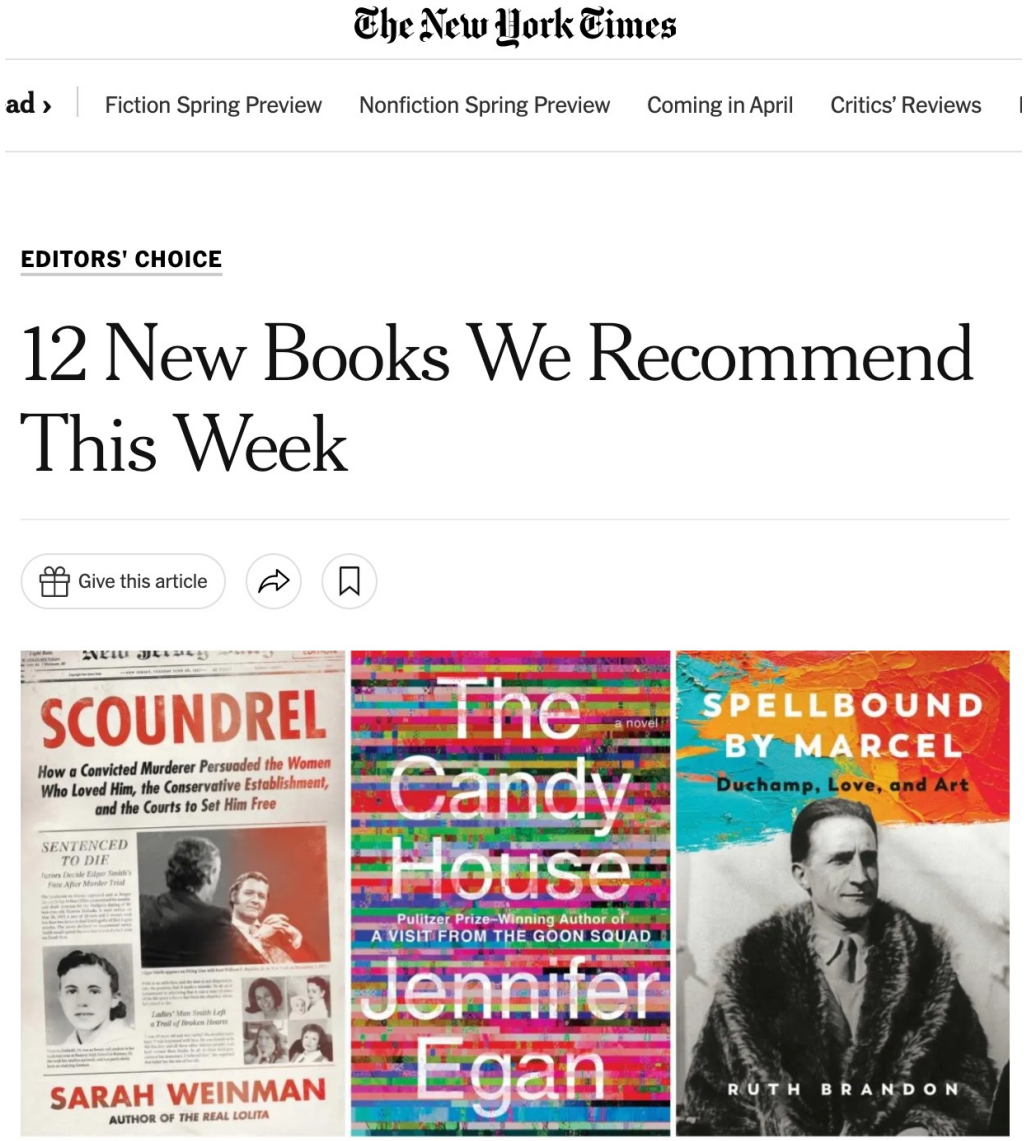
- In short, Amazon recommendations.
- The online bookseller's recommender system noted patterns in buying behavior and suggested that readers who liked *Into Thin Air* would also like *Touching the Void*.
  - People took the suggestion, agreed wholeheartedly, wrote rhapsodic reviews.
  - More sales, more algorithm-fueled recommendations, and the positive feedback loop kicked in.

Sidenote: The Long Tail



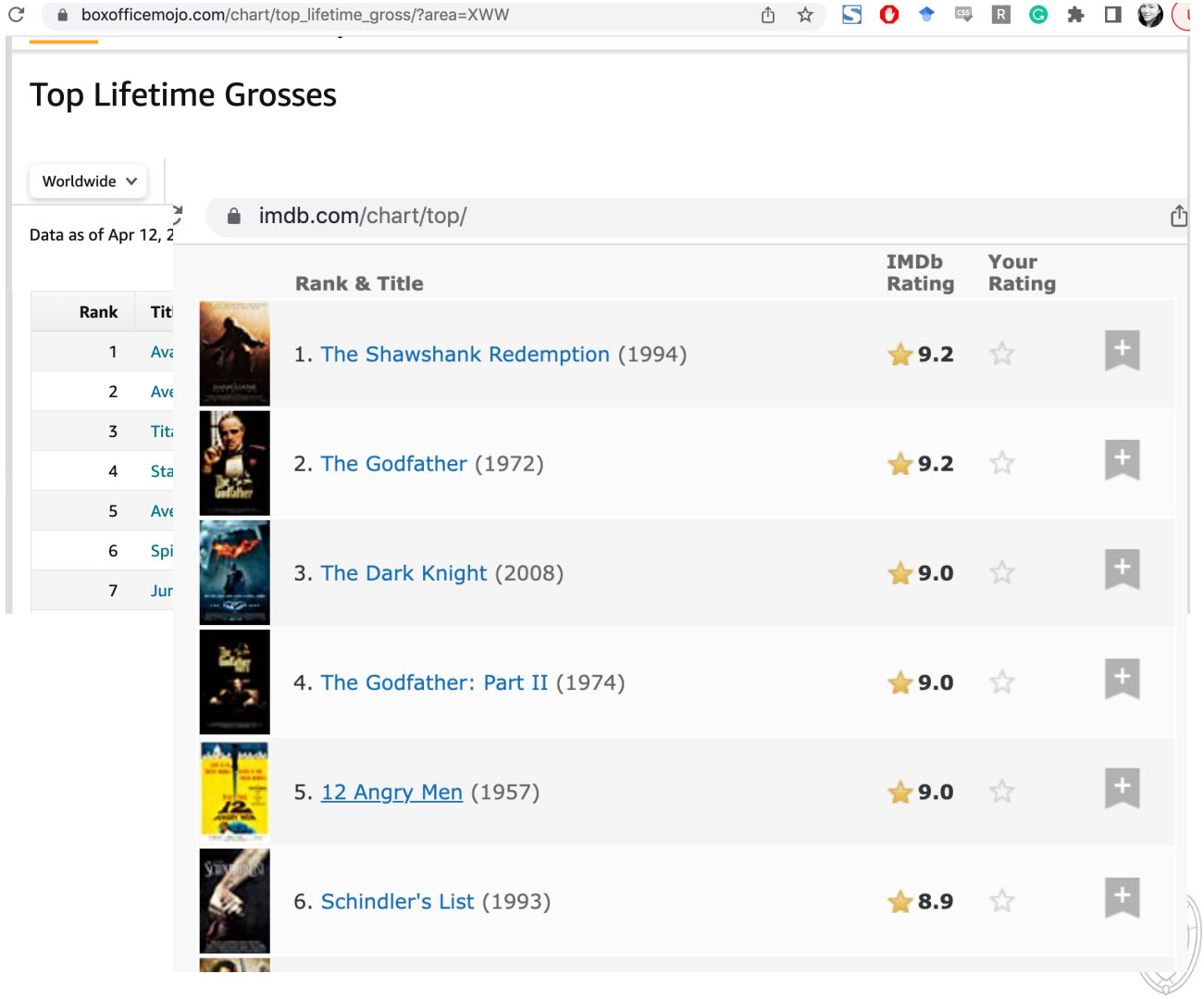
Types of Recommendations

- **Editorial and hand curated**
  - List of favorites
  - Lists of “essential” items
- **Simple aggregates**
  - Top 10, Most Popular, Recent Uploads
- **Tailored to individual users**
  - Amazon, Netflix, …



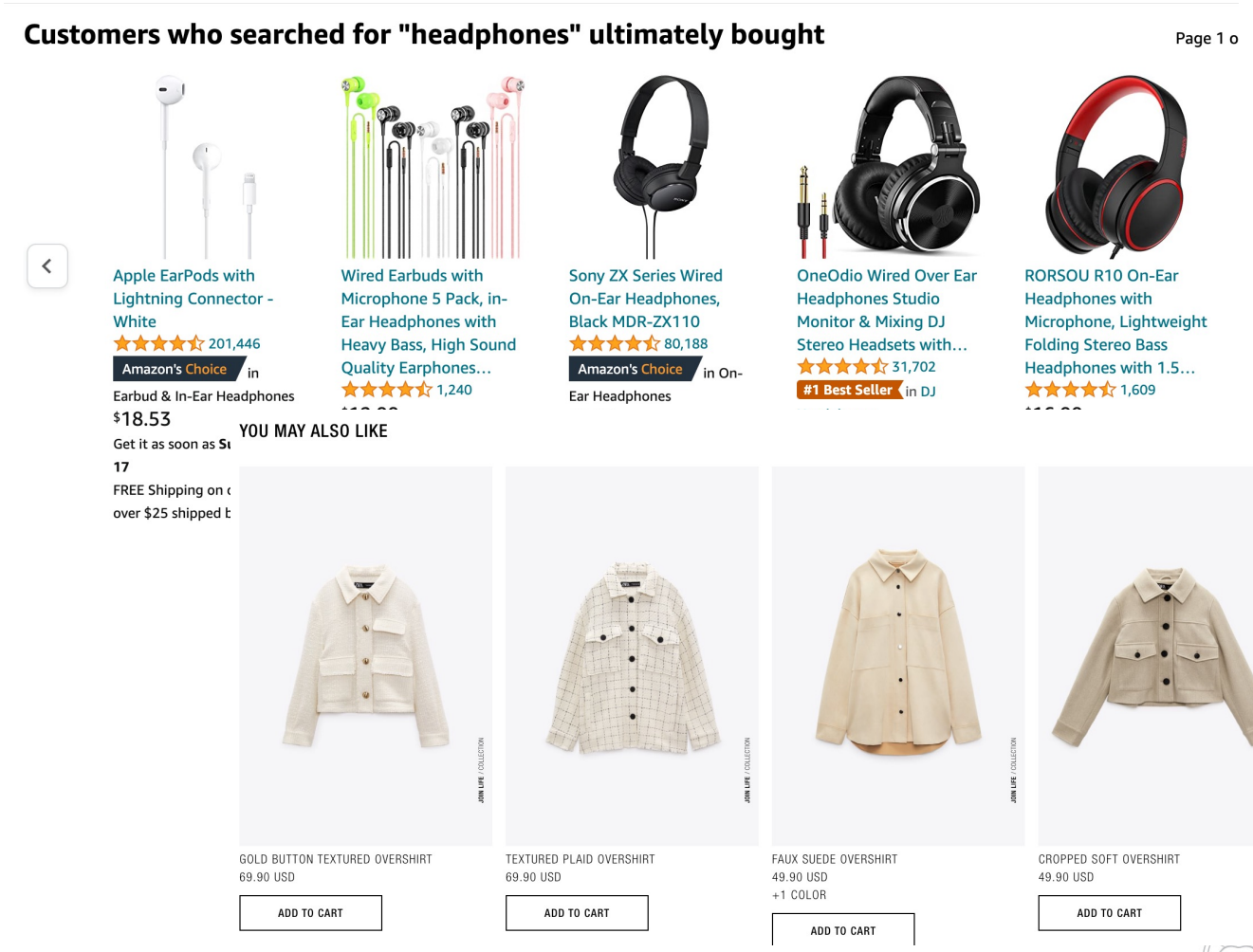
Types of Recommendations

- Editorial and hand curated
  - List of favorites
  - Lists of “essential” items
- **Simple aggregates**
  - Top 10, Most Popular, Recent Uploads
- Tailored to individual users
  - Amazon, Netflix, …



Types of Recommendations

- **Editorial and hand curated**
  - List of favorites
  - Lists of “essential” items
- **Simple aggregates**
  - Top 10, Most Popular, Recent Uploads
- **Tailored to individual users**
  - Amazon, Netflix, …



Formal Model

- ***X*** = set of **Customers**
- ***S*** = set of **Items**

- **Utility function** ***u***: $X \times S \rightarrow R$
  - ***R*** = set of ratings
  - ***R*** is a totally ordered set
  - e.g., **0-5** stars, real number in **[0,1]**

Utility Matrix

	Avatar	The Lord of the Rings	Matrix	Pirates
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

If we want to create a utility matrix in our class

	Avatar	The Lord of the Rings	Matrix	Pirates
Student 1	?			
Student 2				
Student 3				
Student 4				

Question 1: How to collect the data in the utility matrix?

If we want to create a utility matrix in our class

	Avatar	The Lord of the Rings	Matrix	Pirates
Student 1	4.5 →	?		
Student 2				
Student 3				
Student 4				

Question 2: Can we guess users' ratings (unknown) on movies based on existing ratings?

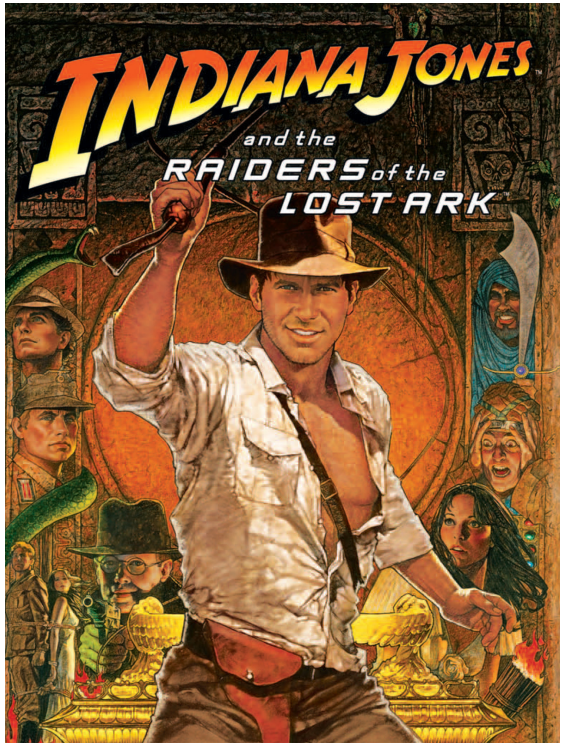
Question 3: What's the performance of our prediction?

Key Problems

- (1) Gathering “known” ratings for matrix
  - How to collect the data in the utility matrix
- (2) Extrapolate unknown ratings from the known ones
  - Mainly interested in high unknown ratings
    - We are not interested in knowing what you don’t like but what you like
- (3) Evaluating extrapolation methods
  - How to measure success/performance of recommendation methods

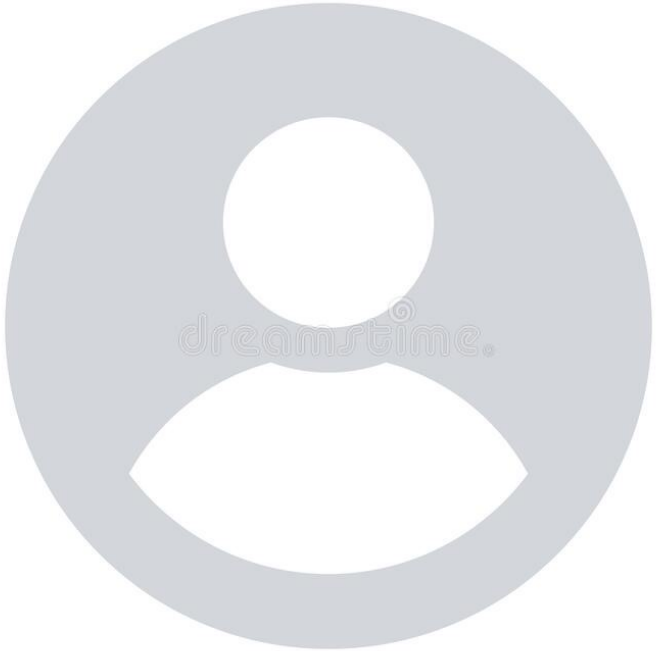
(1) Gathering Ratings

- **Explicit**
  - Ask people to rate items
  - Doesn't work well in practice – customers can't be bothered
- **Implicit**
  - Learn ratings from user actions
    - E.g., purchase implies high rating
  - What about low ratings?



(2) Extrapolating Utilities

- **Key problem:** Utility matrix ***U*** is **sparse**
  - Most people have not rated most items
  - **Cold start:**
    - New items have no ratings
    - New users have no history



(3) How to Measure Performance of Rec Sys?

- Data mining measures (RMSE, F1 score, etc)
- Experimentation (click rate via A/B test, etc)

Recap of Similarity Measures

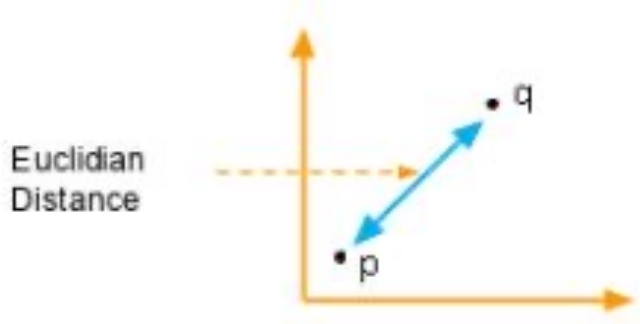
Similarity Measures

- Distance-based (dis)similarity measures
  - Manhattan distance
  - Euclidean distance
  - Minkowski Distance
- Cosine similarity measure
- Pearson correlation-based similarity measure

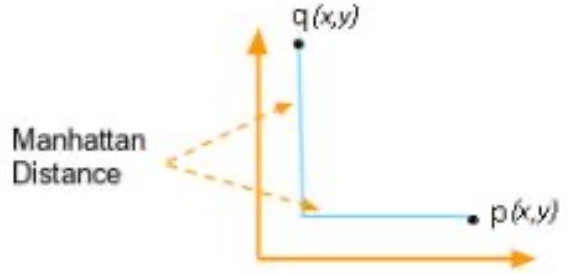
(1) Distance-Based Similarity Measures

- Distance-based (dis)similarity measures between User X and User Y based on n item ratings:
  - Distance >= 0
  - Most similar <-> shortest distance
  - An item rating is considered in the distance measure only if it exists for both users
- We need to use distance-based measures to calculate similarity.

$$\mathbf{d} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$



$$\mathbf{d} = \sum_{i=1}^{n} |q_x - p_x| + |q_x - p_y|$$



(1) When to Use Distance-Based Measures?

- If your data is **dense** (not too many zero or missing attribute values) and the magnitude of the attribute values is important, use distance measures such as Euclidean or Manhattan.
- Because if the data is sparse, then we normally end up with spurious results while performing distance-based measures.

(2) Cosine Similarity Measure

- Cosine similarity measure between Vector X and Vector Y:
  - Cosine similarity lies between -1 and 1
    (-1 total opposites, 0 independent, 1 perfectly similar)
  - Most similar <-> Highest cosine similarity score

$$\cos(x,y) = \frac{x.y}{\|x\|\|y\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

- When to use cosine similarity measure: If the data is **sparse** (too many zero or missing attribute values) consider using Cosine Similarity since it ignores 0 matches.

(3) Pearson Correlation

- Motivation
  - Users often have different rating patterns. For instance, Bill seems to avoid extreme ratings, his ratings range from 2 to 4. Jordyn seems to like everything, her ratings range from 4 to 5. Hailey is a binary person, giving ratings of either 1s or 4s.
  - In other words, users often anchor their ratings at different scales. One user might rate <bad, good, great> as <1, 2, 3>, whereas another user might rate <bad, good, great> as <3, 4, 5>.
  - So we need a way to be able to base similarity on similar trending of ratings, rather than similar absolute ratings.

(3) Pearson Correlation

- Pearson Correlation-based similarity measure between Vector X and Vector Y:
  - Correlation between -1 and 1
    (-1 perfectly negatively correlated, 0 not correlated, 1 perfectly positively correlated)
  - Most Similar <-> Highest Correlation

$$\frac{\sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n}}{\sqrt{\sum_{i=1}^{n} x_i{}^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}} \sqrt{\sum_{i=1}^{n} y_i{}^2 - \frac{\left(\sum_{i=1}^{n} y_i\right)^2}{n}}}$$

- When to use Pearson correlation-based similarity measure: If the data is subject to **grade-inflation** (different users may be using different scales) use Pearson similarity.

Recommendation Systems

Three approaches to recommendation systems:

- Content-based
- Collaborative filtering
- Latent factor based

Content-based Recommendation Systems

Content-based Recommendations

- **Main idea:** Recommend items to customer ***x*** similar to previous items rated highly by ***x***
- Example:
  - **Movie recommendations**
    - Recommend movies with same actor(s), director, genre, …
  - **Websites, blogs, news**
    - Recommend other sites with “similar” content

#GRAMMYs #PremiereCeremony #GRAMMYAwards

64th Annual GRAMMY Awards Premiere Ceremony

8,687,259 views... 207K DISLIKE SHARE SAVE ...

Recording Academy / GRAMMYs
1.86M subscribers
SUBSCRIBE

Music's Biggest Night kicks off with the Premiere Ceremony, featuring 6 special performances

Live chat replay is not available for this video.

GRAMMYs 2022: Must-See Moments!
Entertainment Tonight
1M views • 7 days ago

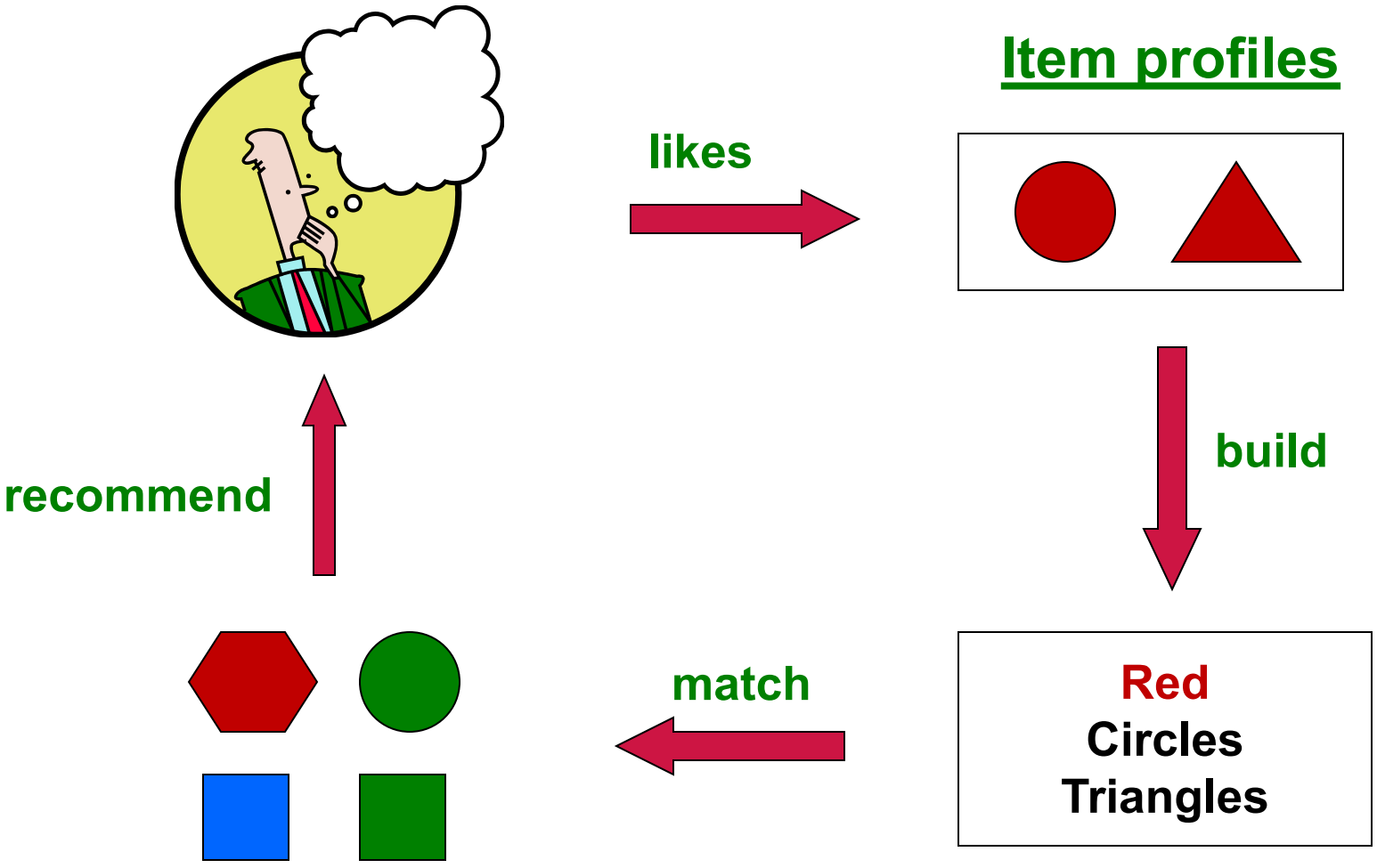
Grammys 2022 Red Carpet FULL Livestream | E! Red Carp...
E! Red Carpet & Award Shows
1.4M views • Streamed 8 days ago

Jennifer Lopez Spotted Wearing Possible Engagement Ring |...
TMZ
6.5K views • 1 day ago
New

SILK SONIC Wins Song Of The Year For "LEAVE THE DOOR...
Recording Academy / GRAMMYs
1.6M views • 8 days ago

Bruno Mars, Morris Day and The Time - Tribute a Prince...
GildaDeMars
8.6M views • 1 year ago

Plan of Action



Item Profiles

- For each item, create an **item profile**
- Profile is a set (vector) of features
  - Movies: author, title, actor, director,…
  - Text: Set of “important” words in document
- How to pick important features?
  - TF-IDF (Term frequency * Inverse Doc Frequency)
    - Term in text analysis → Feature in rec sys
    - Document in text analysis → Item in rec sys

User Profiles and Prediction

- User profile possibilities:
  - Weighted average of rated item profiles
  - Variation**:** weight by difference from average rating for item
  - …
- Prediction heuristic:
  - Given user profile $\boldsymbol{x}$ and item profile $\boldsymbol{i}$, estimate $u(\boldsymbol{x}, \boldsymbol{i}) = \cos(\boldsymbol{x}, \boldsymbol{i}) = \frac{\boldsymbol{x} \cdot \boldsymbol{i}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{i}||}$

Example: Content-base Rec Sys

- Adam has watched ‘Why we Fight and WWII’. Which movie should we recommend to him next?
- Let’s use average of rated item profile as the user profile.

title	duration_minutes	movie	year_added	release_year	TV-PG
Prelude to War	52	1	2017	1942	1
The Battle of Midway	18	1	2018	1942	0
Why We Fight	82	1	2019	1943	0
Undercover	61	0	2020	1943	1
WWII	45	0	2017	1943	0
Tunisian Victory	76	0	2015	1944	1
The Soldier	40	1	2018	1944	0
A Good Day	63	1	2017	1945	0
My Father	59	0	2016	1945	0
One Day in Rome	32	1	2014	1945	0

Original table

title	duration_minutes	movie	year_added	release_year	TV-PG
Prelude to War	52	1	2017	1942	1
The Battle of Midway	18	1	2018	1942	0
Why We Fight	82	1	2019	1943	0
Undercover	61	0	2020	1943	1
WWII	45	0	2017	1943	0
Tunisian Victory	76	0	2015	1944	1
The Soldier	40	1	2018	1944	0
A Good Day	63	1	2017	1945	0
My Father	59	0	2016	1945	0
One Day in Rome	32	1	2014	1945	0

Videos Adam has watched

title	duration_minutes	movie	year_added	release_year	TV-PG
Why We Fight	82	1	2019	1943	0
WWII	45	0	2017	1943	0

Get the average values for each column, then create Adam's profile

User	duration_minutes	movie	year_added	release_year	TV-PG
Adam	63.5	0.5	2018	1943	0

Original table

title	duration_minutes	movie	year_added	release_year	TV-PG
Prelude to War	52	1	2017	1942	1
The Battle of Midway	18	1	2018	1942	0
Why We Fight	82	1	2019	1943	0
Undercover	61	0	2020	1943	1
WWII	45	0	2017	1943	0
Tunisian Victory	76	0	2015	1944	1
The Soldier	40	1	2018	1944	0
A Good Day	63	1	2017	1945	0
My Father	59	0	2016	1945	0
One Day in Rome	32	1	2014	1945	0

Get the average values for each column, then create Adam's profile

User	duration_minutes	movie	year_added	release_year	TV-PG
Adam	63.5	0.5	2018	1943	0

According to the similarity, we should recommend 'A Good Day'.

Calculate the Manhattan distance between user profile and the rest videos.

- Adam and 'Prelude to War' =|63.5 - 52|+|0.5-1|+|2018-2017|+|1943-1942|+|0-1|=15
- Adam and 'The Battle of Midway' =|63.5 - 18|+|0.5-1|+|2018-2018|+|1943-1942|+|0-0|=47

Following the above calculations, we get

- Adam – 'Undercover' = 6
- Adam – 'Tunisian Victory' = 18
- Adam – 'The Solider' = 25
- Adam – 'A Good Day' = 4
- Adam – 'My Father' = 9
- Adam – 'ODIR' = 38

Pros: Content-based Approach

- **+: No need for data on other users**
- **+: Able to recommend to users with unique tastes**
- **+: Able to recommend new & unpopular items**
  - No first-rater problem
- **+: Able to provide explanations**
  - Can provide explanations of recommended items by listing content-features that caused an item to be recommended

Cons: Content-based Approach

- **–: Finding the appropriate features is hard**
  - E.g., images, movies, music
- **–: Recommendations for new users**
  - **Cold start issue: 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**

User	duration_minutes	movie	year_added	release_year	TV-PG
Adam	63.5	0.5	2018	1943	0

Collaborative Filtering

Harnessing quality judgments of other users

Collaborative Filtering Based Recommender Systems

- Collaborative filtering attempts to predict what other *Items* a **user might like based on existing user and item data in the system.**
- Examples of collaborative filtering:
  - Make product recommendations for a user based on what other “similar” users have liked
  - Make song recommendations for a user based on what other “similar” users have played.

People Also Viewed

U AIRism Cotton Oversized Crew Neck T-Shirt

U Crew Neck Short-Sleeve T-Shirt

$ 19.90

Dry Crew Neck Short-Sleeve Color T-Shirt

Supima® Cotton Crew Neck Short-Sleeve T-Shirt

Frequently Bought Together

Anime Jujutsu Kaisen 0 UT (Short-Sleeve Graphic T-Shirt)

Sweat Pullover Long-Sleeve Hoodie

Anime Jujutsu Kaisen 0 UT (Short-Sleeve Graphic T-Shirt)

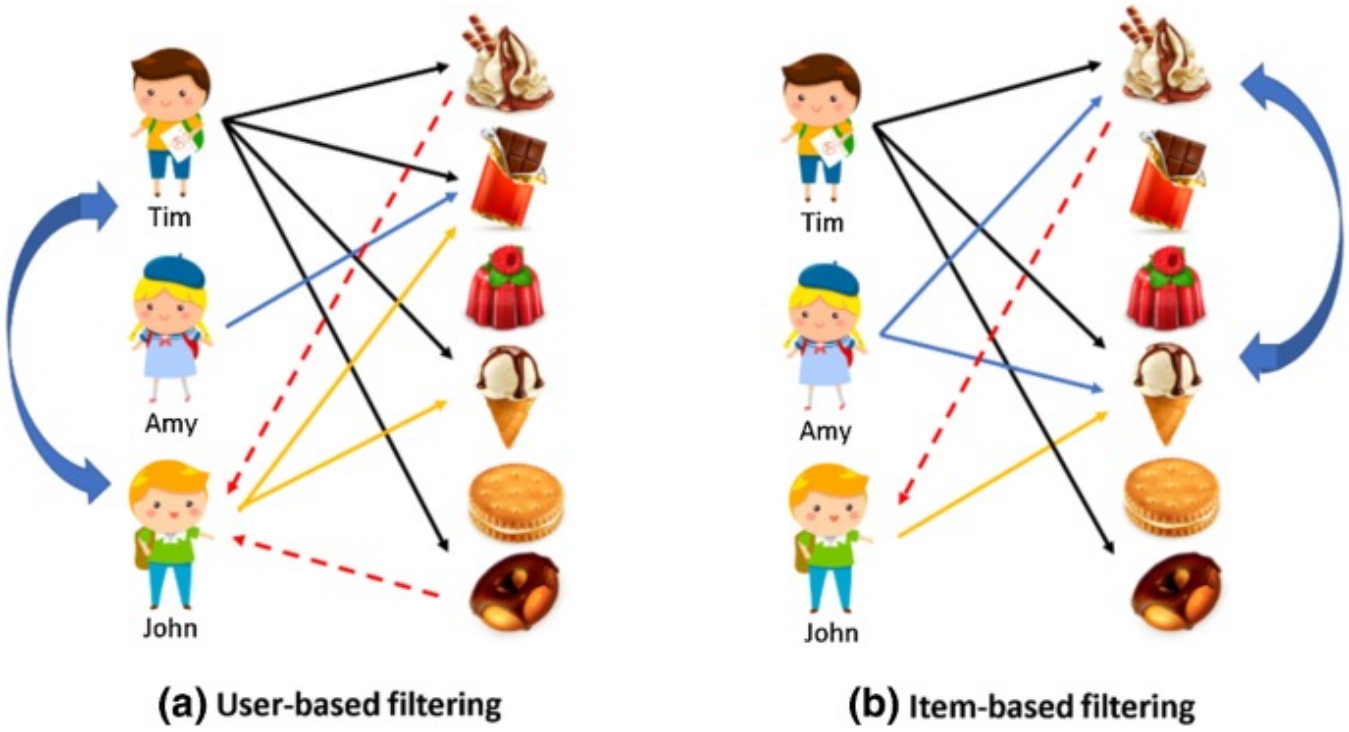
Open Collar Short-Sleeve Shirt

New

Collaborative Filtering

The two most prevalent methods within Collaborative Filtering are

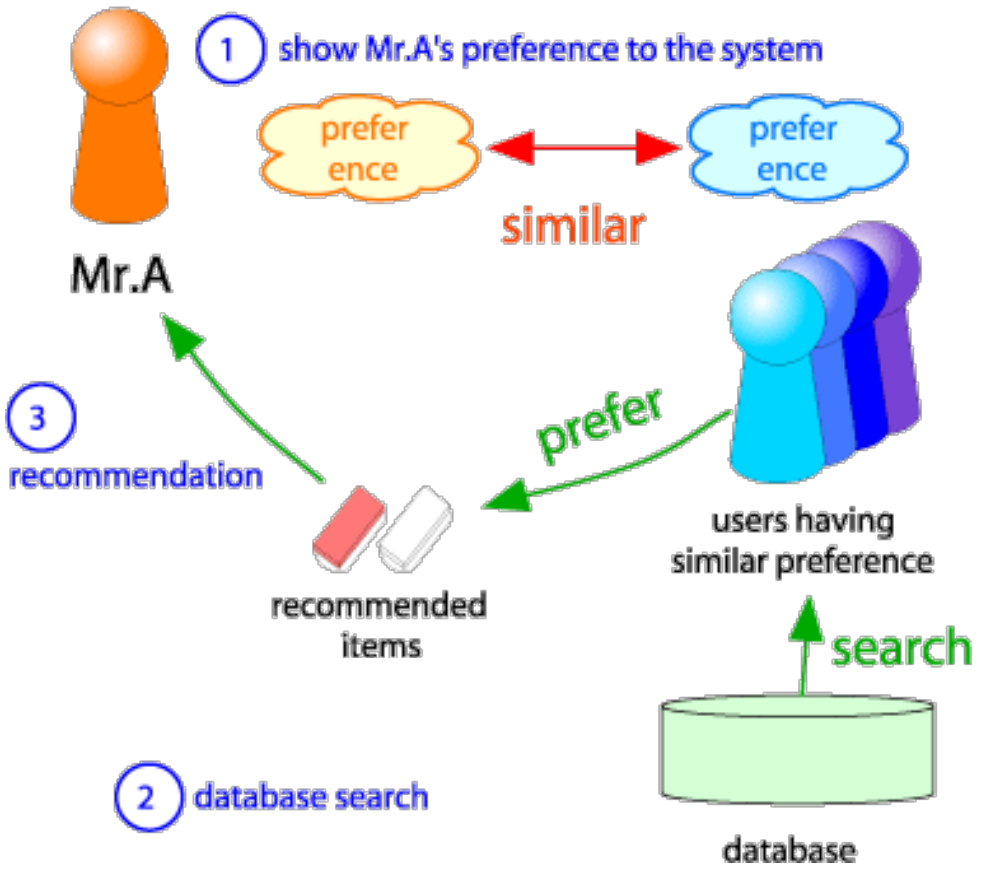
- User-Based Filtering
- Item-Based Filtering



User-based Collaborative Filtering

Collaborative Filtering

- Consider user *x*
- Find set *N* of other users whose ratings are “similar” to *x*’s ratings
- Estimate *x*’s ratings based on ratings of users in *N*



User-based Collaborative Filtering

- We compare a user with every other user to find the closest matches
  - Also called **Memory-Based Filtering** because we need to store all ratings in order to make recommendations
- It's a 3-step process. Let's say we are trying to find item recommendations for User X:
  - Step 1: Find past item ratings from User X
  - Step 2: Find the "most similar" User Y (based on similarity of item ratings) from the remaining user corpus
  - Step 3: Recommend those items to User X that the "most similar" User Y has rated, and that User X hasn't used yet

Running Example for this Section

- Let's say we are trying to find item recommendations for Veronica:
  - We already have Veronica's past item ratings.
  - Now if we can find the user who is "most similar" to Veronica based on their item ratings,
  - then we can recommend those items to Veronica that are highly rated by that "most similar", and that Veronica hasn't already discovered

Ratings	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

Example - Manhattan Distance

- We want to find item recommendations for Veronica.
  - Manhattan distances between Veronica and all other users (only consider those items ratings in distance measure that have been rated by both users)
  - Manhattan distance Equation: $|x_{1i} - x_{1j}| + |x_{2i} - x_{2j}| + \cdots + |x_{ki} - x_{kj}|$

Ratings	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

Example - Manhattan Distance

- We want to find item recommendations for Veronica.
  - Manhattan distances between Veronica and all other users (only consider those items ratings in distance measure that have been rated by both users)

Ratings	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

- Angelica and Veronica=|3.5-3|+|4.5-5|+|5-4|+|1.5-2.5|+|2.5-3|=3.5
- Bill and Veronica = |2-3| + |2-4| + |3.5-2.5| = 4
- Chan and Veronica = |5-3| + |3-5| + |5-4| + |1-2.5| = 6.5
- Dan and Veronica = |3-3| + |3-4| + |4.5-2.5| + |4-3| = 4
- Hailey and Veronica=|4-5|+|4-3|=2
- Jordyn and Veronica = |5-5| + |5-4| + |4.5-2.5| + |4-3| = 4
- Sam and Veronica = |5-3| + |3-5| + |5-4| + |4-2.5| + |5-3| = 8.5

Ratings	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

Manhattan Distance	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Angelica	-	9	4.5	9	5	9.5	8	3.5
Bill	9	-	14	5	5.5	6	8	4
Chan	4.5	14	-	14	4	12	4	6.5
Dan	9	5	14	-	4.5	5	7.5	4
Hailey	5	5.5	4	4.5	-	7.5	4	2
Jordyn	9.5	6	12	5	7.5	-	6	4
Sam	8	8	4	7.5	4	6	-	8.5
Veronica	3.5	4	6.5	4	2	4	8.5	-

Exercise - Manhattan Distance

- User most similar (shortest distance) to Veronica: Hailey (Manhattan Distance 2)
  - Hailey has rated three items that Veronica hasn't: Broken Bells (Rating 4), Deadmau5 (Rating 1), Vampire Weekend (Rating 1)
- So, we can make the following recommendation to Veronica
  - [('Broken Bells',4.0), ('Deadmau5',1.0), ('Vampire Weekend',1.0)]
- Since these are highly rated items by the most similar user Hailey, and these are items that Veronica hasn't discovered yet

Item-based Collaborative Filtering

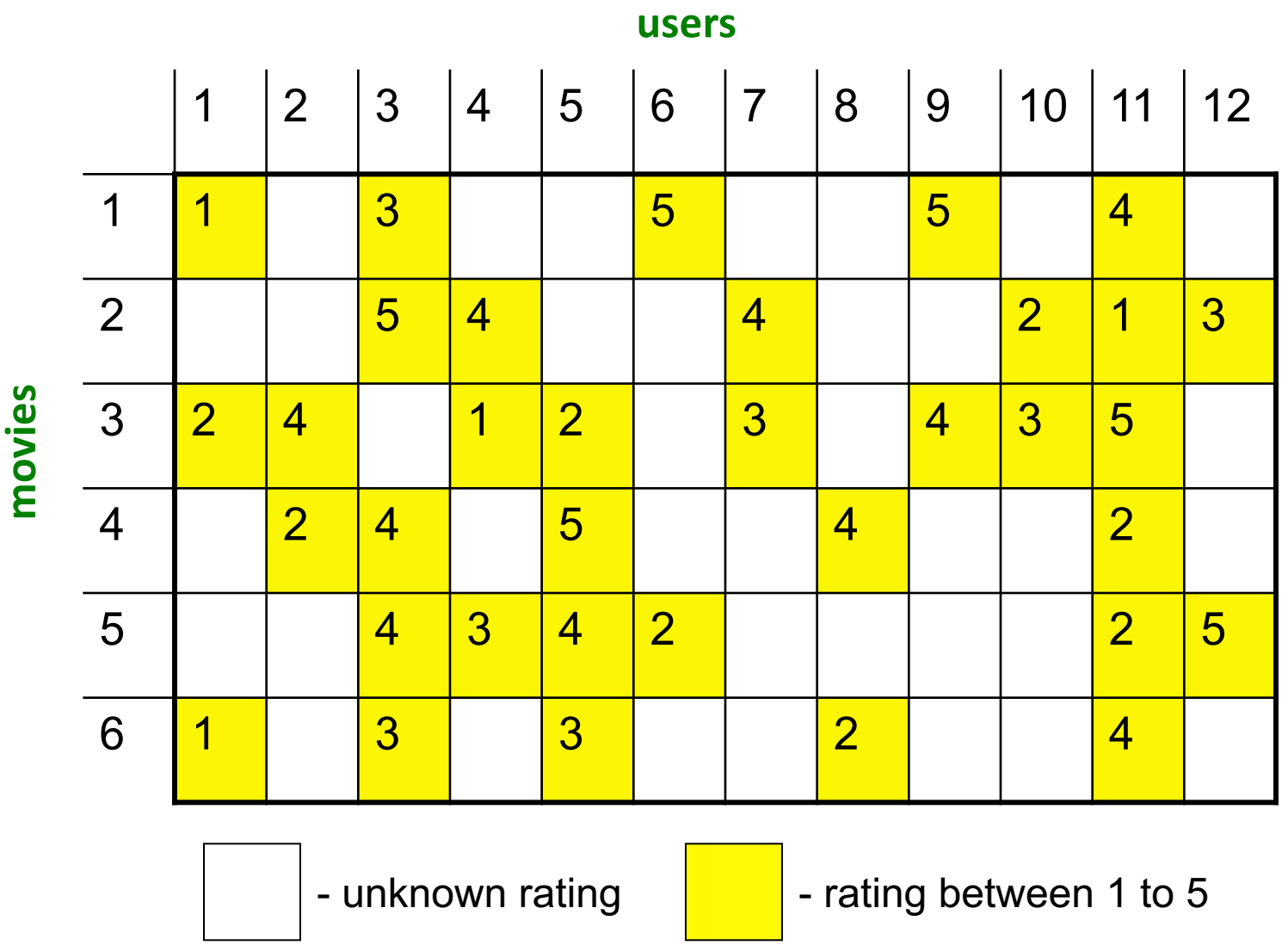
Item-Item Collaborative Filtering

- So far: we have learned user-based collaborative filtering
- Another view: Item-based collaborative filtering
- For item *i*, find other similar items
  - Estimate rating for item *i* based on ratings for similar items
  - Can use same similarity metrics and prediction functions as in user-user model

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

s_{ij}... similarity of items ***i*** and ***j***
r_{xj}...rating of user ***u*** on item ***j***
N(i;x)... set items rated by ***x*** similar to ***i***

Item-Item CF (|N|=2)



Item-Item CF (|N|=2)

users

movies	1	2	3	4	**5**	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- estimate rating of movie **1** by user **5**

Item-Item CF (|N|=2)

users

movies

	1	2	3	4	**5**	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		**0.41**
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		**0.59**

Neighbor selection:
Identify movies similar to movie **1**, **rated by user 5**

Here we use Pearson correlation as similarity:

1) Subtract mean rating m_i from each movie ***i***

$m_1 = (1+3+5+5+4)/5 = 3.6$

row 1: *[-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0.4, 0]*

2) Compute cosine similarities between rows

Item-Item CF (|N|=2)

users

movies

	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		0.41
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		0.59

Compute similarity weights:

$s_{1,3}=0.41$, $s_{1,6}=0.59$

Item-Item CF (|N|=2)

users

movies

	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		**2.6**	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Predict by taking weighted average:

$r_{1.5}$ = (0.41*2 + 0.59*3) / (0.41+0.59) = 2.6

$$r_{ix} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{jx}}{\sum s_{ij}}$$

Ratings	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

K=3

K-Nearest Similarity	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Angelica	-	-0.9	0.82	-0.36	0.42	0.76	0.28	0.83
Bill	-0.9	-	-0.98	0.66	0	-0.71	-0.82	-0.76
Chan	0.82	-0.98	-	-0.96	0.5	0.8	0.77	0.27
Dan	-0.36	0.66	-0.96	-	0.39	-0.04	-0.49	-0.75
Hailey	0.42	0	0.5	0.39	-	0.61	0	0
Jordyn	0.76	-0.71	0.8	-0.04	0.61	-	-0.18	0.75
Sam	0.28	-0.82	0.77	-0.49	0	-0.18	-	-0.56
Veronica	0.83	-0.76	0.27	-0.75	0	0.75	-0.56	-

	Recommendation
Angelica	[('Deadmau5', 1.64)]
Bill	[('The Strokes', 4.0), ('Norah Jones', 1.85)]
Chan	[('The Strokes', 3.82), ('Vampire Weekend', 2.01)]
Dan	[('Norah Jones', 2.58)]
Hailey	[('Phoenix', 5.01), ('Blues Traveler', 2.75), ('Slightly Stoopid', 2.40)]
Jordyn	[('Blues Traveler', 3.84)]
Sam	[('Vampire Weekend', 0.88), ('Deadmau5', 0.68)]
Veronica	[('Broken Bells', 2.64), ('Vampire Weekend', 2.20), ('Deadmau5', 1.71)]

CF: Common Practice

- Define **similarity** s_{ij} of items ***i*** and ***j***
- Select ***k*** nearest neighbors $N(i; x)$
  - Items most similar to ***i***, that were rated by ***x***
- Estimate rating r_{xi} as the weighted average:

Before:

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

$$r_{xi} = b_{xi} + \frac{\sum_{j \in N(i;x)} s_{ij} \cdot (r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} s_{ij}}$$

baseline estimate for r_{xi}

$$b_{xi} = \mu + b_x + b_i$$

- μ = overall mean movie rating
- b_x = rating deviation of user ***x***
  = (*avg. rating of user* ***x***) – μ
- b_i = rating deviation of movie ***i***

Item-based vs. User-based CF

- This can be very confusing, but did you see the differences between User-based and Item-based CF?

Ratings	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

User-based CF

- User-based CF: Generate the similarity between USERS (in the utility matrix below, we get the similarity between different USERS/COLUMNS)

Ratings	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

Item-based CF

- Item-based CF: Generate the similarity between ITEMS (in the utility matrix below, we get the similarity between different ITEMS/ROWS)

Ratings	Angelica	Bill	Chan	Dan	Hailey	Jordyn	Sam	Veronica
Blues Traveler	3.5	2	5	3	-	-	5	3
Broken Bells	2	3.5	1	4	4	4.5	2	-
Deadmau5	-	4	1	4.5	1	4	-	-
Norah Jones	4.5	-	3	-	4	5	3	5
Phoenix	5	2	5	3	-	5	5	4
Slightly Stoopid	1.5	3.5	1	4.5	-	4.5	4	2.5
The Strokes	2.5	-	-	4	4	4	5	3
Vampire Weekend	2	3	-	2	1	4	-	-

Item-based vs. User-based Collaborative Filtering

	Avatar	LOTR	Matrix	Pirates
Alice	1		0.8	
Bob		0.5		0.3
Carol	0.9		1	0.8
David			1	0.4

- In practice, it has been observed that item-item often works better than user-user
- **Why?** Items are simpler, users have multiple tastes

Pros of Collaborative Filtering

- **+ Works for any kind of item**
  - No feature selection needed

Cons of Collaborative Filtering

- **- Cold Start:**
  - Need enough users in the system to find a match
- **- Sparsity:**
  - The user/ratings matrix is sparse
  - Hard to find users that have rated the same items
- **- First rater:**
  - Cannot recommend an item that has not been previously rated
  - New items, Esoteric items
- **- Popularity bias:**
  - Cannot recommend items to someone with unique taste
  - Tends to recommend popular items

Hybrid Methods

- **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

Take 10 minutes break…