

What is Collaborative Filtering?

- Community of users
- To predict a user's opinion, use the opinions of others
- Advantages:
 - No need to analyse (index) content
 - Can capture more subtle things
 - Serendipity

Types of Collaborative Filtering

- User-based collaborative filtering
- Item-based collaborative filtering




User-based Collaborative Filtering

- People who agreed in the past are likely to agree again
- To predict a user's opinion for an item, use the opinion of similar users
- Similarity between users is decided by looking at their overlap in opinions for other items

Ex: User-based Collaborative Filtering



	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	5
User 6 	8	3	8	3	7

Similarity between users

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 4 	7	1	7	3	8

- How similar are users 1 and 2?
- How similar are users 1 and 5?
- How do you calculate similarity?

Similarity between users: simple way

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5

- Only consider items both users have rated
- For each item:
 - Calculate difference in the users' ratings
 - Take the average of this difference over the items

$$\text{Sim}(\text{User1}, \text{User2}) = \frac{\sum_j | \text{rating}(\text{User1}, \text{Item } j) - \text{rating}(\text{User2}, \text{Item } j) |}{\text{Num. of items}}$$

Problems: Similarity between users

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	1	2	3	4	5
User 2	5	4	3	2	1

$$\text{Sim}(\text{User1}, \text{User2}) = 12/5 = 2.4$$

	Item 1	Item 2	Item 3	Item 4	Item 5
User 3	1	2	3	4	5
User 4	4	5	6	7	8

$$\text{Sim}(\text{User3}, \text{User4}) = 15/5 = 3$$

Other Solutions(1): Jaccard Coefficient

① Option 1 :- Use Jaccard co-eff to compute the similarity.

$$\text{Jaccard co-efficient} = \frac{(r_A \cap r_B)}{(r_A \cup r_B)}$$

→ [no. of movies which are rated by both A & B]

→ [Total no. of movies rated by A & B]

Other Solutions(2): Cosine Similarity

② option 2 :- cosine similarity between the users (if the rating is not given, take them zero)

$$\begin{aligned} \text{Sim}(A, B) &= \cos(A, B) = \frac{\vec{r}_A \cdot \vec{r}_B}{|\vec{r}_A| |\vec{r}_B|} \quad \begin{array}{l} i^o = 1 \text{ to } n \\ \checkmark \\ \text{No. of items} \end{array} \\ &= \frac{\sum_{i^o=1}^N r_{A, i^o} r_{B, i^o}}{\left(\sqrt{\sum_{i^o=1}^N r_{A, i^o}^2} \right) \left(\sqrt{\sum_{i^o=1}^N r_{B, i^o}^2} \right)} \\ &= \frac{1 \times 1 + 0 + 0}{\sqrt{1+9} \cdot \sqrt{4+16}} = \frac{1}{\sqrt{10} \cdot \sqrt{20}} \end{aligned}$$

Other Solutions(3): Mean centred Cosine similarity

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



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

The cosine of the angle between A and C is

$$\frac{(5/3) \times (-5/3) + (-7/3) \times (1/3)}{\sqrt{(2/3)^2 + (5/3)^2 + (-7/3)^2} \sqrt{(-5/3)^2 + (1/3)^2 + (4/3)^2}} = -0.559$$

Better Solutions : Pearson Coefficient

- **Use Statistical Correlation Metrics**
 - These measure how well two data sets fit on a straight line
 - Corrects for grade inflation



Perfect Correlation for User3, User4



Inverse Correlation for User1, User2

Similarity to Recommendations

- Similarity provides a ranking for other users, or a weight to associate with each user
 - Identify Similar Users, and recommend what they have rated highly
 - To calculate rating of an item to recommend, give weight to each user's recommendations based on how similar they are to you.

User-based nearest-neighbor collaborative filtering (2)

■ Example

- A database of ratings of the current user, Alice, and some other users is given:

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

- Determine whether Alice will like or dislike *Item5*, which Alice has not yet rated or seen

User-based nearest-neighbor collaborative filtering (3)

■ Some first questions

- How do we measure similarity?
- How many neighbors should we consider?
- How do we generate a prediction from the neighbors' ratings?



	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Measuring user similarity (1)

- A popular similarity measure in user-based CF: Pearson correlation

a, b : users

$r_{a,p}$: rating of user a for item p

$P(a, b)$: set of items, rated both by a and b

– Possible similarity values between -1 and 1


Measuring user similarity (2)

- A popular similarity measure in user-based CF: Pearson correlation

a, b : users

$r_{a,p}$: rating of user a for item p

	Item1	Item2	Item3	Item4	Item5	b
Alice	5	3	4	4	?	
User1	3	1	2	3	3	sim = 0.85
User2	4	3	4	3	5	sim = 0.00
User3	3	3	1	5	4	sim = 0.70
User4	1	5	5	2	1	sim = -0.79



and 1

Making predictions

- A common prediction function:

$$pred(a, p) = \bar{r}_a + \frac{\sum_{b \in N} sim(a, b) * (r_{b,p} - \bar{r}_b)}{\sum_{b \in N} sim(a, b)}$$



- Calculate, whether the neighbors' ratings for the unseen item i are higher or lower than their average
- Combine the rating differences – use the similarity with a as a weight

Example: Similarity(1)

TABLE 4: A simple example of ratings matrix.

	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

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u) (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}}, \quad (1)$$

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. In an example in Table 4, we have $w_{1,5} = 0.756$.

Example: Similarity(2)

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

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 . For the simple example in Table 4, using the user-based CF algorithm, to predict the rating for U_1 on I_2 , we have

TABLE 4: A simple example of ratings matrix.

	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

$$\begin{aligned}
 P_{1,2} &= \bar{r}_1 + \frac{\sum_u (r_{u,2} - \bar{r}_u) \cdot w_{1,u}}{\sum_u |w_{1,u}|} \\
 &= \bar{r}_1 + \frac{(r_{2,2} - \bar{r}_2)w_{1,2} + (r_{4,2} - \bar{r}_4)w_{1,4} + (r_{5,2} - \bar{r}_5)w_{1,5}}{|w_{1,2}| + |w_{1,4}| + |w_{1,5}|} \\
 &= 4.67 + \frac{(2 - 2.5)(-1) + (4 - 4)0 + (1 - 3.33)0.756}{1 + 0 + 0.756} \\
 &= 3.95.
 \end{aligned}$$

Question

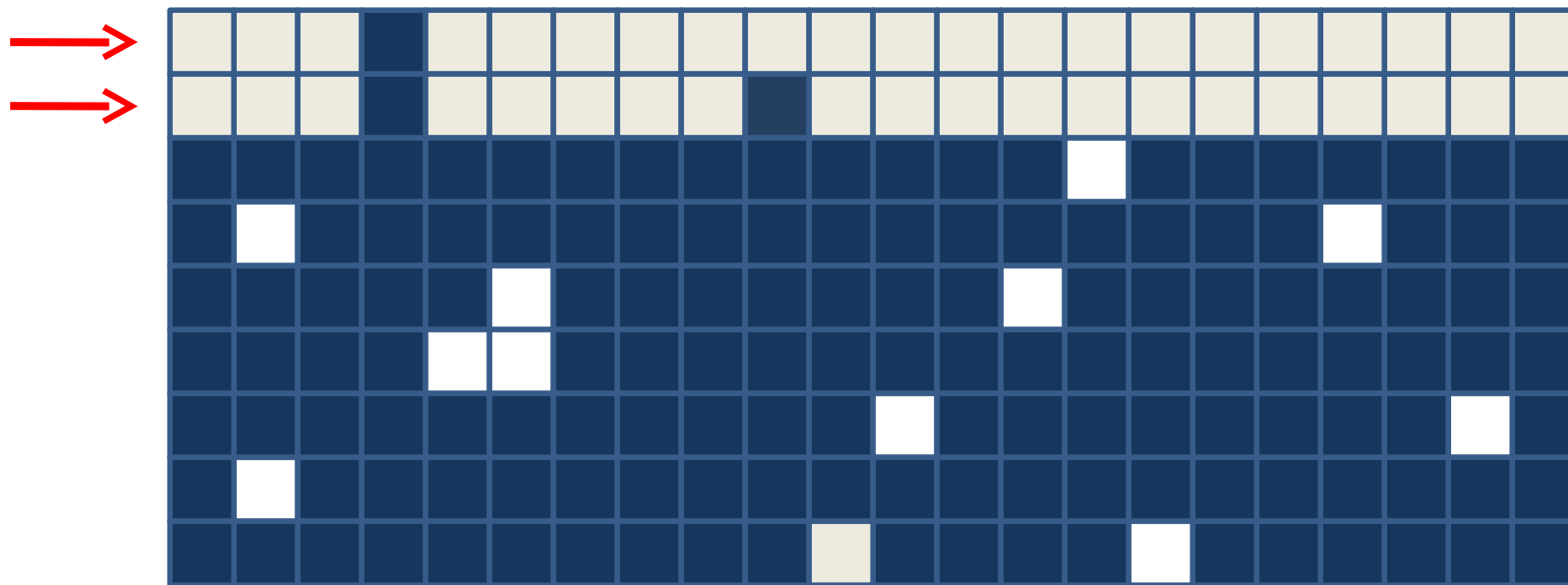
- Given a set of historical preference judgments for the community of users, predict the rating that would be assigned by Bob for item “Garmin” fitness band using User based Collaborative filtering algorithm.

	Fitbit Band	GOQi	Sony	Garmin
Jane	5	2	5	4
John	2	5	4	3
Bob	2	2	5	????
Joe	5	1	--	2

Problems with User-based Collaborative Filtering (1)

- **User Cold-Start problem**

not enough known about new user to decide who is similar (and perhaps no other users yet..)

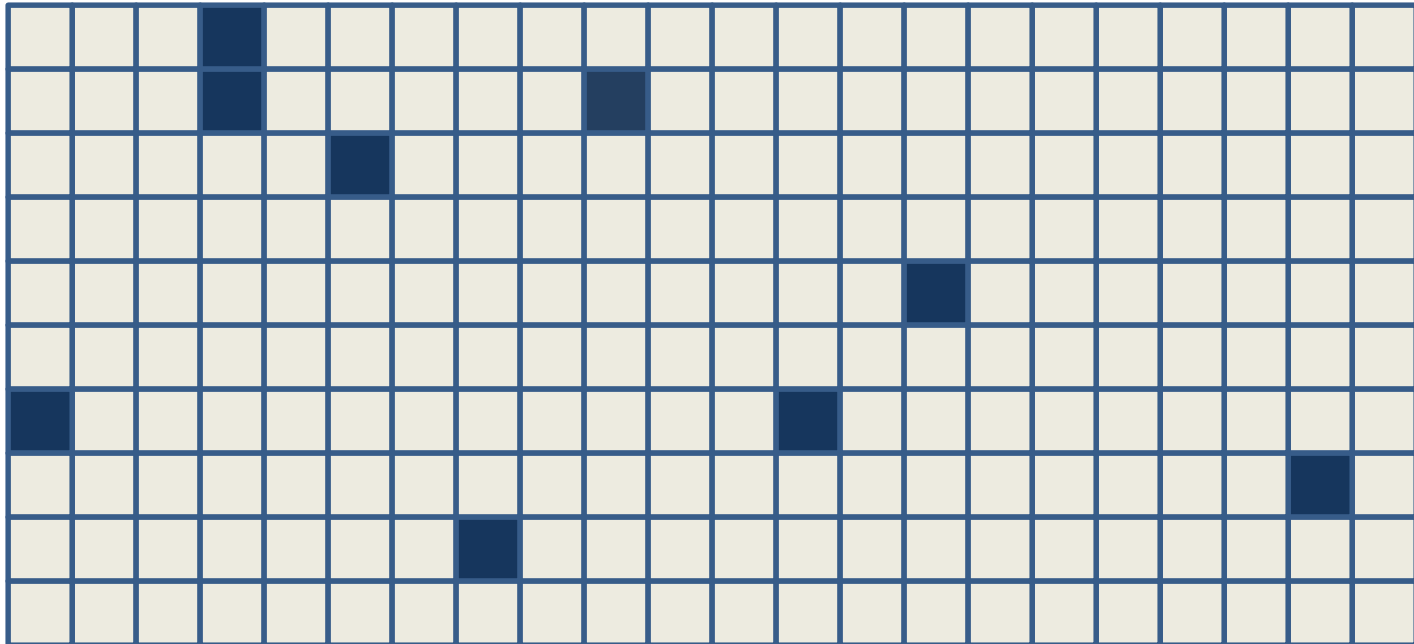


- Need way to motivate early rater

Problems with User-based Collaborative Filtering (2)

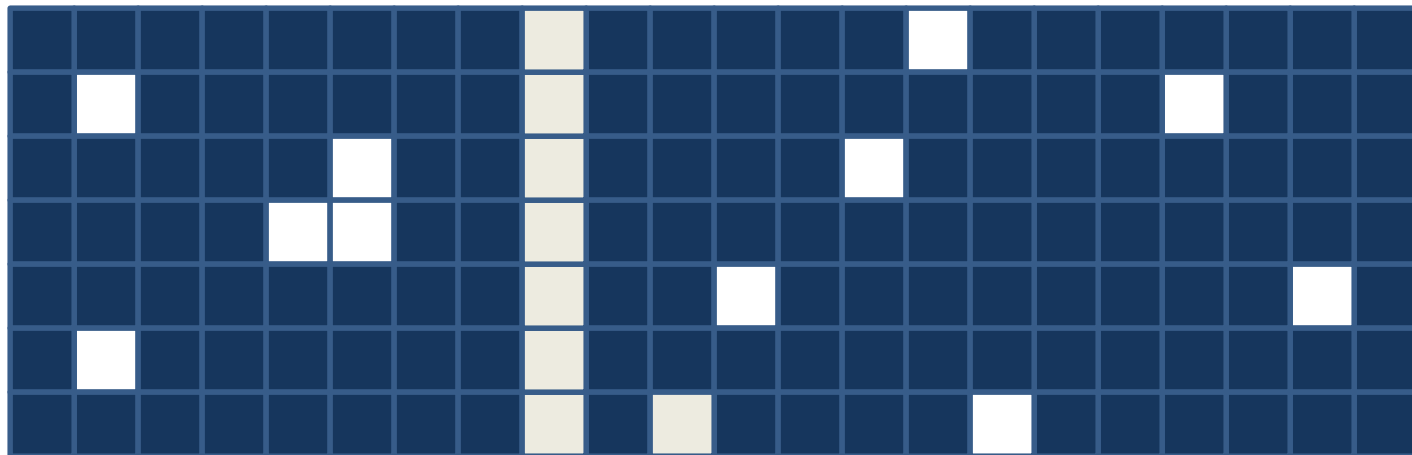
- **Sparsity**

when recommending from a large item set, users will have rated only some of the items
(makes it hard to find similar users)



Problems with User-based Collaborative Filtering (3)

- **Scalability**
 - with millions of ratings, computations become slow
- **Item Cold-Start problem**
 - Cannot predict ratings for new item till some similar users have rated it .



Item-based Collaborative Filtering

Item-item collaborative filtering







- Item-item collaborative filtering, or item-based, or item-to-item, is a form of collaborative filtering based on the similarity between items calculated using people's ratings of those items.
- Item-item collaborative filtering was invented and used by Amazon.com in 1998.[\[1\]](#)

Item-based Collaborative Filtering

- User is likely to have the same opinion for similar items
- Similarity between items is decided by looking at how other users have rated them
 - Star Wars = [Action, Sci-fi...]
 - Star Wars = [User1:8, User2:3, User3:7...]
- Advantage (compared to user-based CF):
 - Prevents User Cold-Start problem
 - Improves scalability (similarity between items is more stable than between users)

Example:

Item-based Collaborative Filtering

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	5
User 6 	8	3	8	3	7

Similarity between items

Item 3	Item 4	Item 5
?	2	7
5	7	5
7	4	7
7	3	8
4	6	5
8	3	7

- How similar are items 3 and 4?
- How similar are items 3 and 5?
- How do you calculate similarity?

Similarity between items: simple way

Item 3	Item 4
?	2
5	7
7	4
7	3
4	6
8	3

- Only consider users who have rated both items
- For each user:
 - Calculate difference in ratings for the two items
 - Take the average of this difference over the users

Sim(Item 3, Item 4) =

$$\frac{\sum_j |\text{rating}(\text{User } j, \text{Item 3}) - \text{rating}(\text{User } j, \text{Item 4})|}{\text{Number of Users}}$$

Item-based collaborative filtering

- **Basic idea:**

- Use the similarity between items (and not users) to make predictions

- **Example:**

- Look for items that are similar to Item5
- Take Alice's ratings for these items to predict the rating for Item5

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

In this section we study a class of item-based recommendation algorithms for producing predictions to users. Unlike the user-based collaborative filtering algorithm discussed in Section 2 the item-based approach looks into the set of items the target user has rated and computes how similar they are to the target item i and then selects k most similar items $\{i_1, i_2, \dots, i_k\}$. At the same time their corresponding similarities $\{s_{i1}, s_{i2}, \dots, s_{ik}\}$ are also computed. Once the most similar items are found, the prediction is then computed by taking a weighted average of the target user's ratings on these similar items. We describe these two aspects namely, the similarity computation and the prediction generation in details here.

	1	2	...	i		j	...	$m-1$	m
1				R		?			
2				R		R			
\vdots									
l				R		R			
\vdots									
$n-1$?		R			
n				R		R			

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 th item by those users

Making predictions

- A common prediction function:

$$pred(u, p) = \frac{\sum_{i \in ratedItem(u)} sim(i, p) * r_{u,i}}{\sum_{i \in ratedItem(u)} sim(i, p)}$$



- Neighborhood size is typically also limited to a specific size
- Not all neighbors are taken into account for the prediction
- An analysis of the MovieLens dataset indicates that "in most real-world situations, a neighborhood of 20 to 50 neighbors seems reasonable" (Herlocker et al. 2002)

Example

User/Item	Item_1	Item_2	Item_3
User_1	2	–	3
User_2	5	2	–
User_3	3	3	1
User_4	–	2	2

$$\text{Similarity}(I1, I2) = \frac{(5*2)+(3*3)}{\sqrt{5^2+3^2}\sqrt{2^2+3^2}} = 0.90$$

$$\text{Similarity}(I2, I3) = \frac{(3*1)+(2*2)}{\sqrt{3^2+2^2}\sqrt{1^2+2^2}} = 0.869$$

$$\text{Similarity}(I1, I3) = \frac{(2*3)+(3*1)}{\sqrt{2^2+3^2}\sqrt{3^2+1^2}} = 0.789$$

User/Item	Item_1	Item_2	Item_3
User_1	2	–	3
User_2	5	2	–
User_3	3	3	1
User_4	–	2	2

User/Item	Item_1	Item_2	Item_3
User_1	2	–	3
User_2	5	2	–
User_3	3	3	1
User_4	–	2	2

Rating of Item_2 for User_1

$$r(U_1, I_2) = \frac{r(U_1, I_1) * s_{I_1 I_2} + r(U_1, I_3) * s_{I_3 I_2}}{s_{I_1 I_2} + s_{I_3 I_2}} = \frac{(2 * 0.9) + (3 * 0.869)}{(0.9 + 0.869)} = 2.49$$

Rating of Item_3 for User_2

$$r(U_2, I_3) = \frac{r(U_2, I_1) * s_{I_1 I_3} + r(U_2, I_2) * s_{I_2 I_3}}{s_{I_1 I_3} + s_{I_2 I_3}} = \frac{(5 * 0.789) + (2 * 0.869)}{(0.789 + 0.869)} = 3.43$$

Rating of Item_1 for User_4

$$r(U_4, I_1) = \frac{r(U_4, I_2) * s_{I_1 I_2} + r(U_4, I_3) * s_{I_1 I_3}}{s_{I_1 I_2} + s_{I_1 I_3}} = \frac{(2 * 0.9) + (2 * 0.789)}{(0.9 + 0.789)} = 2.0$$

Q2: Identify the appropriate technique to develop the recommender system. The technique must consider the varying rating behaviour of users.

- **Apply this technique to compute the rating that would be assigned by the User U2 for the new and unseen video;**
- **Provide the similarity order of all users corresponding to user 2.**

User/ Video	V1	V2	V3	V4	V5
U1	2	3	4	3	4
U2	3	4	3	???	4
U3	4	5	2	4	1
U4	1	2	1	3	2
U5	5	1	5	2	???

Hybrid Recommender Systems

- Use a combination of Content-based and Collaborative Filtering
- Or a combination of User-based and Item-based Collaborative Filtering
- Why would you want to do this?

Disclaimer

- There is a LOT of work on recommender systems
- I have simplified things and left things out....