

# Collaborative Filtering

Harnessing quality judgments  
of other users or items

Professor Wei-Min Shen

University of Southern California

Thanks for source slides and material to:

J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets  
<http://www.mmmds.org>

# Three Approaches to Recommendation Systems

- **1) Content-based (consider single user)**
  - ❑ Use characteristics of an item
  - ❑ Recommend items that
    - have similar content to items that the user liked in the past
    - match pre-defined attributes of the user
- **2) Collaborative filtering (consider other users)**
  - ❑ Build a model from
    - a user's past behavior (e.g., items previously purchased or rated), and
    - similar decisions made by other users
  - ❑ Use the model to predict items that the user may like
  - ❑ Collaborative: suggestions made to a user utilizing information across the entire user base
- **3) Hybrid approaches**

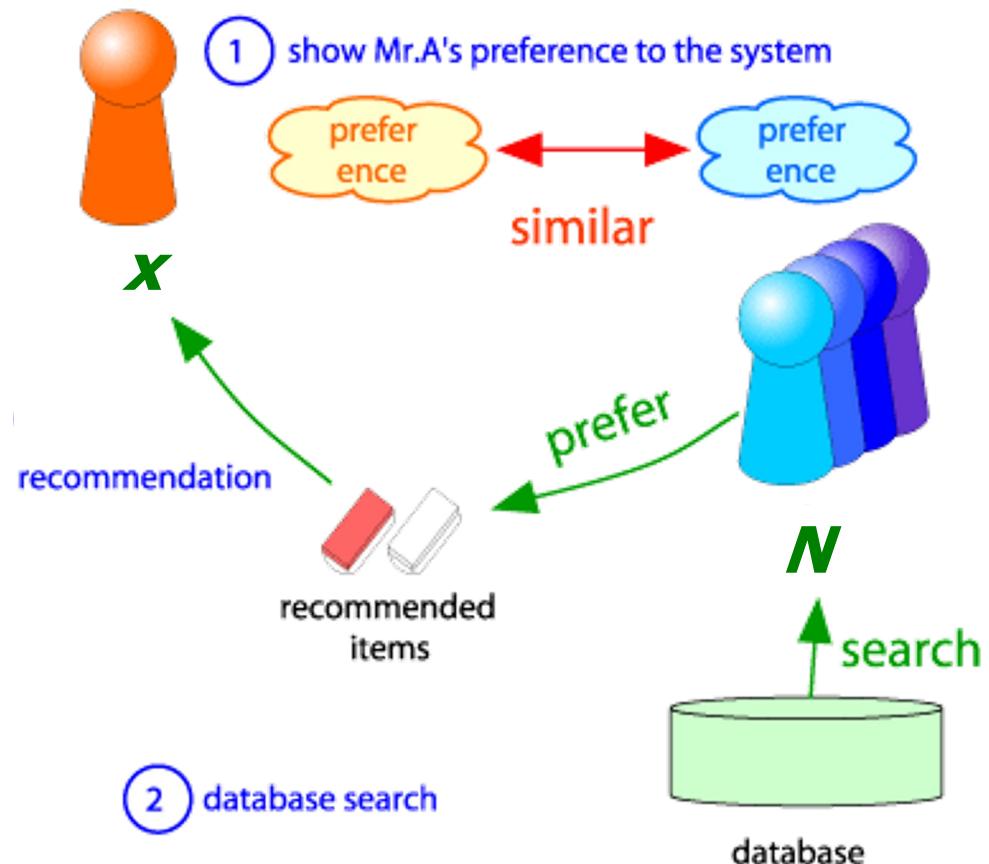
# Collaborative Filtering Example

- User-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  $N$



# Collaborative Filtering: Overview

- CF works by **collecting user feedback**: e.g., **ratings for items**
  - ❑ Exploit **similarities** in rating behavior among **users** in determining recommendations
- **Two classes of CF algorithms:**
  1. **Neighborhood-based or Memory-based approaches**
    - ❑ User-based CF
    - ❑ Item-based CF
  2. **Model-based approaches**
    - ❑ Estimate parameters of statistical models for user ratings
    - ❑ Latent factor and matrix factorization models

**NEIGHBORHOOD-BASED OR  
MEMORY-BASED COLLABORATIVE  
FILTERING**

Identify User's Neighbors => Recommendation

## **USER-BASED CF**

# Neighborhood-based / Memory-based Collaborative Filtering

- Active user: the user we want to make predictions for
- **User-based CF:** A subset of other users (**neighborhood**) is chosen based on their similarity to the active user
- A weighted combination of the ratings of neighbors is used to make predictions for the active user
- **Steps:**
  1. Assign a weight to all users w.r.t. **similarity with the active user**
  2. **Select k neighbor users that have the highest similarity with active user (the neighborhood)**
  3. **Compute a prediction from a weighted combination of the selected neighbors' ratings**

# Similarity between users: by what measure?

- Weight  $w_{x,y}$  is measure of similarity between user  $x$  and active user  $y$

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

Possible similarity metrics:

- Jaccard similarity

- (Intersection / union) of two sets
- Doesn't use non-boolean values: e.g., ratings

$r_x, r_y$  as sets:

$$r_x = \{1, 4, 5\}$$

$$r_y = \{1, 3, 4\}$$

- Cosine similarity

- Treat ratings as vectors to points
- Cosine similarity between points

$r_x, r_y$  as points:

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

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

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Similarity between users:  
by what measure?

## **JACCARD SIMILARITY**

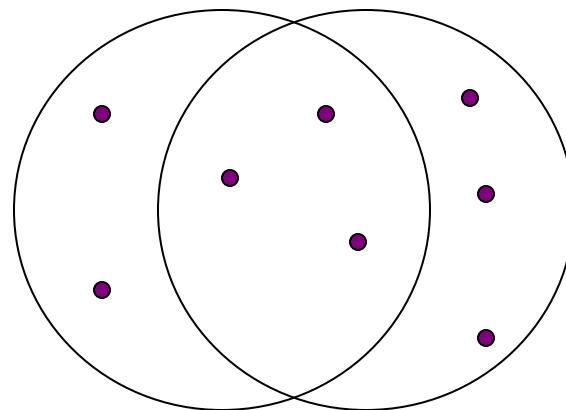
# Jaccard Similarity and Distance of Sets

- The *Jaccard similarity* of two sets is the size of their intersection divided by the size of their union

$$Sim(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$$

- *Jaccard distance* =  $1 - \text{Jaccard Similarity}$

# Example: Jaccard Similarity and Distance



3 in intersection  
8 in union

**Jaccard similarity = 3/8**

- **Jaccard distance** =  $1 - \text{Jaccard Similarity}$  or  
5/8 in this example

# How well do these similarity metrics work?

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

- **Intuitively we want:**  $\text{sim}(A, B) > \text{sim}(A, C)$  (**why?**)
  - ❑ For A and B: One movie rated in common with similar ratings
  - ❑ For A and C: Two movies rated in common but with dissimilar ratings
- **Jaccard similarity (Example 9.7)**
  - ❑ Ignores values (rates) in matrix
  - ❑ Only look at which items are rated in matrix
- Intersection / union:  $\text{sim}(A,B) = 1/5$ ,  $\text{sim}(A,C) = 2/4$
- $1/5 < 2/4$  indicates  $\text{sim}(A,B) < \text{sim}(A,C)$
- **Not a good similarity metric for ratings data!**

Similarity between users:  
by what measure?

## **COSINE SIMILARITY**

## Same matrix with cosine similarity (Example 9.8)

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

- Treat blanks as 0 value: **questionable, since it treats no rating as more similar to disliking a movie than liking it**
- Cosine similarity of A and B is:

$$\frac{4 \times 5}{\sqrt{4^2 + 5^2 + 1^2} \sqrt{5^2 + 5^2 + 4^2}} = 0.380$$

- Cosine similarity of A and C is: 0.322
- $0.380 > 0.322$ : **Indicates A,B slightly more similar than A,C**
- **in this case, cosine similarity better than Jaccard similarity**

# Normalizing ratings (Example 9.9)

- To deal better with non-rated items
- Subtract the average rating of that user from each rating
  - low ratings -> negative numbers; high ratings -> positive numbers

	HP1	HP2	HP3	TW	SW1	SW2	SW3	Avg. Rating
A	4			5	1			10/3
B	5	5	4					14/3
C				2	4	5		11/3
D		3					3	6/2

	HP1	HP2	HP3	TW	SW1	SW2	SW3	(V – Avg)
A	2/3			5/3	-7/3			
B	1/3	1/3	-2/3					
C				-5/3	1/3	4/3		
D		0					0	

Cosine sim A,B vs. A,C: 0.092 > -0.559 (85 degrees vs 124 degrees)

A, C are much further apart than A, B, but neither is close

Similarity between users:  
by what measure?

## **PEARSON CORRELATION**

# Most commonly used measure of similarity: Pearson Correlation Coefficient

- Pearson correlation measures extent to which two variables **linearly related**
- For users  $u, v$ : Pearson correlation is

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}}$$

where the  $i \in I$  summations are over the items that both the users  $u$  and  $v$  have rated and  $\bar{r}_u$  is the average rating of the co-rated items of the  $u$ th user.

- And  $r_{u,i}$  is rating of item  $i$  by user  $u$
- Note: When calculating these similarities, average only over the co-rated items // so be careful

# Example: Pearson Correlation Coefficient

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}}$$

	$I_1$	$I_2$	$I_3$	$I_4$		
$U_1$	4	?	5	5	14/3	
$U_2$	4	2	1			
$U_3$	3		2	4		
$U_4$	4	4				
$U_5$	2	1	3	5	10/3	11/4

- Want correlation of user 1, user 5:  $w_{1,5}$       Set  $I$  of co-rated movies =  $\{I_1, I_3, I_4\}$
- For user 1: average rating on co-rated movies is 14/3; For user 5: 10/3
- Numerator:  $(4 - 14/3)(2 - 10/3) + (5 - 14/3)(3 - 10/3) + (5 - 14/3)(5 - 10/3)$   
 $= 12/9 = 1.3333$
- Denominator:  $\sqrt{(4 - 14/3)^2 + (5 - 14/3)^2 + (5 - 14/3)^2}$   
 $* \sqrt{(2 - 10/3)^2 + (3 - 10/3)^2 + (5 - 10/3)^2}$   
 $= 1.76383$
- Pearson correlation  $w_{1,5} = 1.3333 / 1.76383 = 0.756$

## Example: Pearson Correlation Coefficient

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}}$$

	$I_1$	$I_2$	$I_3$	$I_4$	
$U_1$	4	?	5	5	?
$U_2$	4	2	1		
$U_3$	3		2	4	
$U_4$	4	4			?
$U_5$	2	1	3	5	

- Want correlation of user 1, user 4:  $w_{1,4}$       Set  $I$  of co-rated movies =  $\{?, ?\}$
- For user 1: **average rating on co-rated movies** is ?; For user 4 is ?
- Numerator:
- Denominator:
- Pearson correlation  $w_{1,4} = 0$**

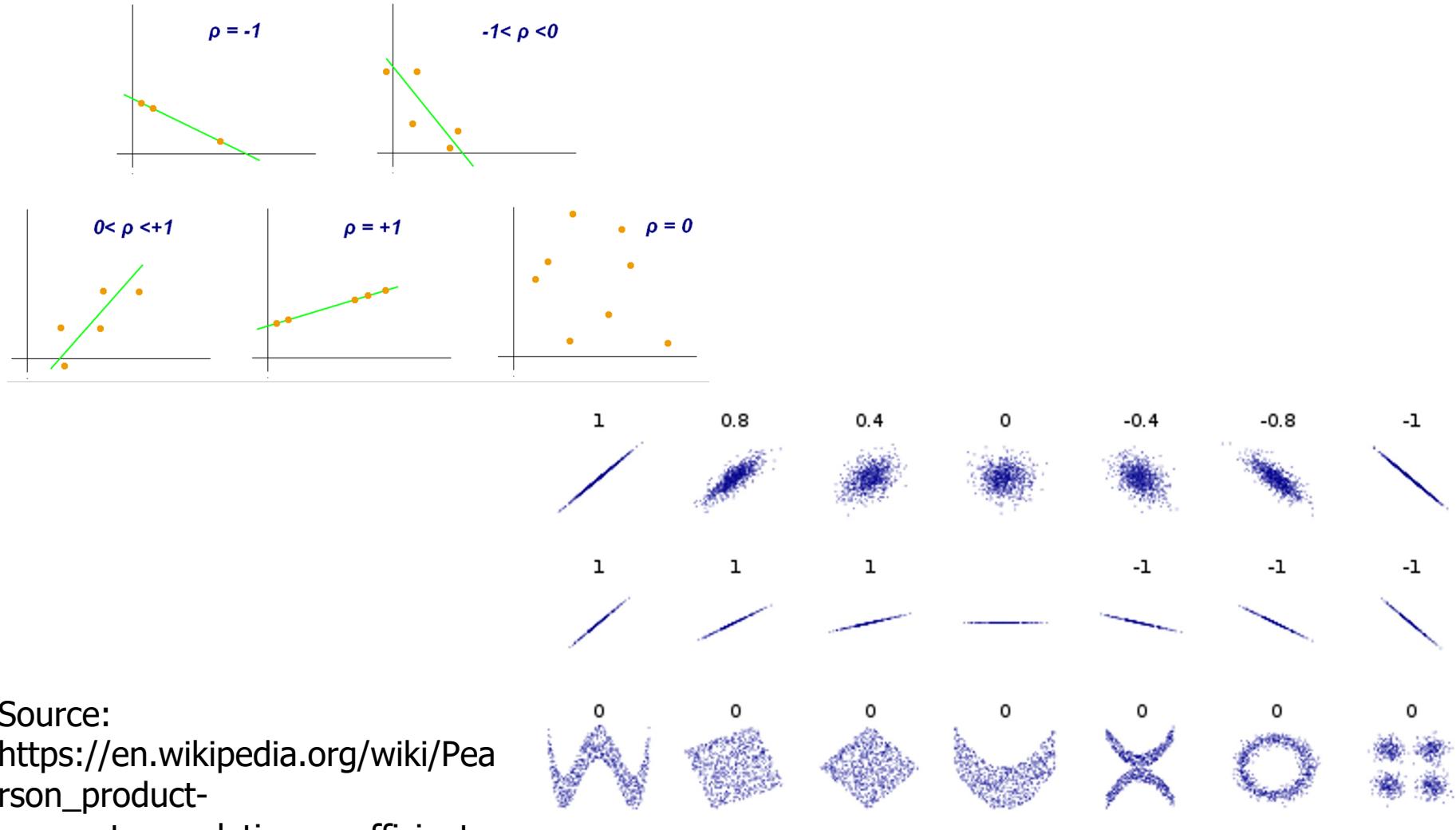
## Example: Pearson Correlation Coefficient

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}}$$

	$I_1$	$I_2$	$I_3$	$I_4$	
$U_1$	4	?	5	5	9/2
$U_2$	4	2	1		5/2
$U_3$	3		2	4	
$U_4$	4	4			
$U_5$	2	1	3	5	

- Want correlation of user 1, user 2:  $w_{1,2}$       Set  $I$  of co-rated movies =  $\{I_1, I_3\}$
- For user 1: **average rating on co-rated movies** is 9/2; For user 2: 5/2
- Numerator:  $(4-4.5)(4-2.5) + (5-4.5)(1-2.5) = -0.5(1.5) + 0.5(-1.5) = -1.5$
- Denominator:  $\text{sqrt}((4-4.5)^2 + (5-4.5)^2) * \text{sqrt}((4-2.5)^2 + (1-2.5)^2)$   
 $= \text{sqrt}(0.5) * \text{sqrt}(1.5^2 + 1.5^2) = \text{sqrt}(0.5) * \text{sqrt}(4.5) = 1.5$
- Pearson correlation  $w_{1,2} = -1.5 / 1.5 = -1$**

# Pearson Correlation Examples



Source:  
[https://en.wikipedia.org/wiki/Pearson\\_product-moment\\_correlation\\_coefficient](https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient)

# Making User-based CF Predictions with Pearson: Weighted Sum of Other Users' Ratings

- Weighted average of their ratings is used to generate predictions
- To make a prediction for an active user  $a$  on an item  $i$ :

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot w_{a,u}}{\sum_{u \in U} |w_{a,u}|}$$

Diagram illustrating the components of the prediction formula:

- Your own opinion:  $\bar{r}_a$  (blue box)
- other people's opinion:  $r_{u,i}$  (blue box)
- How much you like u:  $w_{a,u}$  (blue box)

where  $\bar{r}_a$  and  $\bar{r}_u$  are the average ratings for the user  $a$  and user  $u$  on all other rated items, and  $w_{a,u}$  is the weight between the user  $a$  and user  $u$ . The summations are over all the users  $u \in U$  who have rated the item  $i$ . F

- Note: When making predictions, calculate average of ALL rated items for users  $a$  and  $u$
- Summation is over all users who rated item  $i$

# Continued Example: User-Based CF Prediction with Pearson Correlation Coefficient

	$I_1$	$I_2$	$I_3$	$I_4$	
$U_1$	4	?	5	5	"average other than $I_2$ "
$U_2$	4	2	1		$(4+1)/2$
$U_3$	3		2	4	
$U_4$	4	4			$(4)/1$
$U_5$	2	1	3	5	$(2+3+5)/3$

- Want to predict rating for user  $U_1$  on item  $I_2$ : users  $U_2$ ,  $U_4$  and  $U_5$ , who rated  $I_2$
- Similarity of  $U_1$  to these users:  $w_{1,5} = 0.756$ ,  $w_{1,4} = 0$ ,  $w_{1,2} = -1$

$$P_{1,2} = \bar{r}_1 + \frac{\sum_u (r_{u,2} - \bar{r}_u) \cdot w_{1,u}}{\sum_u |w_{1,u}|} \quad 14/3 = 4.67$$

$$= \bar{r}_1 + \frac{(r_{2,2} - \bar{r}_2)w_{1,2} + (r_{4,2} - \bar{r}_4)w_{1,4} + (r_{5,2} - \bar{r}_5)w_{1,5}}{|w_{1,2}| + |w_{1,4}| + |w_{1,5}|} \quad 5/2 = 2.5$$

$$= 4.67 + \frac{(2 - 2.5)(-1) + (4 - 4)0 + (1 - 3.33)0.756}{1 + 0 + 0.756} \quad 4/1 = 4$$

$$= 3.95. \quad 10/3 = 3.33$$

# Neighborhood-Based algorithms

- In neighborhood-based CF algorithms, **a subset of nearest neighbors** of the active user are **chosen based on their similarity with active user**
- Use these users for predictions rather than all users who have rated the item

Item's Neighbors ↗ Recommendations

## ITEM-BASED CF

# Item-based Collaborative Filtering

- Neighborhood-based CF algorithms do not scale well when applied to millions of users & items
  - ❑ Due to computational complexity of search for similar users (possible solutions?)
- **Item-to-item collaborative filtering**
  - ❑ Rather than matching similar users
  - ❑ **Match user's rated items to similar items**
- In practice, often **leads to faster online systems and better recommendations**
- **Similarities between pairs of items  $i$  and  $j$  are computed off-line**
- Predict rating of user " $a$ " on item " $i$ " with a simple weighted average

## Pearson Correlation between items $i, j$

For the item-based algorithm, denote the set of users  $u \in U$  who rated both items  $i$  and  $j$ , then the *Pearson Correlation* will be

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}, \quad (2)$$

where  $r_{u,i}$  is the rating of user  $u$  on item  $i$ ,  $\bar{r}_i$  is the average rating of the  $i^{\text{th}}$  item by those users, see Figure 2 [40].

- **Note: Sum over set of users  $U$  who rated both items  $i, j$**
- $r_{u,i}$  is rating of user  $u$  on item  $i$
- $\bar{r}_i$  is average rating of  $i^{\text{th}}$  item by those users

# Pearson Correlation between items $i, j$ (Cont'd)

		Items								
		1	2	...	$i$	$j$	...	$m - 1$	$m$	
Users	1				$R$	?				
	2				$R$	$R$				
:										
	$l$				$R$	$R$				
	:									
	$n - 1$				?		$R$			
	$n$				$R$	$R$				

FIGURE 2: item-based similarity ( $w_{i,j}$ ) calculation based on the co-rated items  $i$  and  $j$  from users  $2, l$  and  $n$ .

Source: Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 4.

# Make Item-Based Predictions Using a Simple Weighted Average

- In order to predict rating for user  $u$  on item  $i$ , do
- $w_{i,n}$  is the (similarity) weight between item  $i$  and item  $n$
- $r_{u,n}$  is rating for user  $u$  on item  $n$
- Summation over **neighborhood set  $N$  of items** rated by  $u$  that are most similar to  $i$

$$P_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} |w_{i,n}|}$$

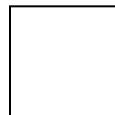
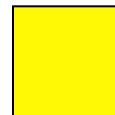
Diagram illustrating the components of the weighted average formula:

- $r_{u,n}$ : User's opinion on the item  $n$  (blue box)
- $w_{i,n}$ : How similar is  $n$  to  $i$  (blue box)

where the summations are over all other rated items  $n \in N$  for user  $u$ ,  $w_{i,n}$  is the weight between items  $i$  and  $n$ ,  $r_{u,n}$  is the rating for user  $u$  on item  $n$ .

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

		users											
		12	11	10	9	8	7	6	5	4	3	2	1
movies (items)	4			5			5			3		1	1
	3	1	2			4			4	5			2
		5	3	4		3		2	1		4	2	3
		2			4			5		4	2		4
	5	2					2	4	3	4			5
		4			2			3		3		1	6

 - unknown rating     - rating between 1 to 5

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

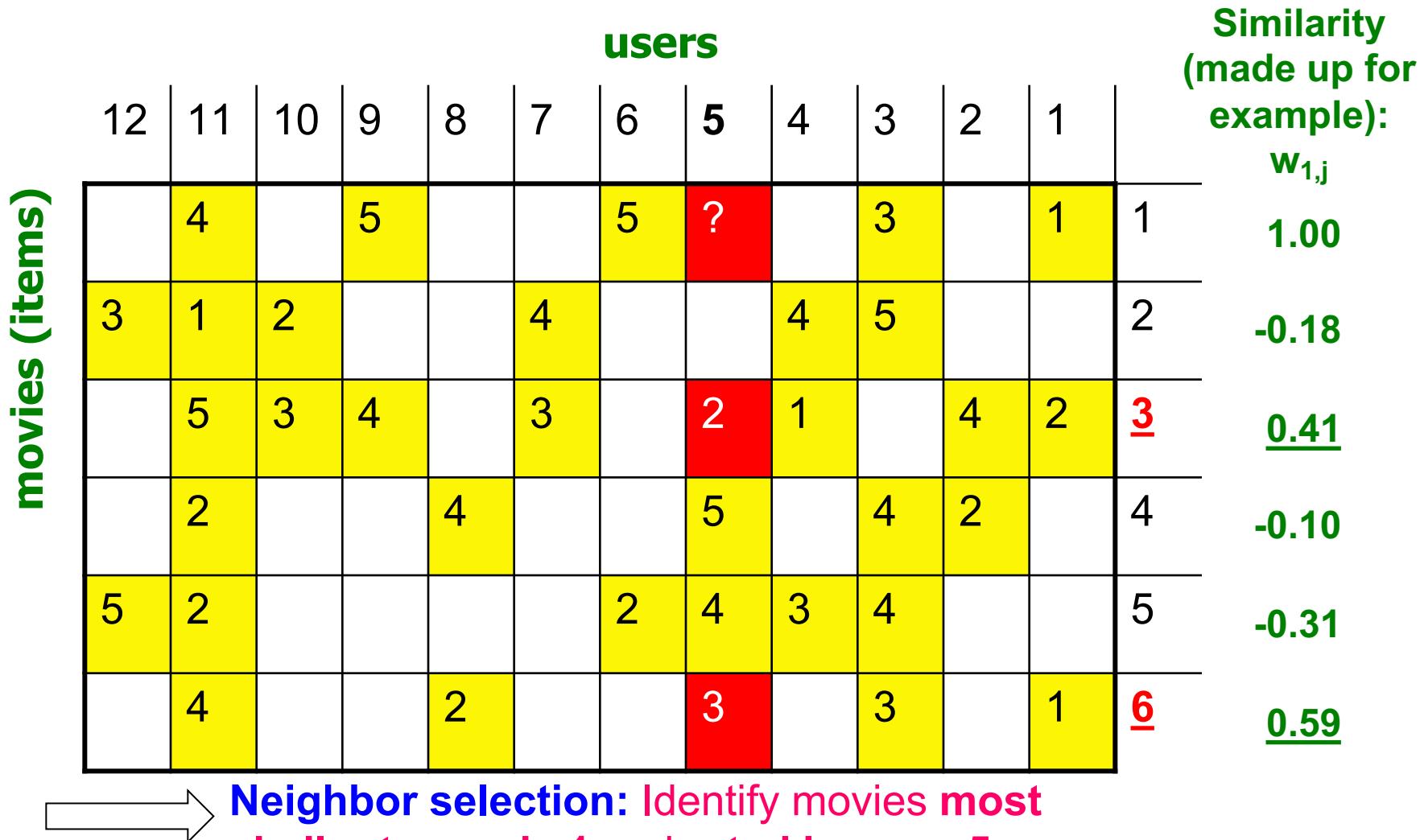
		users												
		12	11	10	9	8	7	6	5	4	3	2	1	
movies (items)	12	4			5			5	?		3		1	1
	3	1	2			4				4	5			2
		5	3	4		3			2	1		4	2	3
		2			4				5		4	2		4
	5	2						2	4	3	4			5
		4			2				3		3		1	6



- estimate rating of movie **1** by user **5**

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

First: what is similarity between items?



Neighborhood size is 2: pick movies 3 and 6

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

	users												Similarity (made up): $w_{1,j}$
movies/items	12	11	10	9	8	7	6	5	4	3	2	1	
	4			5			5	?		3		1	1
3	1	2			4				4	5			2
	5	3	4		3			2	1		4	2	3
	2			4			5			4	2		4
5	2					2	4	3	4				5
	4			2				3		3		1	6

Similarity weights between items:  
 $w_{1,3}=0.41, w_{1,6}=0.59$

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

	users												
movies	12	11	10	9	8	7	6	5	4	3	2	1	
	4			5			5	2.6		3		1	1
	3	1	2			4			4	5			2
	5	3	4		3			2	1		4	2	3
	2			4			5		4	2			4
	5	2					2	4	3	4			5
	4			2				3		3		1	6

Similarity:  
 $w_{1,j}$

1.00

-0.18

0.41

-0.10

-0.31

0.59

Predict by taking weighted average:

$$P_{1.5} = (0.41 \cdot 2 + 0.59 \cdot 3) / (0.41 + 0.59) = 2.6$$

$$P_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} |w_{i,n}|}$$

# Example: Item-to-Item Collaborative Filtering with Pearson Similarity

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

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}$$

- What does set  $U$  represent?
  - ❑ Set of users  $U$  who rated both items  $i, j$
- What are the members of the set  $U$  for  $w_{1,2}$ ?
  - ❑  $U1$  and  $U4$  rated items  $I1$  and  $I2$
- When calculating average ratings for item  $i$ , which ratings do we use?  
All ratings or just for co-rated items?
  - ❑ We will show examples of co-rated items
  - ❑ We can use all ratings as well: based on all ratings:  
 $\text{avg}(r_1) = (2+3+5)/3$ ,  $\text{avg}(r_2) = (1+4+3)/3$

# Example: Item-to-Item Collaborative Filtering with Pearson Similarity

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

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}$$

- Based on all ratings: average ratings for items I1, I2:  $\bar{r}_1 = 10/3, \bar{r}_2 = 8/3$
- Pearson correlation for  $w_{1,2}$ : *Similarity between items 1 and 2*
- Set  $U$  includes  $U_1$  and  $U_4$
- $w_{1,2} = ((r_{U1,I1} - 10/3)(r_{U1,I2} - 8/3) + (r_{U4,I1} - 10/3)(r_{U4,I2} - 8/3)) / (\sqrt{(r_{U1,I1} - 10/3)^2 + (r_{U4,I1} - 10/3)^2} * \sqrt{(r_{U1,I2} - 8/3)^2 + (r_{U4,I2} - 8/3)^2})$
- $w_{1,2} = [(2-10/3)(1-8/3) + (5-10/3)(3-8/3)] / [\sqrt{(2-10/3)^2 + (5-10/3)^2} * \sqrt{(1-8/3)^2 + (3-8/3)^2}] = 2.778/3.628 = 0.765$

# Example: Item-to-Item Collaborative Filtering with Pearson Similarity

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

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}$$

co-rated items only

- Based on co-ratings: average ratings for items I1, I2:  $\bar{r}_1 = 7/2$ ,  $\bar{r}_2 = 4/2$
- Pearson correlation for  $w_{1,2}$ : Similarity between items 1 and 2
- Set U includes U1 and U4
- $w_{1,2} = ((r_{U1,I1} - 7/2)(r_{U1,I2} - 4/2) + (r_{U4,I1} - 7/2)(r_{U4,I2} - 4/2)) /$   
 $(\sqrt{(r_{U1,I1} - 7/2)^2 + (r_{U4,I1} - 7/2)^2} * \sqrt{(r_{U1,I2} - 4/2)^2 + (r_{U4,I2} - 4/2)^2})$
- $w_{1,2} = [(2-7/2)(1-4/2) + (5-7/2)(3-4/2)] /$   
 $[\sqrt{(2-7/2)^2 + (5-7/2)^2} * \sqrt{(1-4/2)^2 + (3-4/2)^2}]$   
 $=3/3=1$

# Item-Based Prediction

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

$$P_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} |w_{i,n}|}$$

- If we have the following item similarities:  
 $w_{2,1} = 0.5, w_{2,3} = 0.2, w_{2,4} = 0.3$
- Which items are in the neighborhood  $N$  for item 2 if  $|N| = 2$ ?
  - ❑ Items I1 and I4
- Predict the rating of user U2 on I2: user = 2, item = 2,  $|N| = \text{items } 1, 4$

$$\begin{aligned} P_{2,2} &= [(r_{2,1} * w_{2,1}) + (r_{2,4} * w_{2,4})] / [0.5 + 0.3] \\ &= [3 * 0.5 + 2 * 0.3] / 0.8 = 2.625 \end{aligned}$$

# **EXTENSIONS TO MEMORY-BASED ALGORITHMS**

# Extensions to Memory-Based Algorithms

- A variety of approaches/extensions have been studied to improve the performance of CF predictions
- Typically involve **modifying the similarity weights** or the **ratings** used in predictions or **guessing missing ratings**
- **User-based CF: (User's Neighbors => recommend)**

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \quad | \quad P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot w_{a,u}}{\sum_{u \in U} |w_{a,u}|}$$

- **Item-Based CF: (Item's Neighbors => recommend)**

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}} \quad | \quad P_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} |w_{i,n}|}$$

# Extensions to Memory-Based Algorithms: Default Voting

- In many collaborative filters, **pairwise similarity is computed only from the ratings in the intersection of the items both users have rated (“co-rated items”)**
  - ❑ Not reliable when there are too few votes to generate similarity values ( $U$  is small)
  - ❑ Focusing on co-rated items (“intersection set similarity”) also neglects global rating behavior reflected in a user’s entire rating history
- Assuming some default voting values for the missing ratings: **can improve CF prediction performance**

# Extensions to Memory-Based Algorithms: Default Voting (cont.)

Approaches to default voting values:

- Herlocker et al. accounts for small intersection sets (small number of co-rated items) by **reducing the weight of users that have fewer than 50 items in common**

$$P_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} |w_{i,n}|}$$

- Chee et al. **use average of the clique (small group of co-rated items) as a default voting** to extend a user's rating history
- Breese et al. **use a neutral or somewhat negative preference for the unobserved ratings** and then computes similarity between users on the resulting ratings data.

# Extensions to Memory-Based Algorithms: Inverse User Frequency

- Universally liked items are not as useful in capturing similarity as less common items
- Inverse frequency
  - ❑  $f_j = \log(n/n_j)$
  - ❑  $n_j$  is number of users who have rated item  $j$
  - ❑  $n$  is total number of users
- If everyone has rated item  $j$ , then  $f_j$  is zero
  - ❑ (Note: looks a lot like Inverse Document Frequency (IDF))
- Approach: transform the ratings
  - ❑ For vector similarity-based CF: new rating = original rating multiplied by  $f_j$

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot w_{a,u}}{\sum_{u \in U} |w_{a,u}|}$$

- ❑ For very popular items, ratings  $r_{u,i}$  will be greatly reduced
- ❑ Less popular items will have greater effect on prediction

# Extensions to Memory-Based Algorithms: Case Amplification

- Transform applied to weights used in CF prediction
- Emphasizes high weights and punishes low weights

$$w'_{i,j} = w_{i,j} \cdot |w_{i,j}|^{\rho-1}, \quad (8)$$

where  $\rho$  is the *case amplification* power,  $\rho \geq 1$ , and a typical choice of  $\rho$  is 2.5 [65].

- Reduces noise in the data
- Favors high weights
- Small values raised to a power become negligible
- Example:
  - For  $w_{i,j} = 0.9$ , weight remains high ( $0.9^{2.5} \approx 0.8$ )
  - For  $w_{i,j} = 0.1$ , weight becomes negligible ( $0.1^{2.5} \approx 0.003$ )

# Extensions to Memory-Based Algorithms: Imputation-Boosted CF

- When the rating data for CF tasks are **extremely sparse**: hard to produce accurate predictions using the Pearson correlation-based CF
- Su et al. proposed imputation-boosted collaborative filtering (IBCF)
- First uses an imputation technique to fill in missing data
- Then use traditional Pearson correlation-based CF algorithm on this completed data to predict a user rating for a specified item
  - ❑ Example imputation techniques: mean imputation, linear regression imputation, predictive mean matching imputation, Bayesian multiple imputation, and machine learning classifiers (including naive Bayes, SVM, neural network, decision tree, lazy Bayesian rules)

# Extensions to Memory-Based Algorithms: Imputation-Boosted CF (Cont'd)

## Simple mean imputation

The data on the left below has one missing observation on variable 2, unit 10.

We replace this with the arithmetic average of the observed data *for that variable*. This value is shown in red in the table below.

Unit	Variables	
	1	2
1	3.4	5.67
2	3.9	4.81
3	2.6	4.93
4	1.9	6.21
5	2.2	6.83
6	3.3	5.61
7	1.7	5.45
8	2.4	4.94
9	2.8	5.73
10	3.6	5.58

## Imputation Example

- This approach is clearly inappropriate for categorical variables.
- It does not lead to proper estimates of measures of association or regression coefficients. Rather, associations tend to be diluted.
- In addition, variances will be wrongly estimated (typically under estimated) if the imputed values are treated as real. Thus inferences will be wrong too.

Source:

[http://missingdata.lshtm.ac.uk/index.php?option=com\\_content&view=article&id=68:simple-mean-imputation&catid=39:simple-ad-hoc-methods-for-coping-with-missing-data&Itemid=96](http://missingdata.lshtm.ac.uk/index.php?option=com_content&view=article&id=68:simple-mean-imputation&catid=39:simple-ad-hoc-methods-for-coping-with-missing-data&Itemid=96)