# Recommendation Systems

## UE 141 Spring 2013

Jing Gao
SUNY Buffalo

# Recommendation Systems

- **Data**

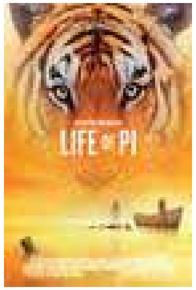
items

users

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 4 | | | |
| | 3 | 5 | | | 5 |
| | | 4 | 5 | | 5 |
| | | 3 | | | |
| | | 3 | | | |
| 2 | | | 2 | | 2 |
| | | | | 5 | |
| | 2 | 1 | | | 1 |
| | 3 | | | 3 | |
| 1 | | | | | |

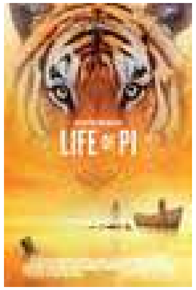
- **Goal**
  - Learn what a user might be interested in and recommend other items he might like

# Ratings from 1 to 5


| | | | | | |
|---|---|---|---|---|---|
| User 1 | 5 | 4 | 3 | 1 | 3 |
| User 2 | 1 | 1 | 2 | 5 | 4 |
| User 3 | 2 | 5 | 1 | 2 | 5 |
| User 4 | 5 | 4 | 4 | 1 | ? |

# Compute Average Rating

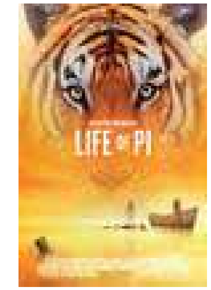




| | | | | | |
|---|---|---|---|---|---|
| User 1 | 5 | 4 | 3 | 1 | 3 |
| User 2 | 1 | 1 | 2 | 5 | 4 |
| User 3 | 2 | 5 | 1 | 2 | 5 |
| User 4 | 5 | 4 | 4 | 1 | ? |

3.25 3.5 2.5 2.25 4

# Subtract Average Rating

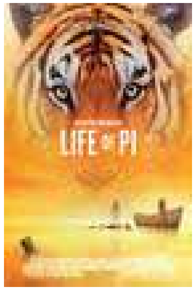





| | | | | | |
|---|---|---|---|---|---|
| User 1 | 1.75 | 0.5 | 0.5 | -1.25 | -1 |
| User 2 | -2.25 | -2.5 | -0.5 | 2.75 | 0 |
| User 3 | -1.25 | 1.5 | -1.5 | -0.25 | 1 |
| User 4 | 1.75 | 0.5 | 1.5 | -1.25 | ? |

# Computing Similarity

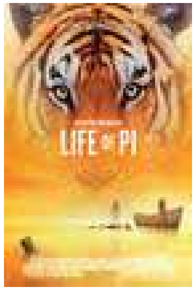





| | | | | | |
|---|---|---|---|---|---|
| User 1 | 1.75 | 0.5 | 0.5 | -1.25 | -1 |
| User 2 | -2.25 | -2.5 | -0.5 | 2.75 | 0 |
| User 3 | -1.25 | 1.5 | -1.5 | -0.25 | 1 |
| User 4 | 1.75 | 0.5 | 1.5 | -1.25 | ? |

Compute a similarity score between two users: The higher the score is, the more likely they enjoy the same movies

# Pearson Correlation Measure

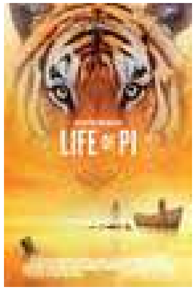

| | | | | | |
|---|---|---|---|---|---|
| User 1 | 1.75 | 0.5 | 0.5 | -1.25 | -1 |
| User 4 | 1.75 | 0.5 | 1.5 | -1.25 | ? |

$$\frac{1.75 \times 1.75 + 0.5 \times 0.5 + 0.5 \times 1.5 + (-1.25) \times (-.1.25)}{\sqrt{1.75^2 + 0.5^2 + 0.5^2 + (-1.25)^2}\sqrt{1.75^2 + 0.5^2 + 1.5^2 + (-1.25)^2}}$$

=0.9305

# Computing New Rating






| | | | | | |
|---|---|---|---|---|---|
| User 1 | 1.75 | 0.5 | 0.5 | -1.25 | -1 |
| User 2 | -2.25 | -2.5 | -0.5 | 2.75 | 0 |
| User 3 | -1.25 | 1.5 | -1.5 | -0.25 | 1 |
| User 4 | 1.75 | 0.5 | 1.5 | -1.25 | ? |

User 1 & User 4: 0.9305  User 2 & User 4: -0.7904
User 3 & User 4: -0.4382

?=4+0.9305*(-1)+(-0.7904)*0+(-0.4382)*1=2.6313

# Investigating the Reasons behind “Likes” and “Dislikes”

- **Based on genres**
  - Bob likes comedy movies
  - “Dumb and Dumber” is a comedy movie, so Bob should like it
- **More factors**
  - Genre, cast, amount of actions, orientation to children, female or male
  - Depth of character development
  - ……

# If We Scored All Factors for All the Movies ……

| | Orientation to female | Amount of action | Character development | Happy ending | … |
|---|---|---|---|---|---|
| Movie 1 | 5 | 2 | 2 | 2 | … |
| Movie 2 | 1 | 4 | 1 | 5 | … |
| … | … | … | … | … | … |

| | Orientation to female | Amount of action | Character development | Happy ending | … |
|---|---|---|---|---|---|
| Mary | 5 | 3 | 2 | 1 | … |

# Example

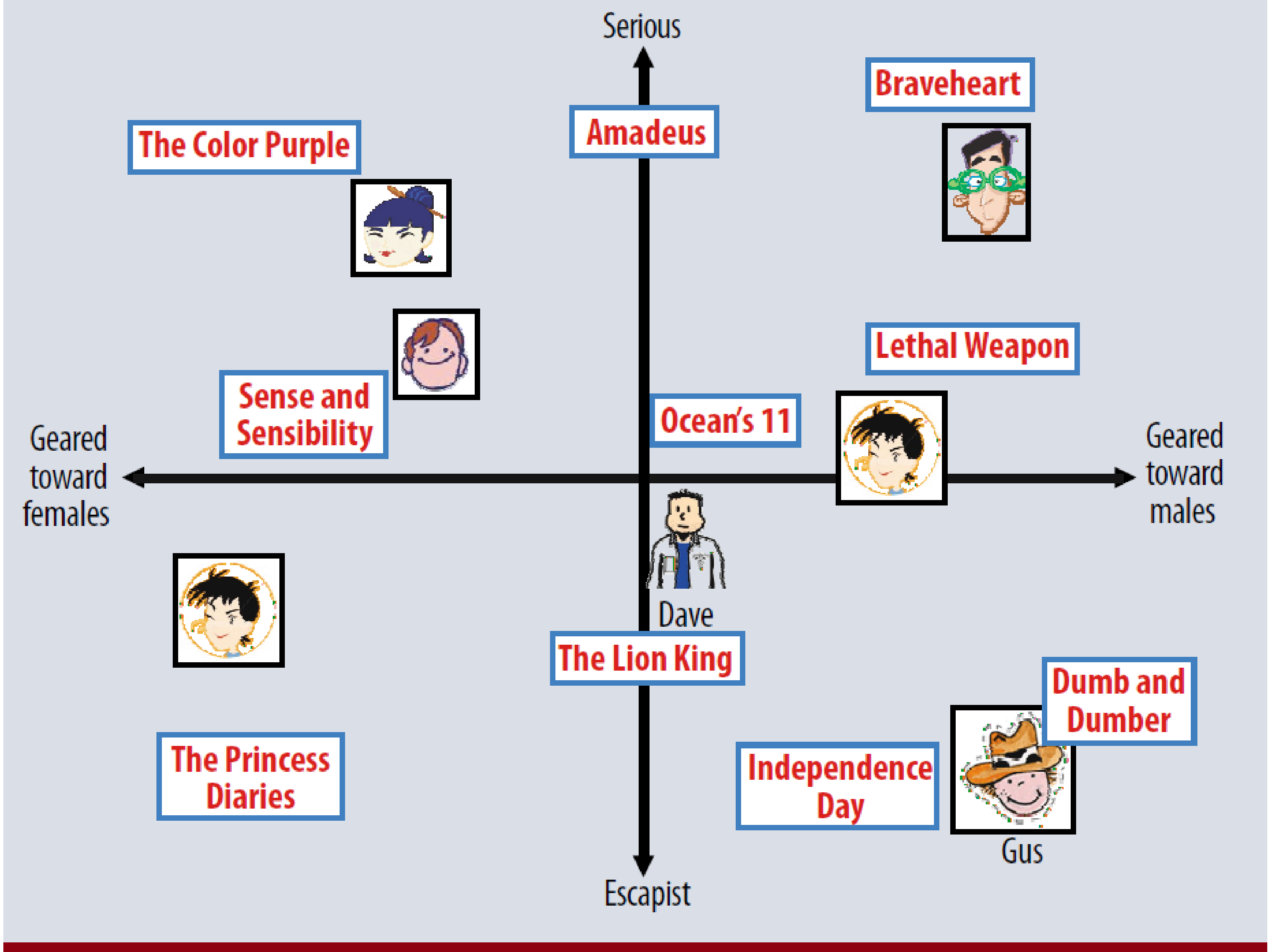



**Figure 2.** A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

# What We Have—One Score Per User Per Movie

items

users

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 4 | | | |
| | 3 | 5 | | | 5 |
| | | 4 | 5 | | 5 |
| | | 3 | | | |
| | | 3 | | | |
| 2 | | | 2 | | 2 |
| | | | | 5 | |
| | 2 | 1 | | | 1 |
| | 3 | | | 3 | |
| 1 | | | | | |

# Modeling Systematic Biases

$$r_{ui} \approx m + b_u + b_i + q^t_i \; p_u$$

| | | | |
|---|---|---|---|
| overall average rating | average rating bias for user u | average rating bias for movie i | user-movie interaction bias |

**Example:**

Mean rating m = 3.7

You are a critical reviewer: your ratings are 1 lower than the average -> $b_u = -1$

Star Wars gets an average rating of 0.5 higher than average movie: $b_i = +0.5$

Based on various factors, your taste tend to prefer Star Wars with a degree $q^t_i \; p_u = +0.2$

Predicted rating for you on Star Wars = 3.7 - 1 + 0.5 + 0.2 = 3.4

# Infer Movie and User Scores from Known Ratings

$$\min_{q,p} \; S_{(u,i)\, \epsilon R} (\, r_{ui} \;-\; (m \;+\; b_u \;+\; b_i + \; q^t_i \;\; p_u)\,)^2$$

sum over all the ratings

true rating

estimated rating

find the values for p and q which minimize the overall error

find the scoring patterns that minimize the difference between the true ratings and the predicted ratings!

# Other Useful Techniques

- **Time factor**
  - A user's taste or rating scale may change over time
  - A movie's popularity may change over time
- **User or movie profiles**
  - Age, gender of users; genre, cast, plots of movies;
- **Ensemble techniques**
  - Design many recommender systems and take their votes
  - Highly effective in real practice

# Netflix

- Movie rentals by DVD (mail) and online (streaming)
- 100k movies, 10 million customers
- Ships 1.9 million disks to customers each day
  - 50 warehouses in the US
  - Complex logistics problem
- Moving towards online delivery of content
- Significant interaction of customers with Web site

# The $1 Million Question

# Training Data

100 million ratings

Rating = [user, movie-id, time-stamp, rating value]

Generated by users between Oct 1998 and Dec 2005

Users randomly chosen among set with at least 20 ratings

# Ratings Data

17,700 movies

480,000 users

| | | | | | |
|---|---|---|---|---|---|
| 1 | 3 | 4 | | | |
| | 3 | 5 | | | 5 |
| | | 4 | 5 | | 5 |
| | | 3 | | | |
| | | 3 | | | |
| 2 | | | ? | | ? |
| | | | | ? | |
| | 2 | 1 | | | ? |
| | 3 | | | ? | |
| 1 | | | | | |

Test Data Set
(most recent ratings)

# Structure of Competition

- Register to enter at Netflix site
- Download training data of 100 million ratings
  - 480k users x 17.7k movies
  - Anonymized
- Submit predictions for 3 million ratings in “test set”
  - True ratings are known only to Netflix
- Can submit multiple times (limit of once/day)
- Prize
  - $1 million dollars if error is 10% lower than Netflix current system
  - Annual progress prize of $50,000 to leading team each year

# Evaluation Comparison

1.054 - just predict the average user rating for each movie

0.953 - Netflix's own system (Cinematch) as of 2006

0.941 - nearest-neighbor method using correlation

0.857 - required 10% reduction to win $1 million

# Million Dollars Awarded Sept 21st 2009

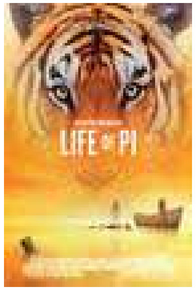

40,000 teams from over 150 countries

# Other Rating Applications

- **Modeled as user-item pairs**
  - User-books, user-CDs, user-hotels, user-products, user-friends, ……
- **Consider unique characteristics of each application**

# Implicit “Preference”






| | | | | | |
|---|---|---|---|---|---|
| User 1 | 1 | 1 | 1 | 0 | 0 |
| User 2 | 0 | 0 | 0 | 1 | 0 |
| User 3 | 0 | 1 | 0 | 0 | 0 |
| User 4 | 0 | 1 | 1 | 0 | 0 |

User check the movie webpage: 1

User did not check the movie webpage: 0

# Implicitly Show Interests

- **All of our online activities are recorded:**
  - Which tweets we read
  - Which product description we browsed
  - Which advertisement we clicked
  - Which videos we viewed
  - ......
- **Recommendation is given based on our activities**
  - Preference may not be that strong compared with explicit ratings

# Question

- **Youtube can gather the following types of implicit feedback**
  - Which videos each user viewed
  - Which videos each user liked
  - Which videos each user shared
  - Which videos each user commented
- **How to use all the information to recommend videos to users?**