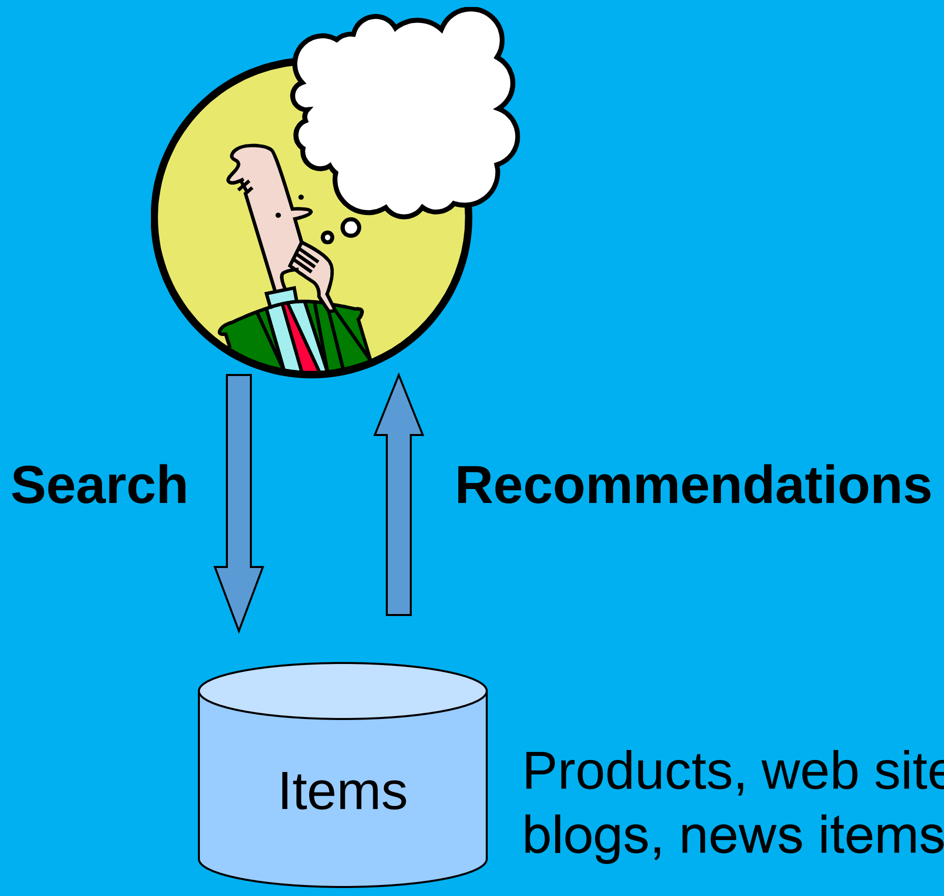
Khasha Dehnad

Recommendation Systems

Application of Recommendation Systems

- Product Recommendations
- Movie Recommendations
- News Articles

Recommendations



Products, web sites, blogs, news items, …

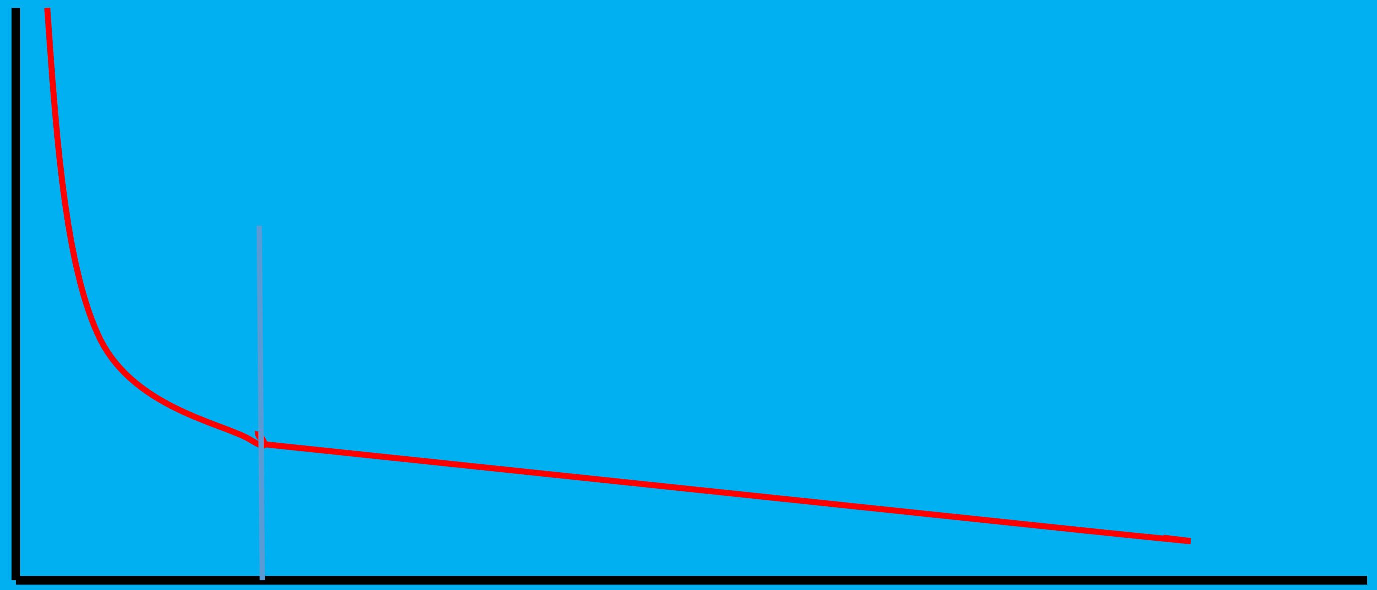
Examples:

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

The Long Tail Curve



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, …

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
- **(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 – people 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
- **Two approaches to recommender systems:**
  - **1)** Content-based
  - **2)** Collaborative

Recommendation Systems

- Content-based

  Focus on properties of items

- Collaborative

  Focus on the relationship between users and item

Content-based Recommender 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

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

Recommendation Systems

- Content-based: Movie Recommendation

  Focus on properties of items

Movie	Actor1	Actor2	Actor3	Actor4	Actor5	Actor6	Rating
A	1			1	1		5
B	1	1	1				4
C				1	1	1	2
D		1					4

Pros: Content-based Approach

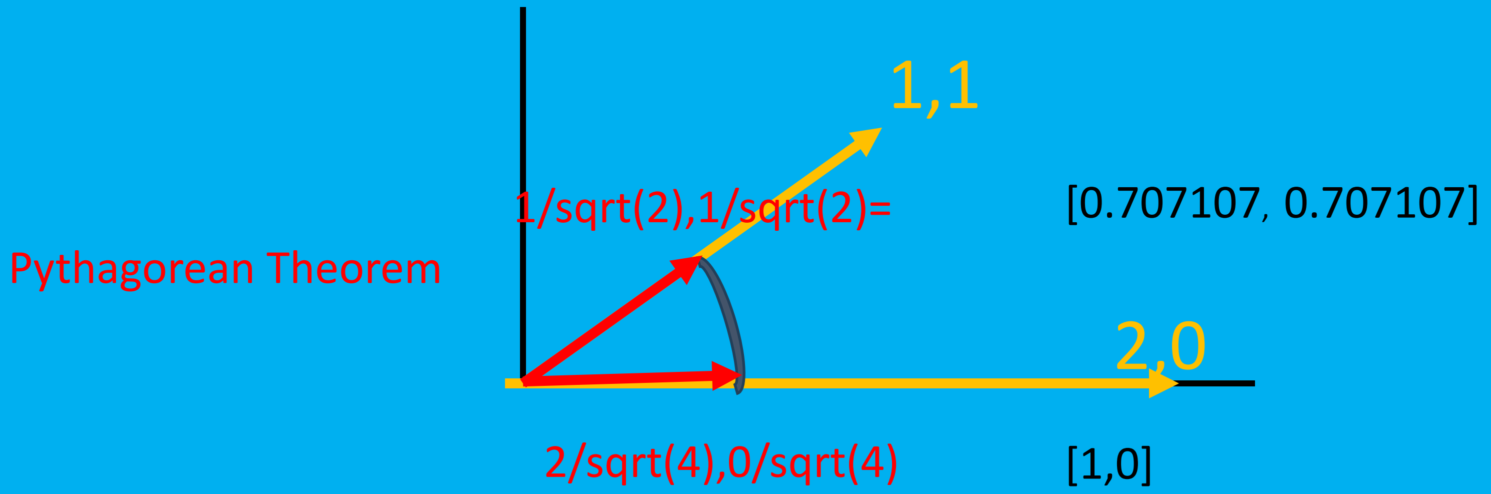
- **+: No need for data on other users**
  - No cold-start or sparsity problems
- **+: Able to recommend to users with unique tastes**
- **+: Able to recommend new & unpopular items**
  - No first-rater problem
- **+: Able to provide explanations**
  - Can provide explanations of recommended items by 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**

Finding "Similar" Users

- **Cosine similarity measure**
  - sim($\boldsymbol{x}$, $\boldsymbol{y}$) = cos($\boldsymbol{r}_x$, $\boldsymbol{r}_y$) = $\frac{r_x \cdot r_y}{||r_x|| \cdot ||r_y||}$
  - **Problem:** Treats missing ratings as "1egative"



Apnedix

Utility Matrix: Relationship between users and items

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Cosine sim:

$$sim(x,y) = \frac{\sum_i r_{xi} \cdot r_{yi}}{\sqrt{\sum_i r_{xi}^2} \cdot \sqrt{\sum_i r_{yi}^2}}$$

Similarity Metric

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

- **Intuitively we want: sim(*A*, *B*) > sim(*A*, *C*)**
- **Jaccard similarity:** 1/5 < 2/4
- **Cosine similarity:** 0.386 > 0.322
  - Considers missing ratings as "negative"
  - **Solution: subtract the (row) mean**

	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 vs. A,C:
0.092 > -0.559

Notice cosine sim. is correlation when data is centered at 0

Recommendation Systems

- Content-based: Movie Recommendation

  Focus on properties of items

Movie	Actor1	Actor2	Actor3	Actor4	Actor5	Actor6	Rating
A	1			1	1		5
B	1	1	1				4
C				1	1	1	2
D		1					4

Collaborative
Focus on the relationship between users and item

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7
A	1			1			
B	1	1	1				
C					1	1	
D		1					1

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7
A	0.666667			1.666667	-2.33333		
B	0.333333	0.333333	-0.66667				
C				-1.66667	0.333333	1.333333	
D		0					0