# Recommender System

# Recommendations


Products, movies, music, news items, …

# Why Recommender System?

## 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

- 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)

# Non Personalized Recommender System

- 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 $u$: $C \times S \rightarrow R$
  - $R$ = 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

**(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

# 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***

# Similar Users (1)

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

- Consider users ***x*** and ***y*** with rating vectors $\boldsymbol{r_x}$ and $\boldsymbol{r_y}$
- We need a similarity metric sim(***x***, ***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)
- Problem: Ignores rating values!

# 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

# Problem with Cosine similarity

- It treats missing ratings as negative
- Missing values are considered to be zero (low rating)
- Consider average ratings

# 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 |
|---|---|---|---|---|---|---|---|
| $A$ | $2/3$ | | | $5/3$ | $-7/3$ | | |
| $B$ | $1/3$ | $1/3$ | $-2/3$ | | | | |
| $C$ | | | | $-5/3$ | $1/3$ | $4/3$ | |
| $D$ | | $0$ | | | | | $0$ |

# Centered Cosine similarity (2)

| | 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"

# User-User Collaborative Filtering

## Rating Predictions

- Let $r_x$ be the vector of user $x$'s ratings
- Let $N$ be the set of $k$ users most similar to $x$ who have also rated item $i$
- Prediction for user $x$ and item $i$

- Option 1: $r_{xi} = 1/k \sum_{y \in N} r_{yi}$
- Option 2: $r_{xi} = \sum_{y \in N} s_{xy} r_{yi} / \sum_{y \in N} s_{xy}$

  where $s_{xy} = \text{sim}(x,y)$

# Item-Item 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

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

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

# Item-Item CF (|N|=2)

users

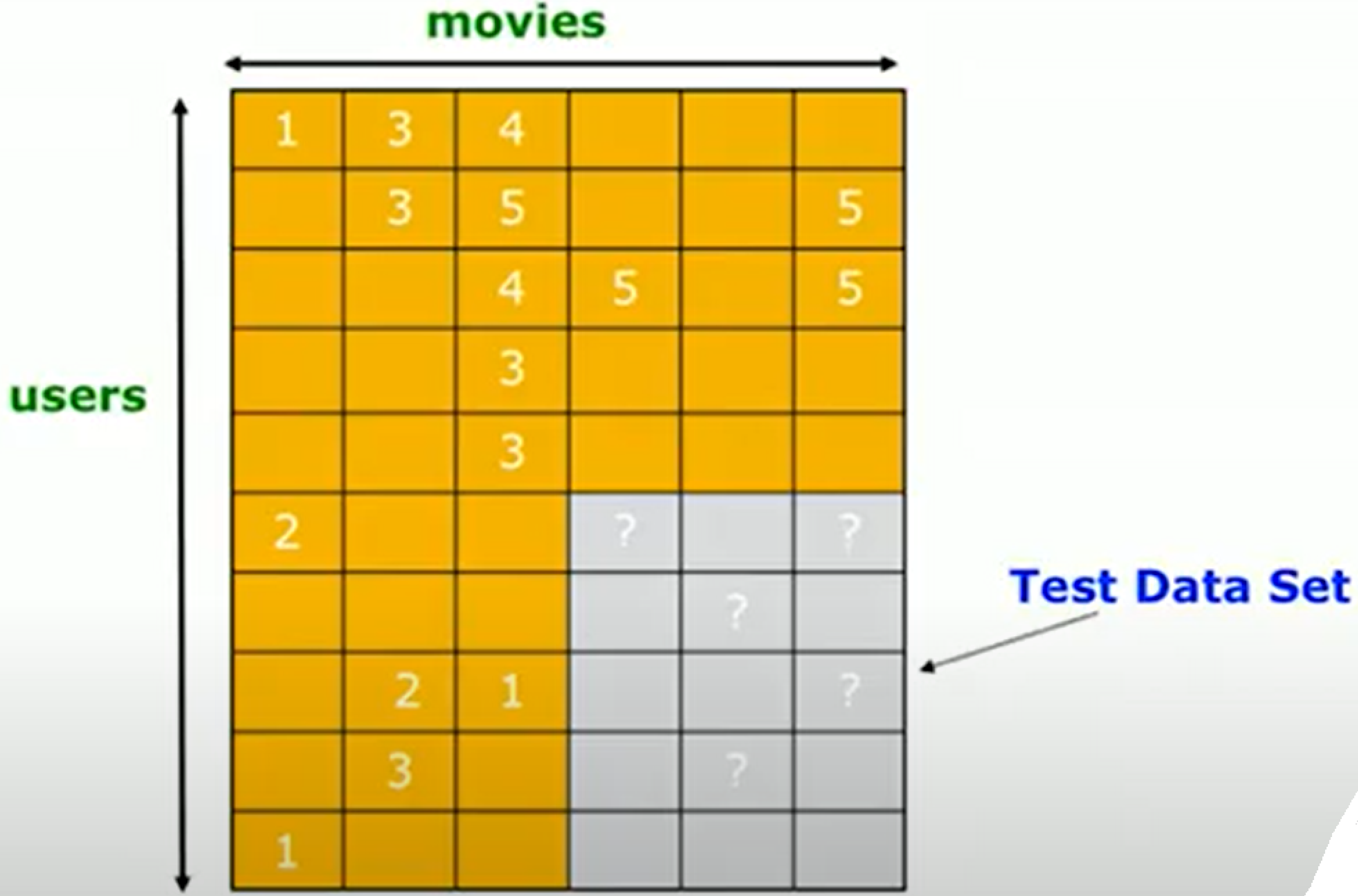
| movies | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | | 3 | | | 5 | | | 5 | | 4 | |
| 2 | | | 5 | 4 | | | 4 | | | 2 | 1 | 3 |
| 3 | 2 | 4 | | 1 | 2 | | 3 | | 4 | 3 | 5 | |
| 4 | | 2 | 4 | | 5 | | | 4 | | | 2 | |
| 5 | | | 4 | 3 | 4 | 2 | | | | | 2 | 5 |
| 6 | 1 | | 3 | | 3 | | | 2 | | | 4 | |

- unknown rating

- rating between 1 to 5

# Item-Item v. 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


movies
users
Test Data Set

# Evaluating Predictions

- Compare predictions against withheld ratings (test set $T$)

- Root-mean-square error (RMSE)

$$\sqrt{\frac{\sum_{(x,i)\in T}(r_{xi}-r^*_{xi})^2}{N}}$$

where $N = |T|$

$r_{xi}$ is the predicted rating

$r_{xi}^*$ is the actual rating