Recommender Systems

Overview
Content-based systems
Collaborative Filtering
Evaluating recommender systems

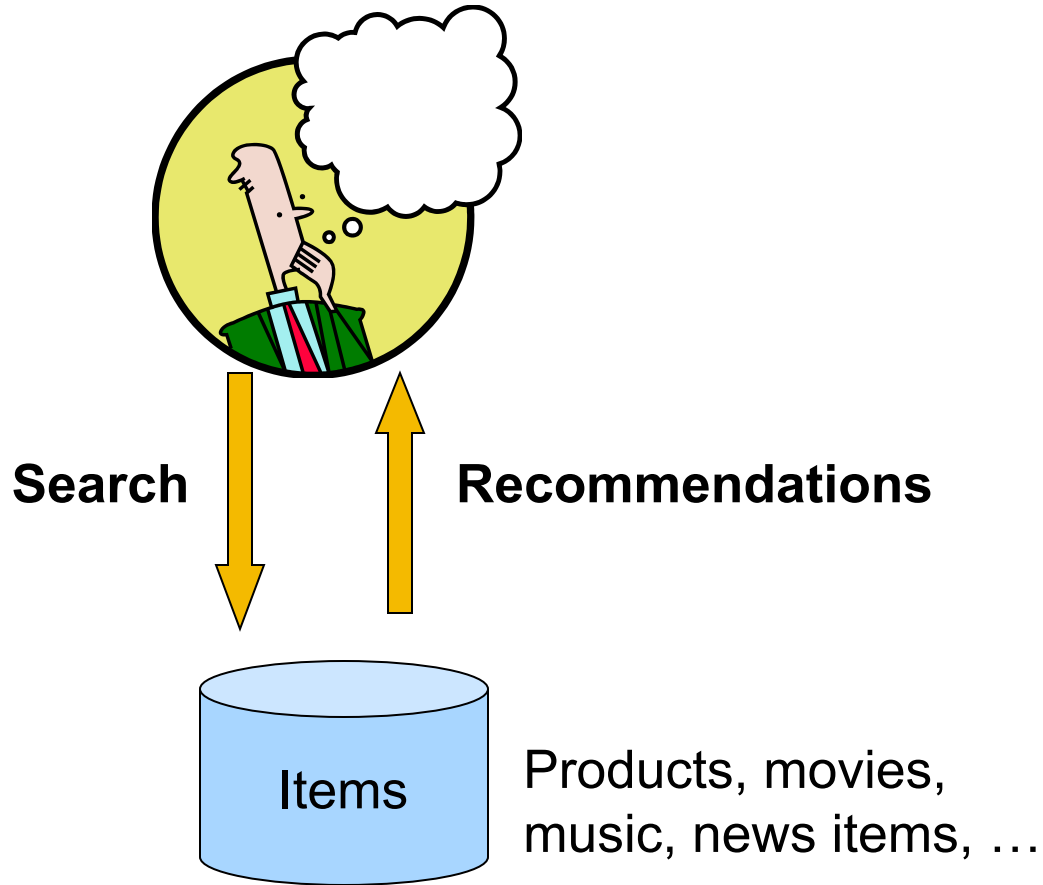**Mining of Massive Datasets**
Leskovec, Rajaraman, and Ullman
Stanford University

Recommendations

Search → **Recommendations** ←
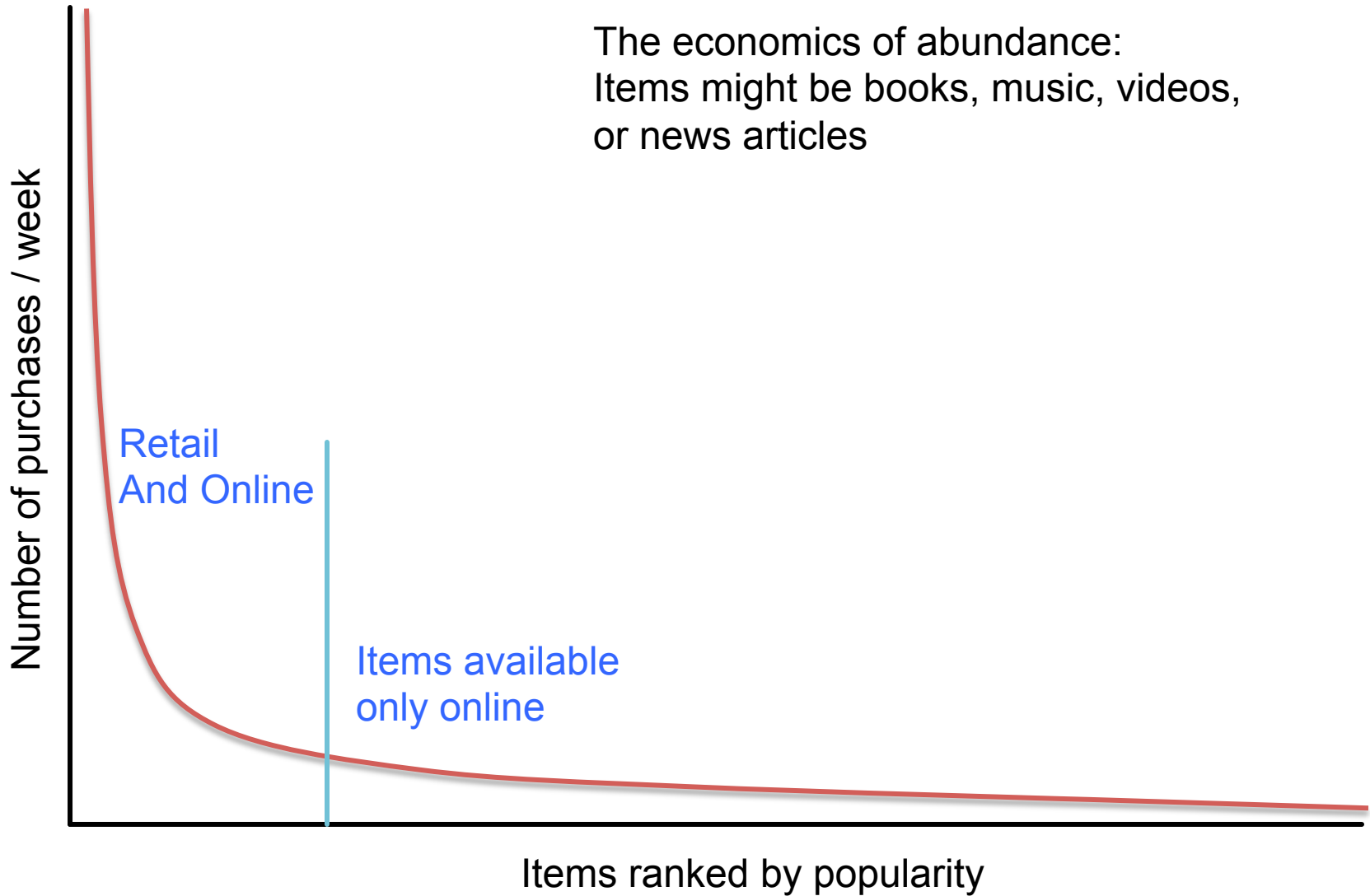
Items — Products, movies, music, news items, …

From Scarcity to Abundance

- Shelf space is a scarce commodity for traditional retailers
  - Also: TV networks, movie theaters,…

- The web enables near-zero-cost dissemination of information about products
  - From scarcity to abundance
  - Gives rise to the "Long Tail" phenomenon

The Long Tail (1)

The economics of abundance:
Items might be books, music, videos,
or news articles

Number of purchases / week

Retail
And Online

Items available
only online

Items ranked by popularity

The Long Tail (2)

- 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)

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, Pandora ...
  - Our focus here

Formal Model

- **_C_** = set of **Customers**
- **_S_** = set of **Items**

- **Utility function** $u: C \times S \to R$
  - **_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

(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 scale: only a small fraction of users leave ratings and reviews

- **Implicit**
  - Learn ratings from user actions
    - E.g., purchase implies high rating
  - What about low ratings?

(2) Extrapolating Utilities

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

- Three approaches to recommender systems
  - **1)** Content-based
  - **2)** Collaborative
  - **3)** Latent factor based