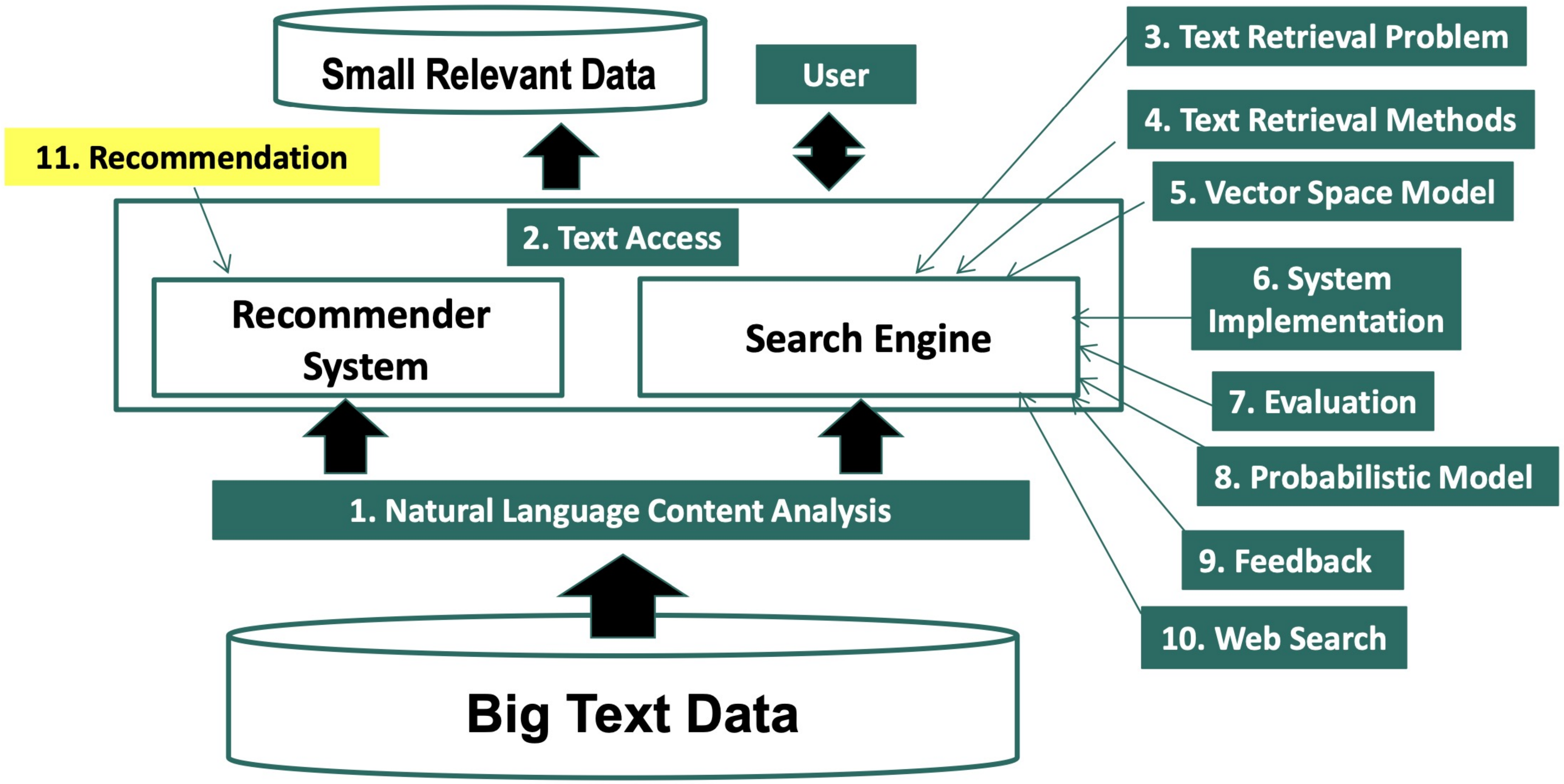
Information Retrieval

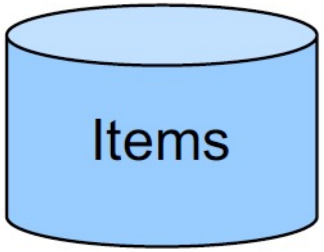
Recommender Systems

Dr. Iqra Safder

Recommender Systems



Recommendations



Examples:

amazon.com

PANDORA

StumbleUpon

del.icio.us

NETFLIX

movielens
helping you find the *right* movies

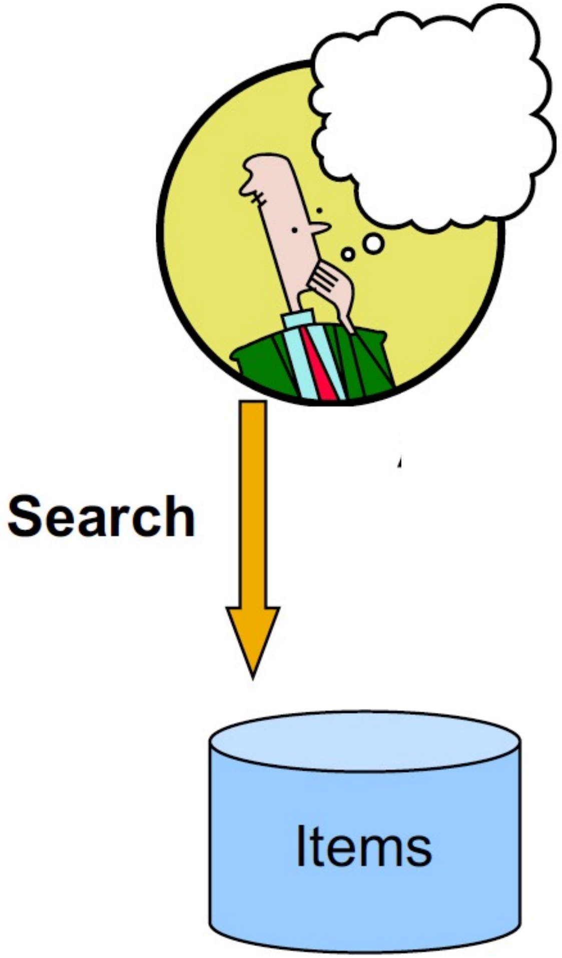
last.fm
the social music revolution

Google News

YouTube

XBOX LIVE

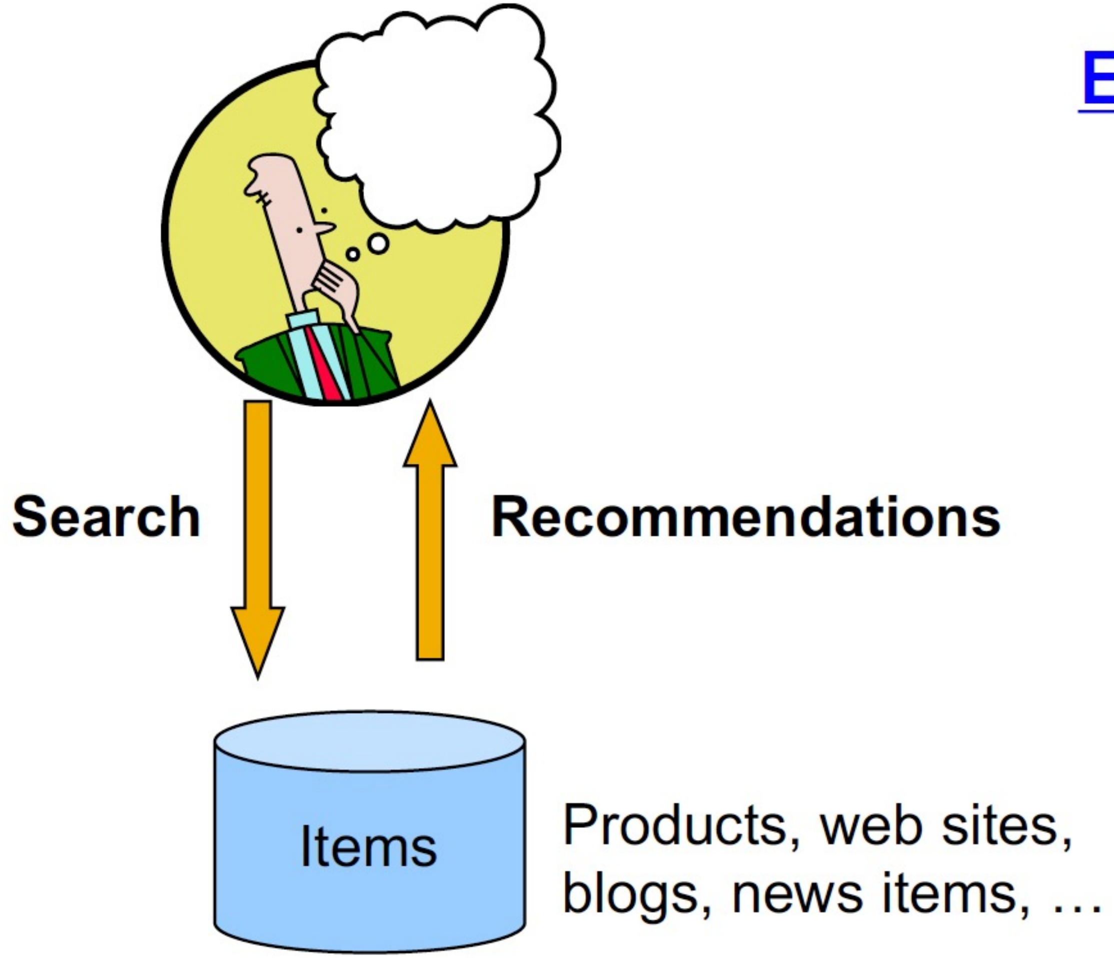
Recommendations



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

Recommendations



Examples:

amazon.com
PANDORA
StumbleUpon
NETFLIX
del.icio.us
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

Types of Recommendations

- 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)
- Examples
  - Books, movies, music, news articles
  - People (friend recommendations on Facebook, LinkedIn, and Twitter)

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	LOTR	Matrix	Pirates
Alice	1		0.2	
Bob		0.5		0.3
Carol	0.2		1	
David				0.4

Key Problems

- (1) Gathering “known” ratings for matrix
  - How to collect the data in the utility matrix

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

1) Gathering Ratings

- **Explicit**
  - Ask people to rate items
  - Doesn’t work well in practice – people can’t be bothered
  - Crowdsourcing: Pay people to label items

1) Gathering Ratings

- **Explicit**
  - Ask people to rate items
  - Doesn’t work well in practice – people can’t be bothered
  - Crowdsourcing: Pay people to label items
- **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
- **Approaches to recommender systems**
  - **1)** Content-based
  - **2)** Collaborative

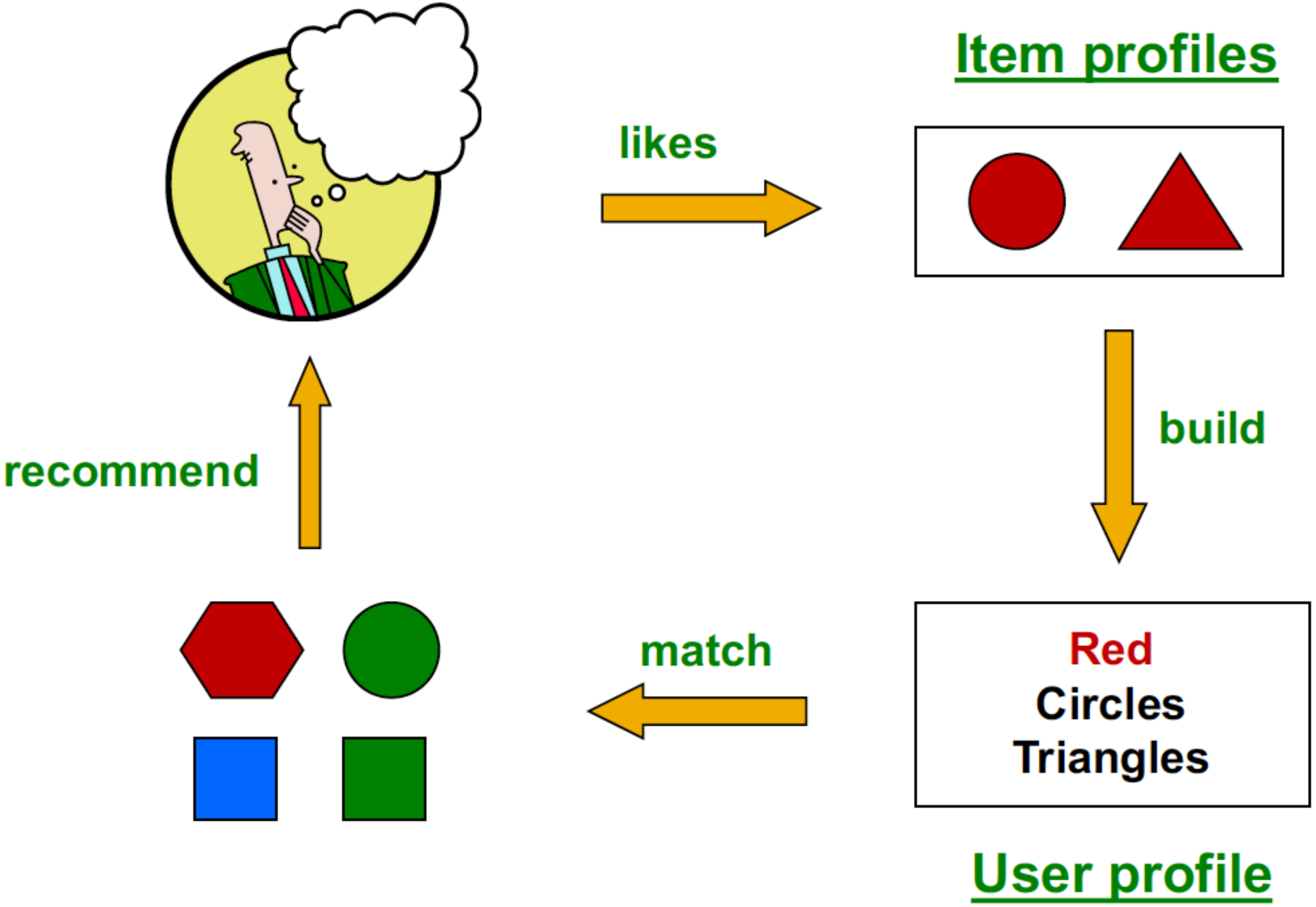
Content-based Recommendations

- **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

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?**
  - Usual heuristic from text mining is **TF-IDF** (Term frequency * Inverse Doc Frequency)
    - **Term … Feature**
    - **Document … Item**

User Profile and Prediction

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

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- **Prediction heuristic:**
  - Given user profile ***x*** and item profile ***i***, estimate

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

Pros: Content-based Approach

- **+: No need for data on other users**
  - No cold-start or sparsity problems
- **+: 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
  - **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-User Collaborative Filtering
Item-Item Collaborative Filtering

Similar Users

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

- Consider users $\boldsymbol{x}$ and $\boldsymbol{y}$ with rating vectors $\boldsymbol{r}_x$ and $\boldsymbol{r}_y$
- We need a similarity metric sim($\boldsymbol{x}$, $\boldsymbol{y}$)
- Capture intuition that sim(A,B) > sim(A,C)

Option 1: Jaccard Similarity

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

- $sim(A,B) = |r_A \cap r_B| / |r_A \cup r_B|$
- $sim(A,B) = 1/5$; $sim(A,C) = 2/4$
  - $sim(A,B) < sim(A,C)$

Option 2: Cosine Similarity

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

- sim(A,B) = cos(r_A, r_B)
- sim(A,B) = 0.38, sim(A,C) = 0.32
  - sim(A,B) > sim(A,C), but not by much

Option 2: Cosine Similarity

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4	0	0	5	1	0	0
B	5	5	4	0	0	0	0
C				2	4	5	
D		3					3

- Problem with cosine:
  - C really loves SW
  - A hates SW
  - B just hasn't seen it
- Another problem: we'd like to normalize the raters
  - D rated everything the same; not very useful

- sim(A,B) = cos(r_A, r_B)
- sim(A,B) = 0.38, sim(A,C) = 0.32
  - sim(A,B) > sim(A,C), but not by much
- Problem: treats missing ratings as negative

Option 3: Centered Cosine

- Normalize ratings by subtracting row mean

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Option 3: Centered Cosine

- Normalize ratings by subtracting row mean

	HP1	HP2	HP3	TW	SW1	SW2	SW3	
A	4			5	1			10/3
B	5	5	4					14/3
C				2	4	5		
D		3					3	

Option 3: Centered Cosine

- Normalize ratings by subtracting row mean

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	$2/3$			$5/3$	$-7/3$		
B	$1/3$	$1/3$	$-2/3$				
C				$-5/3$	$1/3$	$4/3$	
D		0					0

Option 3: Centered Cosine Similarity

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	$2/3$			$5/3$	$-7/3$		
B	$1/3$	$1/3$	$-2/3$				
C				$-5/3$	$1/3$	$4/3$	
D		0					0

- sim(A,B) = cos(r_A, r_B) = 0.09; sim(A,C) = -0.56
  - sim(A,B) > sim(A,C)
- Captures intuition better
  - Missing ratings treated as "average"
  - Handles "tough raters" and "easy raters"

Item-Item Collaborative Filtering

- So far: User-user collaborative filtering
- Another view: Item-item
  - 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

Item-Item Collaborative Filtering

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	

☐ - unknown rating

■ - rating between 1 to 5

Item-Item Collaborative Filtering

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**

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	..
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		

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

Here we use mean centered item-overlap cosine as similarity:
1) Subtract mean rating m_i from each movie i between rows
2) Compute (item-overlapping) cosine similarities

users

movies

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

Subtract mean rating m_i from each movie i

$$m_1 = (1+3+5+5+4)/5 = 18/5$$

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	-13/5		-3/5		?	7/5			7/5		2/5	
3	-1	1		-2	-1		0		1	0	2	
6	-8/5		2/5		2/5			-3/5			7/5	

Showing computation only for #3 and #6

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

Here we use mean centered item-overlap cosine as similarity:
1) Subtract mean rating m_i from each movie i
2) Compute (item-overlapping) cosine similarities between rows

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	-13/5		-3/5		?	7/5			7/5		2/5		1.00
2			5	4			4			2	1	3	..
3	-1	1		-2	-1		0		1	0	2		?
4		2	4		5			4			2		..
5			4	3	4	2					2	5	..
6	-8/5		2/5		2/5			-3/5			7/5		?

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

Here we use mean centered item-overlap cosine as similarity:
1) Subtract mean rating m_i from each movie i
2) Compute (item-overlapping) cosine similarities between rows

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	-13/5		-3/5		?	7/5			7/5		2/5		1.00
2			5	4			4			2	1	3	..
3	-1	1		-2	-1		0		1	0	2		?
4		2	4		5			4			2		..
5			4	3	4	2					2	5	..
6	-8/5		2/5		2/5			-3/5			7/5		?

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

Here we use mean centered item-overlap cosine as similarity:

1) Subtract mean rating m_i from each movie i

2) Compute (item-overlapping) cosine similarities between rows

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	-13/5		-3/5		?	7/5			7/5		2/5		1.00
2			5	4			4			2	1	3	..
3	-1	1		-2	-1		0		1	0	2		?
4		2	4		5			4			2		..
5			4	3	4	2					2	5	..
6	-8/5		2/5		2/5			-3/5			7/5		?

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

Here we use mean centered item-overlap cosine as similarity:

1) Subtract mean rating m_i from each movie i

2) Compute (item-overlapping) cosine similarities between rows

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	-13/5		-3/5		?	7/5			7/5		2/5		1.00
2			5	4			4			2	1	3	..
3	-1	1		-2	-1		0		1	0	2		?
4		2	4		5			4			2		..
5			4	3	4	2					2	5	..
6	-8/5		2/5		2/5			-3/5			7/5		?

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

Here we use mean centered item-overlap cosine as similarity:
1) Subtract mean rating m_i from each movie i
2) Compute (item-overlapping) cosine similarities between rows

Compute Cosine Similarity:

For rows 1 and 3, they both have values for users 1, 9 and 11.

$$\text{sim}(1, 3) = \frac{(-13/5)(-1)+(7/5)(1)+(2/5)(2)}{\sqrt{(-13/5)^2+(7/5)^2+(2/5)^2} \cdot \sqrt{(-1)^2+(1)^2+(2)^2}} \approx 0.658$$

For rows 1 and 6, they both have values for users 1, 3 and 11.

$$\text{sim}(1, 6) = \frac{(-13/5)(-8/5)+(-3/5)(2/5)+(2/5)(7/5)}{\sqrt{(-13/5)^2+(-3/5)^2+(2/5)^2} \cdot \sqrt{(-8/5)^2+(2/5)^2+(7/5)^2}} \approx 0.768$$

Stanford

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.000
2			5	4			4			2	1	3	..
<u>3</u>	2	4		1	2		3		4	3	5		<u>.658</u>
4		2	4		5			4			2		..
5			4	3	4	2					2	5	..
<u>6</u>	1		3		3			2			4		<u>.768</u>

Compute similarity weights:

$s_{1,3}$=.658, $s_{1,6}$=.768 (we compute $s_{1,2}$, $s_{1,4}$, $s_{1,5}$ too; let's assume those are smaller)

Slides adapted from Jure Leskovec, CS246 and J. Leskovec, A. Rajaraman, J. Ullman: *Mining of Massive Datasets*

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		2.54	5			5		4		1.000
2			5	4			4			2	1	3	..
3	2	4		1	2		3		4	3	5		.658
4		2	4		5			4			2		..
5			4	3	4	2					2	5	..
6	1		3		3			2			4		.768

Predict by taking weighted average:

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

$r_{1,5}$ = (0.658*2 + 0.768*3) / (0.658+0.768) = 2.54

Item-Item vs. User-User

- In theory, user-user and item-item are dual approaches
- In practice, item-item outperforms user-user in many use cases
- Items are “simpler” than users
  - Items belong to a small set of “genres”, users have varied tastes
  - Item Similarity is more meaningful than User Similarity

Pros/Cons of Collaborative Filtering

+ Works for any kind of item

- No feature selection needed

- 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

- Popularity bias:

- Cannot recommend items to someone with unique taste
- Tends to recommend popular items

- Ethical and social issues:

- Can lead to filter bubbles and radicalization spirals

Stanford

Thank you