Introduction to Recommender Systems
Recommender systems: The task

Plays an Ella Fitzgerald song
What should we recommend next?

Customer W
Recommendations

Search

Recommendations

Items

Products, web sites, blogs, news items, …

Slides adapted from Jure Leskovec, CS246 and J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets
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, Apple Music...
Knowing how personalized recommendations work

Relevant for building practical news or product recommenders.
QAnon Supporters And Anti-Vaxxers Are Spreading a Hoax That Bill Gates Created the Coronavirus

It has no basis in reality, but that hasn't slowed its spread across Facebook and Twitter.

Las Vegas survivors furious as YouTube promotes clips calling shooting a hoax.

‘Fiction is outperforming reality’: how YouTube’s algorithm distorts truth.

How YouTube Drives People to the Internet's Darkest Corners

Google's video site often recommends divisive or misleading material, despite recent changes designed to fix the problems.
Formal Model

\[ X = \text{set of Users} \]
\[ S = \text{set of Items} \]

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

<table>
<thead>
<tr>
<th></th>
<th>Harry Potter</th>
<th>Twilight</th>
<th>Star Wars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anita</td>
<td>HP1 4</td>
<td>HP2 5</td>
<td>HP3 5</td>
</tr>
<tr>
<td></td>
<td>TW 5</td>
<td>SW1 1</td>
<td>SW2 2</td>
</tr>
<tr>
<td></td>
<td>SW3 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beyonce</td>
<td>B 5</td>
<td>B 5</td>
<td>B 4</td>
</tr>
<tr>
<td></td>
<td>TW 5</td>
<td>SW1 2</td>
<td>SW2 4</td>
</tr>
<tr>
<td></td>
<td>SW3 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calvin</td>
<td>C 3</td>
<td>C 3</td>
<td>C 5</td>
</tr>
<tr>
<td></td>
<td>TW 3</td>
<td>SW1 4</td>
<td>SW2 3</td>
</tr>
<tr>
<td></td>
<td>SW3 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>David</td>
<td>D 3</td>
<td>D 3</td>
<td>D 3</td>
</tr>
</tbody>
</table>

**Figure 9.4:** The utility matrix introduced in Fig. 9.1

#### Jaccard Distance

We could ignore values in the matrix and focus only on the sets of items rated. If the utility matrix only reflected purchases, this measure would be a good one to choose. However, when utilities are more detailed ratings, the Jaccard distance loses important information.

**Example 9.7:** A and B have an intersection of size 1 and a union of size 5. Thus, their Jaccard similarity is 1/5, and their Jaccard distance is 4/5; i.e., they are very far apart. In comparison, A and C have a Jaccard similarity of 2/4, so their Jaccard distance is the same, 1/2. Thus, A appears closer to C than to B. Yet that conclusion seems intuitively wrong. A and C disagree on the two movies they both watched, while A and B seem both to have liked the one movie they watched in common.

#### Cosine Distance

We can treat blanks as a 0 value. This choice is questionable, since it has the effect of treating the lack of a rating as more similar to disliking the movie than liking it.

**Example 9.8:** The cosine of the angle between A and B is: 

\[
\cos \theta = \frac{\sum_{i=1}^{N} (A_i \times B_i)}{\sqrt{\sum_{i=1}^{N} A_i^2} \cdot \sqrt{\sum_{i=1}^{N} B_i^2}}
\]

\[
\cos \theta = \frac{4 \times 5 + 5 \times 5 + 1 \times 1}{\sqrt{4^2 + 5^2 + 1^2} \cdot \sqrt{5^2 + 5^2 + 1^2}}
\]

\[
= \frac{38}{\sqrt{28} \cdot \sqrt{28}}
\]

\[
= \frac{38}{28} = 0.380
\]
Key Problems

1. Gathering “known” ratings for matrix
   ◦ How to collect the data in the utility matrix

2. Extrapolate unknown ratings from 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 performance of recommendation methods

Slides adapted from Jure Leskovec, CS246 and J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets
(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 (or watch video, or read article) implies high rating

Slides adapted from Jure Leskovec, CS246 and J. Leskovec, A. Rajaraman, J. Ullman: *Mining of Massive Datasets*
(2) Extrapolating Utilities

Key problem: Utility matrix $U$ is sparse

- Most people have not rated most items

- The "Cold Start" Problem:
  - New items have no ratings
  - New users have no history
(2) Extrapolating Utilities

Three approaches to recommender systems:

1. Content-based
2. Collaborative Filtering
3. Latent factor (Neural embedding) based

This lecture!
Content-based vs. Collaborative Filtering

Database
- Ella Fitzgerald: Jazz, Mid-20th century, vocal legend, famous duets, ...
- Louis Armstrong: Jazz, Mid-20th century, vocal legend, famous duets, ...

Customer W
- Plays Ella Fitzgerald
- What should we recommend next?

Customer D
- Plays Ella Fitzgerald
- Plays Louis Armstrong

Content-based
- Suggest Louis Armstrong

Collaborative filtering
Introduction to Recommender Systems
Recommender Systems and Collaborative Filtering

Content-based Recommender Systems
Content-based Recommendations

Main idea: Recommend items to customer $x$ similar to previous items rated highly by $x$

Movie recommendations
- Recommend movies with same actor(s), director, genre, ...

Websites, blogs, news
- Recommend other sites with similar types or words

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

- Item profiles
  - Red Circles
  - Triangles

User profile

- likes
- build
- match
- recommend

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

For each item, create an **item profile**

Profile is a set (vector) of features
- **Movies:** genre, director, actors, year...
- **Text:** Set of “important” words in document

How to pick important features?
- **TF-IDF** (Term frequency * Inverse Doc Frequency)
- For example use all words whose tf-idf > threshold, normalized for document length
Content-based Item Profiles

<table>
<thead>
<tr>
<th>Melissa McCarthy</th>
<th>Actor A</th>
<th>Actor B</th>
<th>…</th>
<th>Johnny Depp</th>
<th>Comic Genre</th>
<th>Spy Genre</th>
<th>Pirate Genre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie X</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Movie Y</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

But what if we want to have real or ordinal features too?
Content-based Item Profiles

<table>
<thead>
<tr>
<th>Actor A</th>
<th>Actor B</th>
<th>...</th>
<th>Johnny Depp</th>
<th>Comic Genre</th>
<th>Spy Genre</th>
<th>Pirate Genre</th>
<th>Avg Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie X</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Movie Y</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

For example "average rating"
Maybe we want a scaling factor $\alpha$ between binary and numeric features
### Content-based Item Profiles

<table>
<thead>
<tr>
<th>Melissa McCarthy</th>
<th>Actor A</th>
<th>Actor B</th>
<th>...</th>
<th>Johnny Depp</th>
<th>Comic</th>
<th>Spy Genre</th>
<th>Pirate Genre</th>
<th>Avg Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Movie X</strong></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><strong>Movie Y</strong></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Scaling factor $\alpha$ between binary and numeric features

$$\text{Cosine}(\text{Movie X, Movie Y}) = \frac{2 + 12\alpha^2}{\sqrt{5 + 9\alpha^2} \sqrt{5 + 16\alpha^2}}$$

$\alpha = 1: 0.82 \quad \alpha = 2: 0.94 \quad \alpha = 0.5: 0.69$

---

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

Want a vector with the same components/dimensions as items

- Could be 1s representing user purchases
- Or arbitrary numbers from a rating

User profile is aggregate of items:

- Weighted average of rated item profiles

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

• Items are movies
• Utility matrix has 1 if user has seen movie
• 20% of the movies user U has seen have Melissa McCarthy
• \( U["Melissa McCarthy"] = 0.2 \)

<table>
<thead>
<tr>
<th>User U</th>
<th>Melissa McCarthy</th>
<th>Actor A</th>
<th>Actor B</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.2</td>
<td>.005</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Prediction

- Users and items have the same dimensions!

<table>
<thead>
<tr>
<th>Movie i</th>
<th>Melissa McCarthy</th>
<th>Actor A</th>
<th>Actor B</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>

| User x  | 0.2   | .005  | 0   | 0   | 0   |

- So just recommend the items whose vectors are most similar to the user vector!

- Given user profile $x$ and item profile $i$,
  - estimate $u(x, i) = \cos(x, i) = \frac{x \cdot i}{||x|| \cdot ||i||}$

Slides adapted from Jure Leskovec, CS246 and J. Leskovec, A. Rajaraman, J. Ullman: *Mining of Massive Datasets*
Pros: Content-based Approach

+: No need for data on other users
  ◦ No user sparsity problems

+: Able to recommend to users with unique tastes

+: Able to recommend new & unpopular items
  ◦ No first-rater problem

+: Able to provide explanations
  ◦ Just list the 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
Content-based Recommender Systems
Collaborative Filtering: User-User
Collaborative filtering

Instead of using content features of items to determine what to recommend
Find similar users and recommend items that they like!
Consider user $x$ and unrated item $i$.

Find set $N$ of other users whose ratings are “similar” to $x$’s ratings.

Estimate $x$’s ratings for $i$ based on ratings for $i$ of users in $N$. 

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

Find similar users and recommend items that they like:

- Represent users by their rows in the **utility matrix**
- Two users are similar if their vectors are similar!

<table>
<thead>
<tr>
<th></th>
<th>Harry Potter</th>
<th>Twilight</th>
<th>Star Wars</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HP1</td>
<td>HP2</td>
<td>HP3</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Finding Similar Users

Let \( r_x \) be the vector of user \( x \)'s ratings

\[
\begin{align*}
  r_x &= [*, _, _, *, **] \\
  r_y &= [*, _, **, **, _]
\end{align*}
\]

\[
\begin{align*}
  r_x &= \{1, 0, 0, 1, 3\} \\
  r_y &= \{1, 0, 2, 2, 0\}
\end{align*}
\]

Cosine similarity measure

\[
\text{sim}(x, y) = \cos(r_x, r_y) = \frac{r_x \cdot r_y}{||r_x|| \cdot ||r_y||}
\]

Problem: This representation leads to unintuitive results
Problems with raw utility matrix cosine

<table>
<thead>
<tr>
<th></th>
<th>HP1</th>
<th>HP2</th>
<th>HP3</th>
<th>TW</th>
<th>SW1</th>
<th>SW2</th>
<th>SW3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>4</td>
<td></td>
<td></td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>C</strong></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td><strong>D</strong></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

**Intuitively we want:** $\text{sim}(A, B) > \text{sim}(A, C)$

\[
\text{sim}(A, B) = \frac{4 \times 5}{\sqrt{4^2 + 5^2 + 1^2} \sqrt{5^2 + 5^2 + 4^2}} = 0.380
\]

\[
\text{sim}(A, C) = \frac{5 \times 2 + 1 \times 4}{\sqrt{4^2 + 5^2 + 1^2} \sqrt{2^2 + 4^2 + 5^2}} = 0.322
\]

Yes, 0.380 > 0.322
But only barely works...
Problem with raw cosine

<table>
<thead>
<tr>
<th></th>
<th>HP1</th>
<th>HP2</th>
<th>HP3</th>
<th>TW</th>
<th>SW1</th>
<th>SW2</th>
<th>SW3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td></td>
<td></td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

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
Mean-Centered Utility Matrix: subtract the means of each row

<table>
<thead>
<tr>
<th></th>
<th>HP1</th>
<th>HP2</th>
<th>HP3</th>
<th>TW</th>
<th>SW1</th>
<th>SW2</th>
<th>SW3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>4</td>
<td>5</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>C</strong></td>
<td>3</td>
<td></td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>D</strong></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HP1</th>
<th>HP2</th>
<th>HP3</th>
<th>TW</th>
<th>SW1</th>
<th>SW2</th>
<th>SW3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td>2/3</td>
<td></td>
<td></td>
<td>5/3</td>
<td>-7/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>B</strong></td>
<td>1/3</td>
<td>1/3</td>
<td>-2/3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>C</strong></td>
<td></td>
<td></td>
<td></td>
<td>-5/3</td>
<td>1/3</td>
<td>4/3</td>
<td></td>
</tr>
<tr>
<td><strong>D</strong></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Now a 0 means no information
- And negative ratings means viewers with opposite ratings will have vectors in opposite directions!
Modified Utility Matrix: subtract the means of each row

<table>
<thead>
<tr>
<th></th>
<th>HP1</th>
<th>HP2</th>
<th>HP3</th>
<th>TW</th>
<th>SW1</th>
<th>SW2</th>
<th>SW3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2/3</td>
<td></td>
<td></td>
<td>5/3</td>
<td>-7/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1/3</td>
<td>1/3</td>
<td>-2/3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>-5/3</td>
<td>1/3</td>
<td>4/3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\text{Cos}(A,B) = \frac{(2/3) \times (1/3)}{\sqrt{(2/3)^2 + (5/3)^2 + (-7/3)^2} \sqrt{(1/3)^2 + (1/3)^2 + (-2/3)^2}} = 0.092
\]

\[
\text{Cos}(A,C) = \frac{(5/3) \times (-5/3) + (-7/3) \times (1/3)}{\sqrt{(2/3)^2 + (5/3)^2 + (-7/3)^2} \sqrt{(-5/3)^2 + (1/3)^2 + (4/3)^2}} = -0.559
\]

Now A and C are (correctly) way further apart than A,B
Terminological Note: subtracting the mean is **mean-centering**, not **normalizing** (normalizing is dividing by a norm to turn something into a probability), but the textbook (and common usage) sometimes overloads the term “normalize”
Finding similar users with overlapping-item mean-centering

Let $r_x$ be the vector of user $x$'s ratings

$r_x = \{1, 0, 0, 1, 3\} \\
r_y = \{1, 0, 2, 2, 0\}$

Mean-centering:

- For each user $x$, let $\bar{r}_x$ be mean of $r_x$ (ignoring missing values)
- $\bar{r}_x = (1 + 1 + 3)/3 = 5/3$ \\
- $\bar{r}_y = (1 + 2 + 2)/3 = 5/3$
- Subtract this average from each of their ratings
- (but do nothing to the "missing values"; they stay "null").
- mean centered $r_x = \{-2/3, 0, 0, -2/3, 4/3\}$

One new idea: Keep only items they both rate (unlike 2 slides ago)

$r_x = \{-2/3, \_\_\_\_\_\, , -2/3, \_\_\_\_\_\\} \\
r_y = \{-2/3, \_\_\_\_\_\, , \_\_\_\_\_\_\, , 1/3, \_\_\_\_\_\\}$

Now take cosine:

- Now compute cosine between user vectors
- $\cos([-2/3, -2/3], [-2/3, 1/3])$
Mean-centered overlapping-item cosine similarity

Let \( r_x \) be the vector of user \( x \)'s ratings, and \( \bar{r}_x \) be its mean (ignoring missing values)

Instead of basic cosine similarity measure

\[
\text{sim}(x, y) = \cos(r_x, r_y) = \frac{r_x \cdot r_y}{||r_x|| \ ||r_y||}
\]

Mean-centered overlapping-item cosine similarity

- \( S_{xy} \) = items rated by both users \( x \) and \( y \)

\[
\text{sim}(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}
\]

Variant of Pearson correlation

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

From similarity metric to recommendations for an unrated item $i$:

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 rated item $i$

Prediction for item $i$ of user $x$:

- Rate $i$ as the mean of what $k$-people-like-me rated $i$
  \[ r_{xi} = \frac{1}{k} \sum_{y \in N} r_{yi} \]
- Even better: Rate $i$ as the mean weighted by their similarity to me ...  
  \[ r_{xi} = \frac{\sum_{y \in N} s_{xy} r_{yi}}{\sum_{y \in N} s_{xy}} \]

- Many other tricks possible...

Shorthand:

\[ s_{xy} = \text{sim}(x, y) \]
Collaborative Filtering: User-User
Collaborative Filtering: Item-Item
Collaborative Filtering Version 2: Item-Item Collaborative Filtering

So far: **User-user collaborative filtering**

Alternate view that often works better: **Item-item**
- For item $i$, find other similar items
- Estimate rating for item $i$ based on ratings for those similar items
- Can use same similarity metrics and prediction functions as in user-user model
- "Rate $i$ as the mean of my ratings for other items, weighted by their similarity to $i"\)

\[
\hat{r}_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}
\]

$N(i;x)$...set of items rated by $x$ and similar to $i$
$s_{ij}$... similarity of items $i$ and $j$
$r_{xj}$...rating of user $x$ on item $j$
### Item-Item CF (|N|=2)

#### Users

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Movies

- **Unknown Rating**: 
  - Rating between 1 to 5

- **Rating**: 
  - Rating between 1 to 5
Item-Item CF ($|N| = 2$)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>?</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- estimate rating of movie 1 by user 5

Slides adapted from Jure Leskovec, CS246 and J. Leskovec, A. Rajaraman, J. Ullman: *Mining of Massive Datasets*
### Item-Item CF ($|N|=2$)

**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**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>3</td>
<td>?</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>5</td>
<td>4</td>
<td></td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td></td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Movies**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>3</td>
<td>?</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>5</td>
<td>4</td>
<td></td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td></td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Sim(1,m)**

1.00

..

?

..
**Item-Item CF (|N|=2)**

**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

<table>
<thead>
<tr>
<th>movies</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-13/5</td>
<td>-3/5</td>
<td>?</td>
<td>7/5</td>
<td>7/5</td>
<td>2/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-8/5</td>
<td>2/5</td>
<td>2/5</td>
<td>-3/5</td>
<td>7/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Subtract mean rating $m_i$ from each movie $i$

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

Showing computation only for #3 and #6

**users**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>?</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Item-Item CF ($|N|=2$)

### 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

<table>
<thead>
<tr>
<th>users</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-13/5</td>
<td>-3/5</td>
<td>?</td>
<td>7/5</td>
<td>7/5</td>
<td>2/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-8/5</td>
<td>2/5</td>
<td>2/5</td>
<td>-3/5</td>
<td>7/5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**sim(1,m)**

- 1.00
- ..
- ?
- ..
- ..
- ?
Compute Cosine Similarity:

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

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

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

\[
sim(1, 6) = \frac{(-13/5)(-8/5)+(-3/5)(2/5)+(2/5)(7/5)}{\sqrt{(-13/5)^2+(-3/5)^2+(2/5)^2}} \approx 0.768
\]
## Item-Item CF ($|N|=2$)

### 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*
Item-Item CF (|N|=2)
Approximate rating with weighted mean

Predict by taking weighted average:
\[ r_{1,5} = \frac{(0.658\times2 + 0.768\times3)}{(0.658+0.768)} = 2.54 \]
Item-Item vs. User-User

- **In practice, item-item often works better than user-user**
- **Why?** Items are simpler, users have multiple tastes
  - (People are more complex than objects)

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

Slides adapted from Jure Leskovec, CS246 and J. Leskovec, A. Rajaraman, J. Ullman: *Mining of Massive Datasets*
Collaborative Filtering: Item-Item
Recommender Systems and Collaborative Filtering

Simplified item-item similarity computation for our tiny PA6 dataset
Simplified item-item for our tiny PA6 dataset

First, assume you've converted all the values to
+1 (like),
0 (no rating)
–1 (dislike)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>3</td>
<td></td>
<td>5</td>
<td></td>
<td>5</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>5</td>
<td>4</td>
<td></td>
<td>4</td>
<td></td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
<td>4</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Simplified item-item for our tiny PA6 dataset

First, assume you've converted all the values to

+1  (like),
0    (no rating)
−1  (dislike)

<table>
<thead>
<tr>
<th>movies</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td></td>
<td>-1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

users
Simplified item-item for our tiny PA6 dataset

Assume you've **binarized**, i.e. converted all the values to
- +1 (like), 0 (no rating) –1 (dislike)

For this binary case, some tricks that the TAs recommend:
- Don't mean-center users, just keep the raw +1,0,-1
- Don't normalize (i.e. don't divide the product by the sum)
- i.e., instead of this:
  \[
  r_{xi} = \frac{\sum_{j \in N(i; x)} s_{ij} r_{xj}}{\sum_{j \in N(i; x)} s_{ij}}
  \]
- Just do this:
  \[
  r_{xi} = \sum_{\text{\(j \in N(i; x)\)}} s_{ij} r_{xj}
  \]
- Don't use mean-centered item-overlap cosine to compute \(s_{ij}\)
- Just use cosine
1. binarize, i.e. convert all values to
   ° +1 (like), 0 (no rating) −1 (dislike)
2. The user x gives you (say) ratings for 2 movies m1 and m2
   ° \( r_{xj} \)…rating of user x on item j
3. For each movie i in the dataset
   ° \( r_{xi} = \sum_{j \in (m1,m2)} s_{ij} r_{xj} \)
   ° Where \( s_{ij} \)… cosine between vectors for movies i and j
4. Recommend the movie i with max \( r_{xi} \)
Simplified item-item similarity computation for our tiny PA6 dataset
Evaluation and Implications
YouTube's Recommendation Algorithm

Covington, Adams, Sargin 2016. Deep Neural Networks for YouTube Recommendations

1. Represent each video and user as an embedding
2. Train a huge neural net classifier (softmax over millions of possible videos) to predict the next video the user will watch
3. Input features:
   - User's watch history (video ids)
   - User's recent queries (word embeddings)
   - Date, popularity, virality of video
4. Learn **embeddings** for **videos** and **users** in training
Evaluation

movies

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

users

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

Test Data Set
Evaluating Predictions

Compare predictions with known ratings

- **Root-mean-square error (RMSE)**

\[
\sqrt{\frac{\sum_{x_i}(r_{xi}-r_{xi}^*)^2}{N}}
\]

- where \(r_{xi}\) is predicted, \(r_{xi}^*\) is the true rating of \(x\) on \(i\)

- **Rank Correlation:**

  - Spearman’s *correlation* between system’s and user’s complete rankings

Slides adapted from Jure Leskovec, CS246 and J. Leskovec, A. Rajaraman, J. Ullman: *Mining of Massive Datasets*
But is predicting watching the right loss function?
What could go wrong? Ethical and societal implications in recommendation engines.


- Spread of misinformation and propaganda
- Filter bubbles
- Inappropriate or unethical content
- Opacity
- Violating user privacy
What could go wrong? Ethical and societal implications

Propaganda campaigns

- Russia Internet Research Agency (IRA)
  - attack on the United States 2013-2018
  - computational propaganda on YouTube, Facebook, Instagram, to misinform/polarize US voters.
  - Goal: induce African American, Mexican American voters to boycott elections

Ethical and societal implications: Filter bubbles

How YouTube Drives People to the Internet's Darkest Corners

Google’s video site often recommends divisive or misleading material, despite recent changes designed to fix the problems.

“I realized really fast that YouTube’s recommendation was putting people into filter bubbles,” Chaslot said. “There was no way out. If a person was into Flat Earth conspiracies, it was bad for watch-time to recommend anti-Flat Earth videos, so it won’t even recommend them.”
“The question before us is the ethics of leading people down hateful rabbit holes full of misinformation and lies at scale just because it works to increase the time people spend on the site – and it does work”

– Zeynep Tufekci
Open research questions

What would algorithms look like that could recommend but also include these social costs?
Evaluation and Implications