COEN 169

Recommendation Systems I

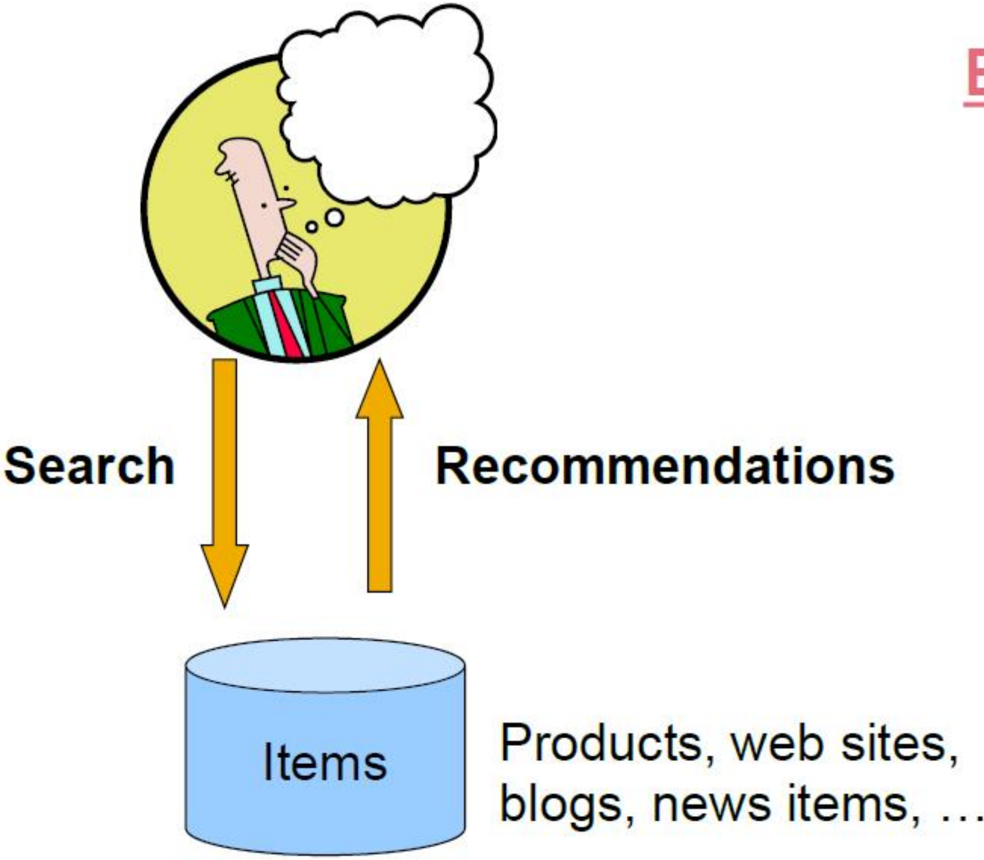
Yi Fang

Department of Computer Engineering

Santa Clara University

Information Overload

Recommendation vs Search



Examples:

amazon.com
PANDORA
StumbleUpon
del.icio.us
NETFLIX
movielens
helping you find the *right* movies
last.fm
the social music revolution
Google News
YouTube
XBOX LIVE

Types of Recommendations

- Editorial

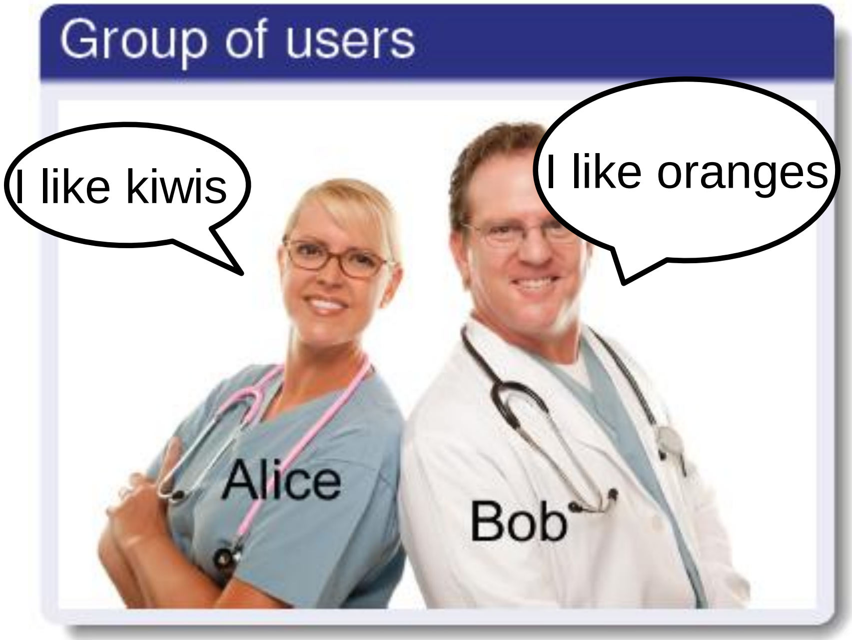
- Simple aggregates:
  - Top 10, Most Popular, Recent Uploads

- Tailored to individual users
  - Amazon, Netflix, …

Recommendation Algorithms

- Collaborative filtering
- Content-based recommendation
- Hybrid methods

What is Collaborative Filtering?



- Observe some user-item preferences
- Predict new preferences

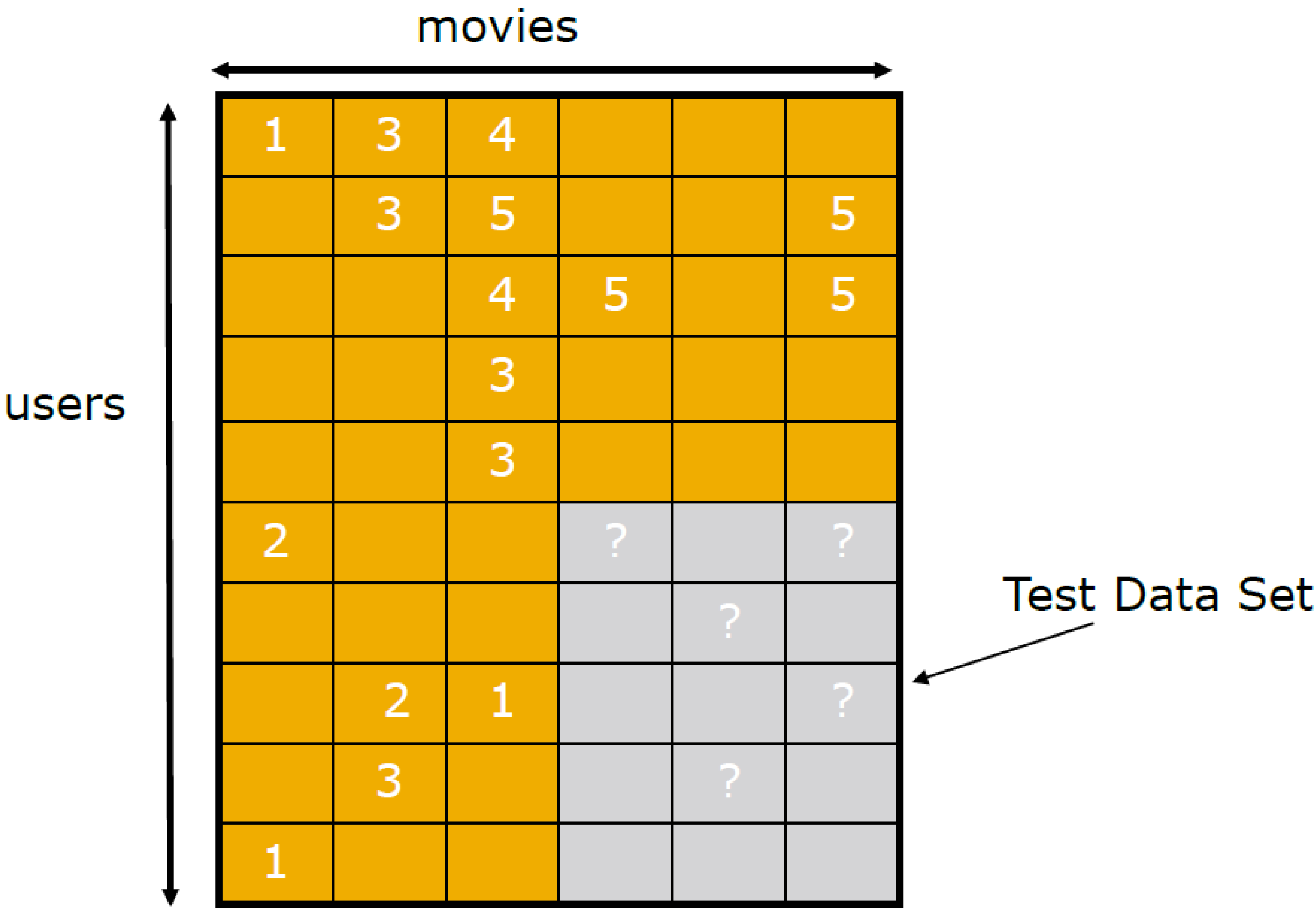
Does Bob like strawberries???

Data and Task

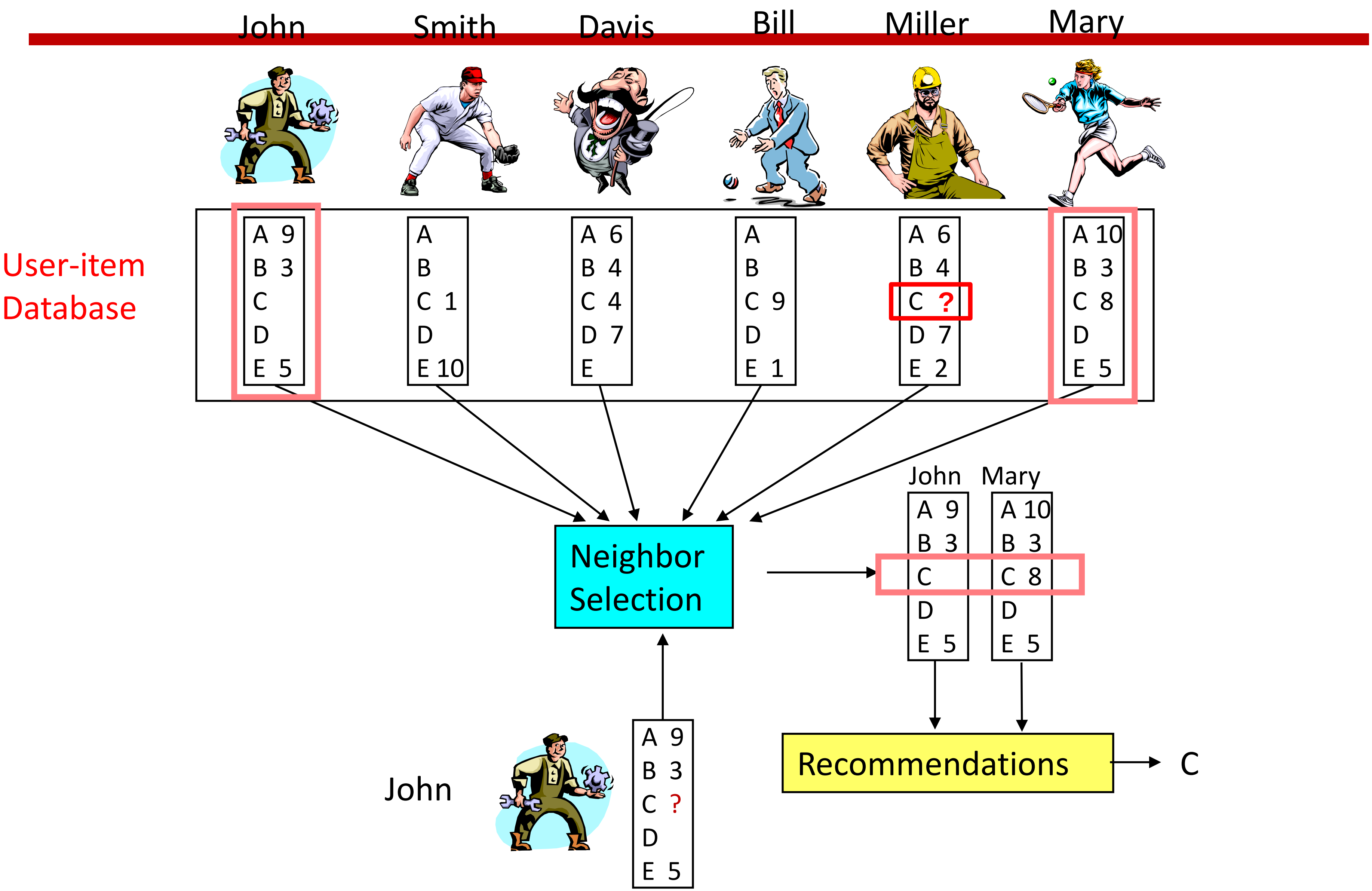
- Set U={u_1, …, u_m} of *m* users
- Set I={i_1, …, i_n} of *n* items (e.g. Movies, books)
- Set R={$r_{u,i}$ } of ratings/preference (e.g., 1-5, 1-10, binary)

- Task:
  - Recommend new items for an active user *a*
  - Usually formulated as a rating prediction problem

Data



User-based Collaborative Filtering



User-based Collaborative Filtering

- Consider the active user *a*
- Find *k* other users whose ratings are "similar" to *a*'s ratings
- Estimate *a*'s ratings based on ratings of the *k* similar users
- Called *k*-nearest neighborhood method

Neighbor Selection

- How similar are the users?

John

A	9
B	3
C	
D	
E	5

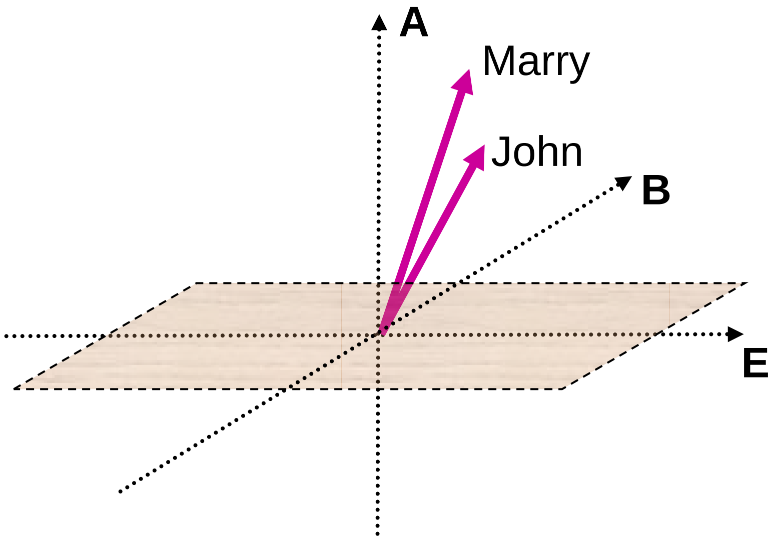
Mary

A	10
B	3
C	8
D	
E	5

$$similarity(John, Marry) = \frac{9\times10+3\times3+5\times5}{\sqrt{9^2+3^2+5^2}\times\sqrt{10^2+3^2+5^2}} = 0.9989$$

- Cosine Vector Similarity

$$similarity(u_1,u_2) = \cos\vartheta_{u_1,u_2} = \frac{\sum_{i=1}^{n} r_{u_1,i}\times r_{u_2,i}}{\sqrt{\sum_{i=1}^{n} r_{u_1,i}^2}\times\sqrt{\sum_{i=1}^{n} r_{u_2,i}^2}}$$



Rating Prediction (Cosine Similarity)

- For a given active user, *a*, select the most similar *k* users, *u*, based on similarity weights, $w_{a,u}$
- Predict a rating, $p_{a,i}$, for each item *i* and active user *a* by

similar user u's importance

Weighted average of similar users' ratings

similar user u's rating on the item

$$p_{a,i} = \boxed{\frac{\sum_{u=1}^{k} w_{a,u} r_{u,i}}{\sum_{u=1}^{k} w_{a,u}}} = \sum_{u=1}^{k} \boxed{\frac{w_{a,u}}{\sum_{u=1}^{k} w_{a,u}}} \boxed{r_{u,i}}$$

Centering Your Data

- Some users give systematically higher/lower ratings
- User's average rating is treated as her neutral attitude
- **Deviation** from the average reflect whether she is positive or negative about the item
- So we want to model the deviation

Rating Prediction (Pearson Correlation)

- To account for users different ratings levels (e.g., some users tend to give higher ratings), base predictions on *differences* from a user's *average* rating

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{k} w_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{k} |w_{a,u}|}$$

where $$w_{a,u} = \frac{\sum_{i=1}^{t} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{t} (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{i=1}^{t} (r_{u,i} - \bar{r}_u)^2}}$$

Pearson Correlation

- A measure of how correlated two variables are (how well their values fit on a straight line)
  - It is a value between 1 and –1
    - 1: the variables are perfectly correlated
    - 0: the variables are not correlated
    - –1: the variables are perfectly inversely correlated

Exercise

- Predict User D's rating on Item 4

	$Item1$	$Item2$	$Item3$	$Item4$	$Item5$
User A	4	4	1	4	3
User B	2	1	4	2	5
User C	3	1	3	2	1
User D	5	4	2		3

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{k} w_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{k} |w_{a,u}|}$$

$$where \qquad w_{a,u} = \frac{\sum_{i=1}^{t} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{t} (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{i=1}^{t} (r_{u,i} - \bar{r}_u)^2}}$$

Computing the average

- r_a is the average of the active user's ratings, in other words, D's ratings, which is (5+4+2+3)/4 = 3.5
- r_A is the average of user A's ratings which is (4+4+1+3)/4 = 3
- r_B is the average of user B's ratings which is (2+1+4+5)/4 = 3
- r_C is the average of user C's ratings which is (3+1+3+1)/4 = 2

Pearson Correlation

$$w_{A,D} = \frac{[(5-3.5)\cdot(4-3)] + [(4-3.5)\cdot(4-3)] + [(2-3.5)\cdot(1-3)] + [(3-3.5)\cdot(3-3)]}{\sqrt{(5-3.5)^{2} + (4-3.5)^{2} + (2-3.5)^{2} + (3-3.5)^{2}} \cdot \sqrt{(4-3)^{2} \cdot (4-3)^{2} \cdot (1-3)^{2} \cdot (3-3)^{2}}} =$$

$$w_{A,D} = \frac{(1.5\cdot 1) + (0.5\cdot 1) + (-1.5\cdot -2) + (-0.5\cdot 0)}{\sqrt{(1.5)^{2} + (0.5)^{2} + (-1.5)^{2} + (-0.5)^{2}} \cdot \sqrt{(1)^{2} \cdot (1)^{2} \cdot (-2)^{2} \cdot (0)^{2}}} =$$

$$w_{A,D} = \frac{(1.5) + (0.5) + (3) + (0)}{\sqrt{2.25 + 0.25 + 2.25 + 0.25} \cdot \sqrt{1 + 1 + 4 + 0}} =$$

$$w_{A,D} = \frac{5}{\sqrt{5} \cdot \sqrt{6}} = 0.9$$

Computing Pearson Correlation

- Similarly,

$$w_{B,D} = \frac{-5}{\sqrt{5} \cdot \sqrt{10}} = -0.7 \text{ and } w_{C,D} = \frac{0}{\sqrt{5} \cdot \sqrt{4}} = 0$$

- User D's rating on Item 4 is then

$$p_{A,item4} = 3.5 + \frac{[(4-3) \cdot 0.9] + [(2-3) \cdot (-0.7)] + [(2-2) \cdot 0]}{0.9 + 0.7 + 0} =$$

$$p_{A,item4} = 3.5 + \frac{(1 \cdot 0.9) + [(-1) \cdot (-0.7)] + [0 \cdot 0]}{0.9 + 0.7 + 0} = 3.5 + \frac{1.6}{1.6} = 4.5$$

Evaluating predictions

- Root-mean-square error (RMSE)

$$\sqrt{\frac{1}{S}\sum_{(u,i)\in test}(p_{u,i}-r_{u,i})^2}$$

$p_{u,i}$ is the predicted rating of user *u* for item *i*

$r_{u,i}$ is the true rating of user *u* for item *i*

$(u,i)\in test$ denotes the missing ratings in the test set

S is the total number of missing ratings to be predicted movies

users

1	3	4			
	3	5			5
		4	5		5
		3			
		3			
2			?		?
				?	
	2	1			?
	3			?	
1					

Test Data Set

movies

users

1	3	4			
	3	5			5
		4	5		5
		3			
		3			
2			2		2
				5	
	2	1			1
	3			3	
1					

Universally liked movies

IMDb

Find Movies, TV shows, Celebrities and more... | All

Movies | TV | News | Trailers | Community | IMDbPro | Ap

The Dark Knight (2008)

88

PG-13 152 min - Action | Crime | Drama - 18 July 2008 (USA)

9.0

Your rating: ★★★★★★★★★★ -/10

Ratings: **9.0**/10 from 832,786 users   Metascore: 82/100

Reviews: 3,567 user | 581 critic | 39 from Metacritic.com

When Batman, Gordon and Harvey Dent launch an assault on the mob, they let the clown out of the box, the Joker, bent on turning Gotham on itself and bringing any heroes down to his level.

Director: Christopher Nolan

Writers: Jonathan Nolan (screenplay), Christopher Nolan (screenplay), and 3 more credits »

Stars: Christian Bale, Heath Ledger and Aaron Eckhart | See full cast and crew

Less common movies

Improving Predictions I

- Universally liked movies are not as useful in capturing similarity as less common movies
- How can we penalize universally liked movies?
- Use an analogy to IDF: Inverse User Frequency (*IUF*)

$$IUF(j) = \log \frac{m}{m_j}$$

- m_j is the number of users that have rated item *j*
- m is the total number of users
- Multiply the original ratings by *IUF*

Improving Predictions II

- Case amplification refers to a transform applied to the weights used in the basic collaborative filtering prediction. The transform emphasizes high weights and punishes low weights

$$w'_{a,u} = w_{a,u} \cdot \left| w_{a,u} \right|^{\rho - 1}$$

- where ρ is the case amplification power, $\rho \geq 1$, and a typical choice of ρ is 2.5
- It tends to favor high weights as small values raised to a power become negligible
- E.g., $0.9^{2.5} \approx 0.8, 0.5^{2.5} \approx 0.177, 0.1^{2.5} \approx 0.003$