



## Recommendation in Social Media

# SOCIAL MEDIA MINING



# Dear instructors/users of these slides:

Please feel free to include these slides in your own material, or modify them as you see fit. If you decide to incorporate these slides into your presentations, please include the following note:

R. Zafarani, M. A. Abbasi, and H. Liu, *Social Media Mining: An Introduction*, Cambridge University Press, 2014.  
Free book and slides at **<http://socialmediamining.info/>**

or include a link to the website:

**<http://socialmediamining.info/>**

# Difficulties of Decision Making

- Which digital camera should I buy?
- Where should I spend my holiday?
- Which movie should I rent?
- Whom should I follow?
- Where should I find interesting news article?
- Which movie is the best for our family?
  
- If interested, see two recent conference tutorials
  - SIGKDD2014, Recommendation in Social Media
  - RecSys2014, Personalized Location Recommendation

# When Does This Problem Occur?

- There are many choices
- There are no obvious advantages among them
- We do not have enough resources to check all options (**information overload**)
- We do not have enough knowledge and experience to choose, or
  - I'm lazy, but don't want to miss out on good stuff
  - Defensive decision making

**Goal of Recommendation:**  
**To come up with a short list of items that fits user's interests**

# Common Solutions to the Problem

- Consulting friends
- Obtaining information from a trusted third party
- Hiring a team of experts
- Search the Internet
- Following the crowd
  - Pick the item from top- $n$  lists
  - Best sellers on Amazon
- **Can we automate all of the above?**
  - **Using a recommender algorithm**
  - **Also known as recommender systems**

# Recommender Systems - Examples

## Book recommendation in Amazon

The screenshot shows the product page for 'Networks: An Introduction' by Mark Newman. It highlights the 'Frequently Bought Together' section, which lists four related books: 'Networks, Crowds, and Markets: Reasoning About a Highly Connected World' by David Easley, 'Simply Complexity: A Clear Guide to Complexity Theory' by Neil Johnson, 'Networks, Crowds, and Markets: Reasoning About a Highly Connected World' by David Easley, and 'Social Network Analysis: Methods and Applications' by Steven Wasserman. Below this, the 'Customers Who Bought This Item Also Bought' section is circled in red and lists five more books: 'Networks, Crowds, and Markets: Reasoning About a... by David Easley', 'Dynamical Processes on Complex Networks by Alan Newell', 'Simply Complexity: A Clear Guide to Complexity Theory' by Neil Johnson, 'Social Network Analysis: Methods and Applications' by Steven Wasserman, and 'Networks of the Brain by Olaf Sporns'. The entire 'Customers Who Bought This Item Also Bought' section is highlighted with a red border.

## Video clip recommendation in YouTube

The screenshot shows a YouTube video titled 'Arizona Wildfire Near Flagstaff at 10,000 Acres' uploaded by fal2grace. The video has 510 views and 15 likes. The 'Suggestions' sidebar on the right side of the video player is circled in red and displays several other wildfire-related videos from various sources, such as 'Schultz Fire - Flagstaff, AZ - June 20, 2010' and 'Flagstaff Father's Day Fire #2 - Schultz Wildfire'.

## Product Recommendation in ebay

The screenshot shows the ebay homepage. It features a 'Recommendations for you' section with items like 'Dr. Seuss's Second Beginner Book Collection' and 'The Cat in the Hat'. Below this is a 'Popular on ebay' section with items like 'AAA 1000MAH RECHARGEABLE BATTERY X4' and '8x 3000mAh'. There is also an 'eBay stories' section with a link to 'eBay's hidden gem: eBay Radio'. The bottom of the page includes a 'Support Toys for Tots' banner and links for buying and selling on ebay.

## Restaurant Recommendation in Yelp

The screenshot shows the Yelp search results for 'restaurants' in 'Tempe, AZ'. The results page includes a map of Tempe with restaurant locations marked. The top result is '1. The Dhaba' with a \$10 for \$20 Certificate. Other results include '2. China Farm Chinese Buffet' with a \$5 for \$10 Certificate and '3. Capriotti's Sandwich Shop' with a \$7 for \$15 Certificate. The page also features a 'Concierge' button and a 'Less Map' button.

# Main Idea behind Recommender Systems

**Use historical data such as the user's past preferences or similar users' past preferences to predict future likes**

- Users' preferences are likely to remain stable, and change smoothly over time.
  - By watching the past users' or groups' preferences, we try to predict their future interests
  - Then we can recommend items of interest to them
- Formally, a recommender system takes a set of users  $U$  and a set of items  $I$  and learns a function  $f$  such that:

$$f : U \times I \rightarrow \mathbb{R}$$

# Recommendation vs. Search

- One way to get answers is using search engines
- Search engines find results that match the query provided by the user
- The results are generally provided as a list ordered with respect to the relevance of the item to the given query
- Consider the query “**best 2014 movie to watch**”
  - The same results for an 8 year old and an adult

**Search engines' results are not customized**

# Challenges of Recommender Systems

- **The Cold Start Problem**

- Recommender systems use historical data or information provided by the user to recommend items, products, etc.
- When user join sites, they still haven't bought any product, or they have no history.
- It is hard to infer what they are going to like when they start on a site.

- **Data Sparsity**

- When historical or prior information is insufficient.
- Unlike the cold start problem, this is in the system as a whole and is not specific to an individual.

# Challenges of Recommender Systems

- **Attacks**
  - **Push Attack**: pushing ratings up by making fake users
  - **Nuke attack**: DDoS attacks, stop the whole recommendation systems
- **Privacy**
  - Using one's private info to recommend to others.
- **Explanation**
  - Recommender systems often recommend items with no explanation on why these items are recommended

# **Classical Recommendation Algorithms**

- **Content-based algorithms**
- **Collaborative filtering**

# Content-Based Methods

**Assumption:** a user's interest should match the description of the items that the user should be recommended by the system.

- The more similar the item's description to that of the user's interest, the more likely that the user finds the item's recommendation interesting.

**Goal:** find the similarity between the user and all of the existing items is the core of this type of recommender systems

# Content-based Recommendation: An Example

Book Database

Title	Genre	Author	Type	Price	Keywords
<i>The Night of the Gun</i>	Memoir	David Carr	Paperback	29.90	press and journalism, drug addiction, personal memoirs, New York
<i>The Lace Reader</i>	Fiction Mystery	Brunonia Barry	Hardcover	49.90	American contemporary fiction, detective, historical
<i>Into the Fire</i>	Romance, Suspense	Suzanne Brockmann	Hardcover	45.90	American fiction, murder, neo-Nazism
...					

User Profile

Title	Genre	Author	Type	Price	Keywords
...	Fiction, Suspense	Brunonia Barry, Ken Follett	Paperback	25.65	detective, murder, New York

# Content-based Recommendation Algorithm

1. Describe the items to be recommended
2. Create a profile of the user that describes the types of items the user likes
3. Compare items with the user profile to determine what to recommend

*The profile is often created, and updated automatically in response to feedback on the desirability of items that are presented to the user*

# Content-based Recommendation: Example

The screenshot shows the 'Edit Favorites' section of the Amazon.com website. At the top, there are links for 'Michael's Store', 'See All 32 Product Categories', 'Your Account', 'Cart', 'Your Lists', 'Help', and a search bar. Below the search bar is a link to 'Improve Your Recommendations'. The main area is titled 'Edit Favorites' with the sub-instruction 'Mark the categories that interest you the most.' A checkbox labeled 'Books' is checked. In the 'Your Books Favorites' section, under 'Categories', there are two checkboxes: 'Biographies & Memoirs' and 'Business & Investing', both checked. There is also a checked checkbox for 'Nonfiction'. In the 'Add to Your Favorites' section, there are two columns of checkboxes: 'Arts & Photography', 'Children's Books', 'Comics & Graphic Novels', 'Cooking, Food & Wine', and 'Entertainment' in the first column; and 'Outdoors & Nature', 'Parenting & Families', 'Professional & Technical', 'Reference', and 'Religion & Spirituality' in the second column.

Items Recommended

## User Profile

The screenshot shows the 'Recommended For You' section for the 'Books' category on Amazon.com. The title 'Recommended For You > Books' is at the top. It states that recommendations are based on 'items you own' and more. It lists two items:

- The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture**  
by John Battelle  
Average Customer Review: ★★★★  
Publication Date: September 8, 2005  
Our Price: \$16.35  
Used & new from \$10.95  
Add to cart  
Add to Wish List
- Writing Successful Science Proposals**  
by Andrew J. Friedland, Carol L Folt  
Average Customer Review: ★★★★  
Publication Date: June 10, 2000  
Add to cart

Below the items, there are buttons for 'I Own It', 'Not interested', and a rating scale from 1 to 5 stars. It also says 'Recommended because you purchased Amazonia and more (edit)'.

# More formally

- We represent user profiles and item descriptions by vectorizing them using a set of  $k$  keywords
- We can vectorize (e.g., using **TF-IDF**) both users and items and compute their similarity

$$I_j = (i_{j,1}, i_{j,2}, \dots, i_{j,k}) \quad U_i = (u_{i,1}, u_{i,2}, \dots, u_{i,k})$$

$$\text{sim}(U_i, I_j) = \cos(U_i, I_j) = \frac{\sum_{l=1}^k u_{i,l} i_{j,l}}{\sqrt{\sum_{l=1}^k u_{i,l}^2} \sqrt{\sum_{l=1}^k i_{j,l}^2}}$$

We can recommend the top most similar items to the user

# Content-Based Recommendation Algorithm

---

## Algorithm 9.1 Content-based recommendation

---

**Require:** User  $i$ 's Profile Information, Item descriptions for items  $j \in \{1, 2, \dots, n\}$ ,  $k$  keywords,  $r$  number of recommendations.

- 1: **return**  $r$  recommended items.
  - 2:  $U_i = (u_1, u_2, \dots, u_k) =$  user  $i$ 's profile vector;
  - 3:  $\{I_j\}_{j=1}^n = \{(i_{j,1}, i_{j,2}, \dots, i_{j,k}) =$  item  $j$ 's description vector $\}_{j=1}^n$ ;
  - 4:  $s_{i,j} = sim(U_i, I_j), 1 \leq j \leq n$ ;
  - 5: Return top  $r$  items with maximum similarity  $s_{i,j}$ .
- 

- We compute the topmost similar items to a user  $j$  and then recommend these items in the order of similarity

# Collaborative Filtering

**Collaborative filtering:** the process of selecting information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc.

**Advantage:** we don't need to have additional information about the users or content of the items

- Users' rating or purchase history is the only information that is needed to work

# Rating Matrix: An Example

**Movies You've Rated**

Based on your 745 movie ratings, this is the list of movies you've seen. As you discover movies on the website that you've seen, rate them and they will show up on this list. On this page, you may change the rating for any movie you've seen, and you may remove a movie from this list by clicking the 'Clear Rating' button.

TITLE	MPAA	GENRE	STAR RATING ▾
<a href="#">Add </a> <a href="#">12 Angry Men (1957)</a>	UR	Classics	<a href="#">Clear Rating</a>
<a href="#">Add </a> <a href="#">The 39 Steps (1935)</a>	UR	Classics	<a href="#">Clear Rating</a>
<a href="#">Add </a> <a href="#">An American in Paris (1951)</a>	UR	Classics	<a href="#">Clear Rating</a>
<a href="#">Add </a> <a href="#">The Andromeda Strain (1971)</a>	G	Sci-Fi & Fantasy	<a href="#">Clear Rating</a>
<a href="#">Add </a> <a href="#">Apollo 13 (1995)</a>	PG	Drama	<a href="#">Clear Rating</a>
<a href="#">Add </a> <a href="#">The Battle of Algiers (1965) La Battaglia di Algeri</a>	UR	Foreign	<a href="#">Clear Rating</a>
<a href="#">Add </a> <a href="#">Being There (1979)</a>	PG	Drama	<a href="#">Clear Rating</a>
<a href="#">Add </a> <a href="#">Big Deal on Madonna Street (1958) I soliti ignoti</a>	UR	Foreign	<a href="#">Clear Rating</a>
<a href="#">Add </a> <a href="#">The Birds (1963)</a>	PG-13	Thrillers	<a href="#">Clear Rating</a>
<a href="#">Add </a> <a href="#">Blade Runner (1982)</a>	R	Sci-Fi & Fantasy	<a href="#">Clear Rating</a>

Value	Graphic representation	Textual representation
5		Excellent
4		Very good
3		Good
2		Fair
1		Poor



Table 9.1: User-Item Matrix

	Lion King	Aladdin	Mulan	Anastasia
John	3	0	3	3
Joe	5	4	0	2
Jill	1	2	4	2
Jane	3	?	1	0
Jorge	2	2	0	1

# Rating Matrix

Users rate (rank) items (purchased, watched)

## Explicit ratings:

- entered by a user directly
- i.e., “Please rate this on a scale of 1-5”



## Implicit ratings:

- Inferred from other user behavior
- E.g., Play lists or music listened to, for a music Rec system
- The amount of time users spent on a webpage

# Collaborative Filtering

## Types of Collaborative Filtering Algorithms:

- **Memory-based**: Recommendation is directly based on previous ratings in the stored matrix that describes user-item relations
- **Model-based**: Assumes that an underlying model (hypothesis) governs how users rate items.
  - This model can be approximated and learned.
  - The model is then used to recommend ratings.
  - **Example**: users rate low budget movies poorly

# Memory-Based Collaborative Filtering

Two memory-based methods:

## User-based CF

Users with similar **previous** ratings for items are likely to rate future items similarly

	I1	I2	I3	I4
U1	1	2	4	4
U2	1	2	4	?
U3	2	5	2	2
U4	5	2	3	3

## Item-based CF

Items that have received similar ratings **previously** from users are likely to receive similar ratings from future users

	I1	I2	I3	I4
U1	1	2	4	4
U2	1	2	4	?
U3	2	5	2	2
U4	5	2	3	3

# Collaborative Filtering: Algorithm

1. Weigh all users/items with respect to their similarity with the current user/item
2. Select a subset of the users/items (**neighbors**) as recommenders
3. Predict the rating of the user for specific items using neighbors' ratings for the same (or similar) items
4. Recommend items with the highest predicted rank

# Measuring Similarity between Users (or Items)

## Cosine Similarity

$$sim(U_u, U_v) = \cos(U_u, U_v) = \frac{U_u \cdot U_v}{\|U_u\| \|U_v\|} = \frac{\sum_i r_{u,i} r_{v,i}}{\sqrt{\sum_i r_{u,i}^2} \sqrt{\sum_i r_{v,i}^2}}.$$

## Pearson Correlation Coefficient

$$sim(U_u, U_v) = \frac{\sum_i (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_i (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_i (r_{v,i} - \bar{r}_v)^2}}$$

# User-based Collaborative Filtering

- User-based collaborative filtering
  - The system finds the most similar user (users) to the current user and uses their preferences for recommendation
- The user-based approach is not as popular as the item-based approach
  - **Why?** With large number of users, even the smallest change in the user data is likely to reset the entire group of similar users

# User-based CF

Updating the ratings:

$$r_{u,i} = \bar{r}_u + \frac{\sum_{v \in N(u)} sim(u,v)(r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u)} sim(u,v)}$$

Annotations for the equation:

- User  $u$ 's mean rating (points to  $\bar{r}_u$ )
- Predicted rating of user  $u$  for item  $i$  (points to  $r_{u,i}$ )
- User  $v$ 's mean rating (points to  $\bar{r}_v$ )
- Observed rating of user  $v$  for item  $i$  (points to  $r_{v,i}$ )

# User-based CF, Example

	Lion King	Aladdin	Mulan	Anastasia
John	3	0	3	3
Joe	5	4	0	2
Jill	1	2	4	2
Jane	3	?	1	0
Jorge	2	2	0	1

Predict Jane's rating  
for Aladdin



1- Calculate average ratings

$$\bar{r}_{John} = \frac{3 + 3 + 0 + 3}{4} = 2.25$$

$$\bar{r}_{Joe} = \frac{5 + 4 + 0 + 2}{4} = 2.75$$

$$\bar{r}_{Jill} = \frac{1 + 2 + 4 + 2}{4} = 2.25$$

$$\bar{r}_{Jane} = \frac{3 + 1 + 0}{3} = 1.33$$

$$\bar{r}_{Jorge} = \frac{2 + 2 + 0 + 1}{4} = 1.25$$

2- Calculate user-user similarity

$$sim(Jane, John) = \frac{3 \times 3 + 1 \times 3 + 0 \times 3}{\sqrt{10} \sqrt{27}} = 0.73$$

$$sim(Jane, Joe) = \frac{3 \times 5 + 1 \times 0 + 0 \times 2}{\sqrt{10} \sqrt{29}} = 0.88$$

$$sim(Jane, Jill) = \frac{3 \times 1 + 1 \times 4 + 0 \times 2}{\sqrt{10} \sqrt{21}} = 0.48$$

$$sim(Jane, Jorge) = \frac{3 \times 2 + 1 \times 0 + 0 \times 1}{\sqrt{10} \sqrt{5}} = 0.84$$

# User-based CF, Example- continued

3- Calculate Jane's rating for Aladdin, Assume that neighborhood size = 2

$$\begin{aligned} r_{Jane,Aladdin} &= \bar{r}_{Jane} + \frac{sim(Jane, Joe)(r_{Joe,Aladdin} - \bar{r}_{Joe})}{sim(Jane, Joe) + sim(Jane, Jorge)} \\ &\quad + \frac{sim(Jane, Jorge)(r_{Jorge,Aladdin} - \bar{r}_{Jorge})}{sim(Jane, Joe) + sim(Jane, Jorge)} \\ &= 1.33 + \frac{0.88(4 - 2.75) + 0.84(2 - 1.25)}{0.88 + 0.84} = 2.33 \end{aligned}$$

# Item-based CF

Calculate the similarity between items and then predict new items based on the past ratings for similar items

$$r_{u,i} = \bar{r}_i + \frac{\sum_{j \in N(i)} sim(i,j)(r_{u,j} - \bar{r}_j)}{\sum_{j \in N(i)} sim(i,j)}$$

Item  $i$ 's mean rating

$i$  and  $j$  are two items

# Item-based CF, Example

## 1- Calculate average ratings

$$\bar{r}_{Lion\ King} = \frac{3 + 5 + 1 + 3 + 2}{5} = 2.8.$$

$$\bar{r}_{Aladdin} = \frac{0 + 4 + 2 + 2}{4} = 2.$$

$$\bar{r}_{Mulan} = \frac{3 + 0 + 4 + 1 + 0}{5} = 1.6.$$

$$\bar{r}_{Anastasia} = \frac{3 + 2 + 2 + 0 + 1}{5} = 1.6.$$

## 2- Calculate item-item similarity

$$sim(Aladdin, Lion\ King) = \frac{0 \times 3 + 4 \times 5 + 2 \times 1 + 2 \times 2}{\sqrt{24} \sqrt{39}} = 0.84$$

$$sim(Aladdin, Mulan) = \frac{0 \times 3 + 4 \times 0 + 2 \times 4 + 2 \times 0}{\sqrt{24} \sqrt{25}} = 0.32$$

$$sim(Aladdin, Anastasia) = \frac{0 \times 3 + 4 \times 2 + 2 \times 2 + 2 \times 1}{\sqrt{24} \sqrt{18}} = 0.67$$

## 3- Calculate Jane's rating for Aladdin, Assume that neighborhood size = 2

$$\begin{aligned} r_{Jane, Aladdin} &= \bar{r}_{Aladdin} + \frac{sim(Aladdin, Lion\ King)(r_{Jane, Lion\ King} - \bar{r}_{Lion\ King})}{sim(Aladdin, Lion\ King) + sim(Aladdin, Anastasia)} \\ &\quad + \frac{sim(Aladdin, Anastasia)(r_{Jane, Anastasia} - \bar{r}_{Anastasia})}{sim(Aladdin, Lion\ King) + sim(Aladdin, Anastasia)} \\ &= 2 + \frac{0.84(3 - 2.8) + 0.67(0 - 1.6)}{0.84 + 0.67} = 1.40 \end{aligned}$$

# Recommendation to a Group

# Recommendation to Groups

- Find content of interest to all members of a group of socially acquainted individuals
- Examples:
  - A movie for friends to watch together
  - A travel destination for a family to spend a break
  - A good restaurant for colleagues to have lunch
  - A music to be played in a public area

# Tasks of a Group Recommender System

- Acquiring preferences
- Generating recommendations
- Explaining recommendations
- Helping group members to achieve consensus

# Aggregation Strategies

## Maximizing Average Satisfaction

- Average everyone's ratings and choose the max

$$R_i = \frac{1}{n} \sum_{u \in G} r_{u,i}$$

## Least Misery

- This approach tries to minimize the dissatisfaction among group's members (max of all mins)

$$R_i = \min_{u \in G} r_{u,i}$$

## Most Pleasure

- The maximum of individuals' maximum ratings is taken as group's rating

$$R_i = \max_{u \in G} r_{u,i}$$

# Recommendation to Group, an Example

Table 9.3: User-Item Matrix

	Soda	Water	Tea	Coffee
John	1	3	1	1
Joe	4	3	1	2
Jill	2	2	4	2
Jorge	1	1	3	5
Juan	3	3	4	5

## Average Satisfaction

$$R_{Soda} = \frac{1 + 2 + 3}{3} = 2.$$

$$R_{Water} = \frac{3 + 2 + 3}{3} = 2.66$$

$$R_{Tea} = \frac{1 + 4 + 4}{3} = 3.$$

$$R_{Coffee} = \frac{1 + 2 + 5}{3} = 2.66$$

## Least Misery

$$R_{Soda} = \min\{1, 2, 3\} = 1$$

$$R_{Water} = \min\{3, 2, 3\} = 2$$

$$R_{Tea} = \min\{1, 4, 4\} = 1$$

$$R_{Coffee} = \min\{1, 2, 5\} = 1$$

## Most Pleasure

$$R_{Soda} = \max\{1, 2, 3\} = 3$$

$$R_{Water} = \max\{3, 2, 3\} = 3$$

$$R_{Tea} = \max\{1, 4, 4\} = 4$$

$$R_{Coffee} = \max\{1, 2, 5\} = 5$$

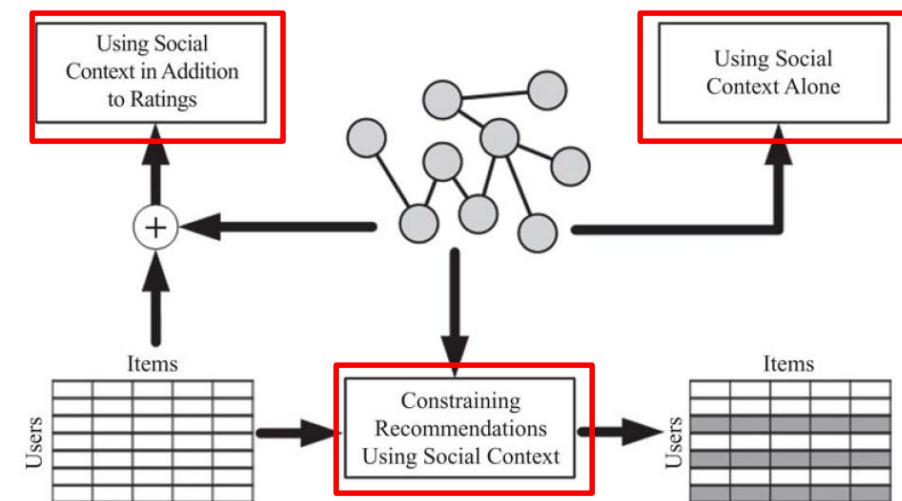
# Recommendation Using Social Context

- Recommendation using social context alone
- Extending classical methods with social context
- Recommendation constrained by social context

# Information Available in Social Context

- In social media, in addition to ratings of products, there is additional information
  - E.g., the friendship network
- This information can be used to improve recommendations

- Assuming that friends have an impact on the ratings ascribed by the individual.
- This impact can be due to homophily, influence, or confounding



# I. Recommendation Using Social Context Alone

- Consider a network of friendships for which no user-item rating matrix is provided.
- In this network, we can still recommend users from the network to other users for friendship.
- This is an example of friend recommendation in social networks [**Next Chapter!**]

### 3. Recommendation Constrained by Social Context

- In classical recommendation,
  - To estimate ratings, we determine similar users or items.
  - Any user similar to the individual can contribute to the predicted ratings for the individual.
- We can limit the set of individuals that can contribute to the ratings of a user to the set of friends of the user.
  - $S(i)$  is the set of  $k$  most similar **friends** of an individual

$$r_{u,i} = \bar{r}_u + \frac{\sum_{v \in S(u)} sim(u, v)(r_{v,i} - \bar{r}_v)}{\sum_{v \in S(u)} sim(u, v)}$$

# Example

$$A = \begin{bmatrix} & John & Joe & Jill & Jane & Jorge \\ John & 0 & 1 & 0 & 0 & 1 \\ Joe & 1 & 0 & 1 & 0 & 0 \\ Jill & 0 & 1 & 0 & 1 & 1 \\ Jane & 0 & 0 & 1 & 0 & 0 \\ Jorge & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

	Lion King	Aladdin	Mulan	Anastasia
John	4	3	2	2
Joe	5	2	1	5
Jill	2	5	?	0
Jane	1	3	4	3
Jorge	3	1	1	2

$$\bar{r}_{John} = \frac{4 + 3 + 2 + 2}{4} = 2.75.$$

$$\bar{r}_{Joe} = \frac{5 + 2 + 1 + 5}{4} = 3.25.$$

$$\bar{r}_{Jill} = \frac{2 + 5 + 0}{3} = 2.33.$$

$$\bar{r}_{Jane} = \frac{1 + 3 + 4 + 3}{4} = 2.75.$$

$$\bar{r}_{Jorge} = \frac{3 + 1 + 1 + 2}{4} = 1.75.$$

$$sim(Jill, John) = \frac{2 \times 4 + 5 \times 3 + 0 \times 2}{\sqrt{29} \sqrt{29}} = 0.79$$

$$sim(Jill, Joe) = \frac{2 \times 5 + 5 \times 2 + 0 \times 5}{\sqrt{29} \sqrt{54}} = 0.50$$

$$sim(Jill, Jane) = \frac{2 \times 1 + 5 \times 3 + 0 \times 3}{\sqrt{29} \sqrt{19}} = 0.72$$

$$sim(Jill, Jorge) = \frac{2 \times 3 + 5 \times 1 + 0 \times 2}{\sqrt{29} \sqrt{14}} = 0.54$$

Average Ratings

User Similarity

$$\begin{aligned}
 r_{Jill,Mulan} &= \bar{r}_{Jill} + \frac{sim(Jill, Jane)(r_{Jane,Mulan} - \bar{r}_{Jane})}{sim(Jill, Jane) + sim(Jill, Jorge)} \\
 &\quad + \frac{sim(Jill, Jorge)(r_{Jorge,Mulan} - \bar{r}_{Jorge})}{sim(Jill, Jane) + sim(Jill, Jorge)} \\
 &= 2.33 + \frac{0.72(4 - 2.75) + 0.54(1 - 1.75)}{0.72 + 0.54} = 2.72
 \end{aligned}$$

# Evaluation of Recommender Systems

# Evaluating Recommender Systems is difficult

- Different algorithms may be better or worse on different datasets (applications)
  - Many algorithms are designed specifically for datasets where there are many more users than items or vice versa.
  - Similar differences exist for rating density, rating scale, and other properties of datasets
- The goals to perform evaluation may differ
  - Early evaluation work focused specifically on the "accuracy" of algorithms in "predicting" withheld ratings.
  - Other properties different from accuracy also have important effect on user satisfaction and performance
- There is a significant challenge in deciding what combination of measures should be used in comparative evaluation

# Evaluating Recommender Systems

- A myriad of algorithms are proposed, **but**
  - Which one is the best in a given application domain?
  - What are the success factors of different algorithms?
  - Comparative analysis based on an optimality criterion?

Main questions are:

- Is a RS efficient with respect to specific criteria like accuracy, user satisfaction, response time, etc.
- Do customers like/buy recommended items?
- Do customers buy items they otherwise would have not?
- Are they satisfied with a recommendation after purchase?

# How Do We Evaluate Recommenders

- Application outcomes
  - Add-on sales
  - Click-through rates
  - The number of products purchased
    - And not returned!
- Research measures
  - User satisfaction
- Metrics
  - To anticipate the above beforehand (offline)

# Accuracy Metrics

- **Predictive accuracy**
  - How close are the recommender system's predicted ratings are to the true user ratings?
- **Classification accuracy**
  - The ratio with which a recommender system makes correct vs. incorrect decisions about whether an item is good.
  - Classification metrics are thus appropriate for tasks such as *Find Good Items* when users have binary preferences.
- **Rank accuracy**

# I. Predictive accuracy - Metrics measure error rate

- **Mean Absolute Error (*MAE*).**

The average absolute deviation between a predicted rating ( $p$ ) and the user's true rating ( $r$ )

- $NMAE = MAE / (r_{max} - r_{min})$

$$MAE = \frac{\sum_{ij} |\hat{r}_{ij} - r_{ij}|}{n}$$

- **Root Mean Square Error (*RMSE*).**

Similar to *MAE*, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{1}{n} \sum_{i,j} (\hat{r}_{ij} - r_{ij})^2}$$

# Evaluation Example

<i>Item</i>	<i>Predicted Rating</i>	<i>True Rating</i>
1	1	3
2	2	5
3	3	3
4	4	2
5	4	1

$$MAE = \frac{|1 - 3| + |2 - 5| + |3 - 3| + |4 - 2| + |4 - 1|}{5} = 2$$

$$NMAE = \frac{MAE}{5 - 1} = 0.5$$

$$\begin{aligned} RMSE &= \sqrt{\frac{(1 - 3)^2 + (2 - 5)^2 + (3 - 3)^2 + (4 - 2)^2 + (4 - 1)^2}{5}} \\ &= 2.28 \end{aligned}$$

## II. Classification Accuracy: Precision and Recall

**Precision:** a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved

$$P = \frac{N_{rs}}{N_s}$$

**Recall:** a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items

$$R = \frac{N_{rs}}{N_r}$$

	Selected	Not Selected	Total
Relevant	$N_{rs}$	$N_m$	$N_r$
Irrelevant	$N_{is}$	$N_{in}$	$N_i$
Total	$N_s$	$N_n$	$N$

# Evaluating Relevancy, Example

	<i>Selected</i>	<i>Not Selected</i>	<i>Total</i>
<i>Relevant</i>	9	15	24
<i>Irrelevant</i>	3	13	16
<i>Total</i>	12	28	40

$$P = \frac{9}{12} = 0.75$$

$$R = \frac{9}{24} = 0.375$$

$$F = \frac{2 \times 0.75 \times 0.375}{0.75 + 0.375} = 0.5$$

# III. Evaluating Ranking of Recommendation

- **Spearman's Rank Correlation**

$$\rho = 1 - \frac{6 \sum_{i=1}^n (x_i - yi)^2}{n^3 - n}$$

- **Kendall's  $\tau$** 
  - Compares concordant the items of the recommended ranking list against the ground truth ranking list
    - If the two orders are consistent, it is concordant
    - E.g., for top 4 items in ranking list, there are  $4 \times 3 / 2 = 6$  pairs

$$\tau = \frac{c-d}{\binom{n}{2}}$$

- $c$  is the number of concordants
- $d$  is the number of discordants