Association Rules

19

Recommending with Association Rules

- Example task
  - we want to be able to predict which pages the visitor is most interested at some point of the session
  - The goal is to:
    - provide recommendations
    - improve usability
    - improve sales/loyalty.
- Strategy
  - look for pages that tend to be accessed in the same sessions and look for sets of pages that predict other sets of pages
  - this is done using Association Rule discovery
  - The built model is a set of association rules
- Steps
  - 1) Training → 2) Deploying

20

Recall Association Rules:

- Rule

$$X \Rightarrow Y$$

- Support

Support($X \Rightarrow Y$) = $\frac{Number\ of\ transactions\ containing\ both\ X\ and\ Y}{Total\ number\ of\ transactions}$

- Confidence

Confidence($X \Rightarrow Y$) = $\frac{Support(X \Rightarrow Y)}{Support(X)}$

Recommending with Association Rules (cont.)

Modelling - *training*

- From the historic transactions, a DB is built from a set of rules involving items that
- Use a low **Support** ($\frac{10}{|BD|}$)
- Have a **Confidence**
  - > 50%: every recommendation is more likely to be relevant
  - < 50%: riskier recommendations
  - << 50%: no-recommendation situations
- You may use other filters for association rules
- A model is the resulting set of rules

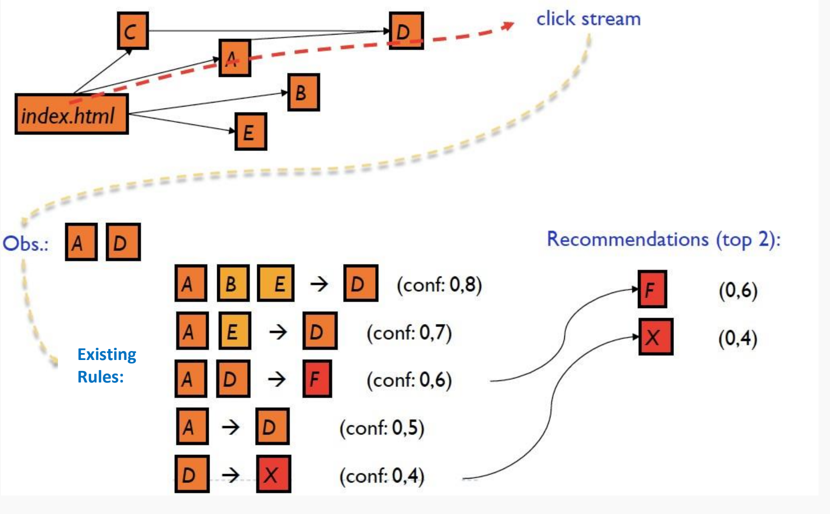
Recommending with Association Rules (cont.)

Recommending - *deploying*

For an active user "viewing" items `Obs`

1. **Look for rules** `A` → `C` such that
   - `A` is a subset of `Obs`
2. **Disregard rules** wich have `C` in the `Obs`
3. **Sort rules by confidence** (descending)
   - if they have the same confidence **prefer higher support**
   - if needed, prefer simpler rules
4. **Recommendation**
   - For a given N, recommend different **consequents of top N rules**

Recommending with Association Rules (example)



Recommending with Association Rules (cont.)

Applying Association Rules in Web Mining is useful for:

- Product / item automatic recommendation
    - cross-selling, up-selling
- Improve site navigation
    - recommending links
- Product bundling

Case Study: A news stories portal

- A Web portal for readers and journalists
- There is restricted access
    - Login is needed
- There are detailed access records
    - weblogs
- Data was collected for one year
- Business goals
    - increase frequency and length of visits
    - increase the number of articles read
    - create article bundles

Case Study: A news stories portal

- Pre-processed data

user	artigo
2	As-voltas-que-o-crédito-dá
2	O-valor-da-inovação-ou-vice-versa
2	Como-reconciliar-o-Marketing-e-as-Operações
2	Gestão-em-oito-lições
3	Chairman-e-CEO---um-cargo-ou-dois
4	A-guerra-pelo-talento
4	Steve-Ballmer-Um-computador-para-cada-membro-da-família
4	Universitários-trocam-cafés-por-portáteis
6	Novos-empresários-para-o-comércio
6	Retalhistas-com-vida-facilitada
6	F-C--Porto-lidera-transferências-na-pré-temporada
6	O-que-está-a-dar-no-retalho---Parte-I
6	Rotas-Úteis---Retalho
6	Leroy-Merlin-expande-se-para-sul-com-150-milhões-para-investir
6	aQuem-está-empregado-tem-muitos-direitosa
6	Grandes-superfícies-perdem--EUR-20-milhões
6	Leroy-Merlin-quer-investir-150-milhões-de-euros-até-2013
6	Modelo-Continente-com-vários-pedidos-de-licenciamento
7	IKEA-monta-casa-em-Portugal
9	Turismo-mundial-registou-a-maior-quebra-de-sempre
9	Rotas-Uteis
9	Rotas-Úteis---Marketing
10	Rotas-Uteis
10	138-projectos-aprovados-pelo-Programa-Operacional

Case Study: A news stories portal

- Association Rules for recommendation
  - User reads articles A
  - Site knows the rule A → B
  - The rule has a certain confidence (assume > 20%)
  - Site displays articles B to the user
  - User chooses whether to follow recommendation or not

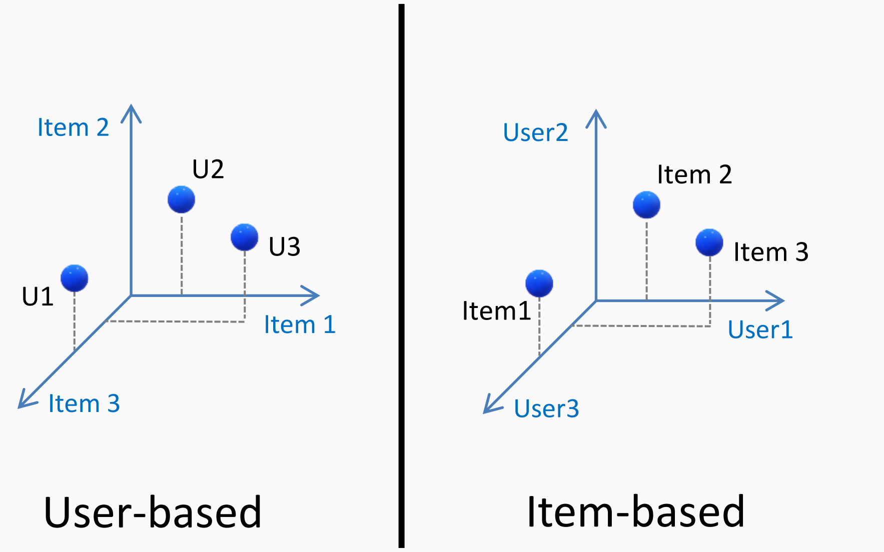
Notes:
- The rules are discovered from user activity
- Discovery is off-line
- Rule application is on-line

Case Study: A news stories portal

- Recommendation 1
  - Seen:
    - "Medidas-de-combate-à-fraude-e-evasão-fiscal"
  - Recommended:
    - "Impacto-das-medidas-fiscais-Orçamento-do-Estado-2005" (0.97)
    - "Principais-alterações-em-sede-de-IRS" (0.75)
    - "Rotas-Uteis" (0.28)
- Recommendation 2
  - Seen:
    - "Medidas-de-combate-à-fraude-e-evasão-fiscal"
    - "Peter-Cohan-Não-penso-que-haja-uma-retoma"
  - Recommended:
    - "Impacto-das-medidas-fiscais-Orçamento-do-Estado-2005" (0.97)
    - "O-valor-de-Peter-Cohan" (0.75)
    - "Principais-alterações-em-sede-de-IRS" (0.75)
    - "Rotas-Uteis" (0.28275)

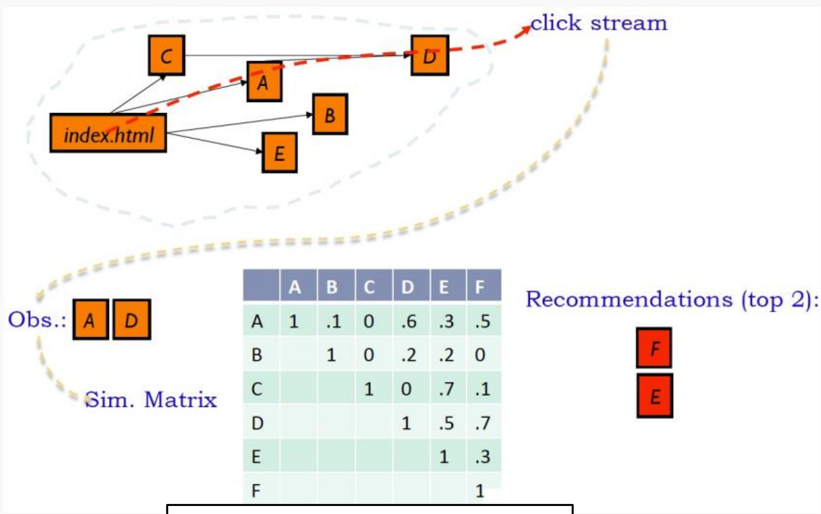
Collaborative Filtering

Collaborative Filtering - idea of neighbors



31

Collaborative Filtering item-based (General Idea)



	A	B	C	D	E	F
A	1	.1	0	.6	.3	.5
B		1	0	.2	.2	0
C			1	0	.7	.1
D				1	.5	.7
E					1	.3
F						1

Similarity Matrix between **items**

DM II / ATDS - 23/24: Web Mining: Recommender Systems - Collaborative Filtering

32

Collaborative Filtering (observations)

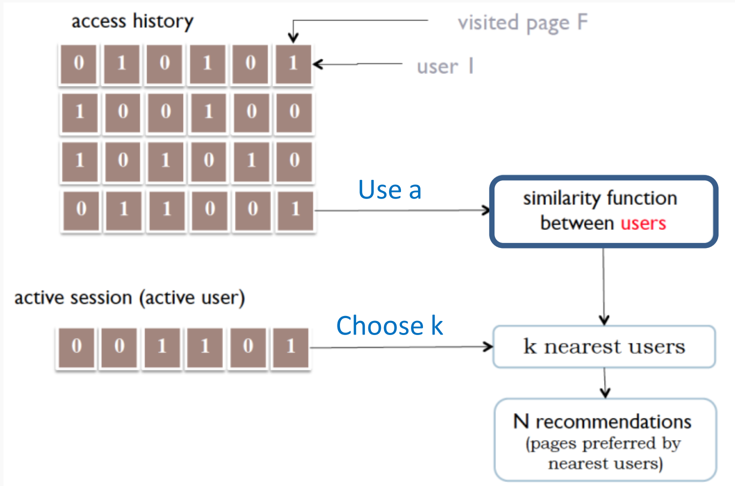
- In Collaborative Filtering (CF) we don't need to know anything about an item **except who else has liked, viewed or ignored it.**
- Two items are not considered similar because of their content, but because they were **liked, viewed or ignored** by a similar set of users.
- Data types for rating:
  - **binary** ratings
    - web: accessed / did not accessed
    - e-commerce: bought / did not bought
  - **ordinal** ratings
    - movies: 5 ⋆ system
  - **unary** (positive ratings)
  - **continuous** ratings

Collaborative Filtering (neighborhood)

CF neighborhood-based methods:

- User-based CF:
  similar users provide similar ratings on the same item;
  - the information provided by similar users to a target user A is used to make **recommendations** for A.
- Item-based CF:
  similar items are rated in a similar way by the same user
  - for a given target item I, the information provided by a particular user A on a set of similar items S is used to **predict** the rating of user A for item I. → the ratings

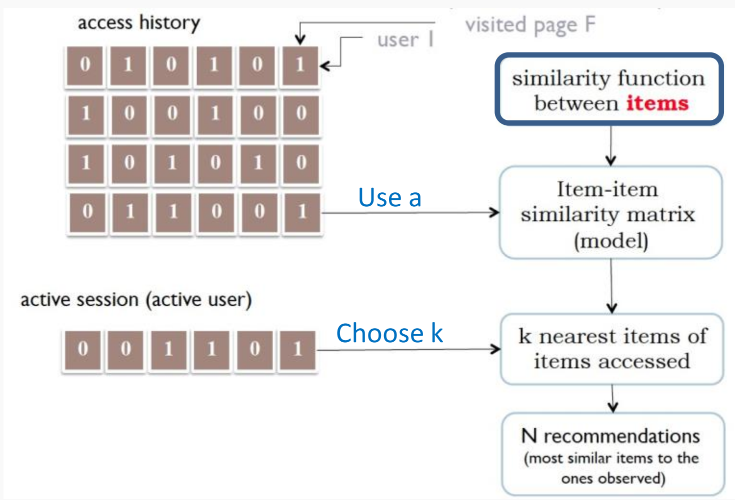
User-based Collaborative Filtering (General Idea)



DM II / ATDS - 23/24: WEB MINING: RECOMMENDER SYSTEMS - COLLABORATIVE FILTERING

35

Item-based Collaborative Filtering (General Idea)



DM II / ATDS - 23/24: WEB MINING: RECOMMENDER SYSTEMS - COLLABORATIVE FILTERING

36

Similarity measures

- Cosine similarity
- Pearson correlation coefficient
- Jaccard similarity
- ...

Most used

But **distance measures** can also be used

- Euclidean distance
- Manhattan distance
- Hamming distance
- ...

Any one is represented as a matrix

Note: some similarity measures are invariant to magnitude, which might be useful.

37

Similarity measures (cont.)

Set of users: $\boldsymbol{U}$
Set of items: $\boldsymbol{I}$

- **Cosine Similarity**
  - item-based similarity (**column-wise**)

$$sim(i,j) = \cos(\vec{i},\vec{j}) = \frac{\sum_{u \in U} i_u \times j_u}{\sqrt{\sum_{u \in U} i_u^2} \times \sqrt{\sum_{u \in U} j_u^2}}$$

  - user-based similarity (**row-wise**)

$$sim(u,v) = \cos(\vec{u},\vec{v}) = \frac{\sum_{i \in I} u_i \times v_i}{\sqrt{\sum_{i \in I} u_i^2} \times \sqrt{\sum_{i \in I} v_i^2}}$$

38

Similarity measures (cont.)

Consider the user-page access matrix (binary rating)

	A	B	C	D	E	F
u1	0	1	0	1	0	1
u2	1	0	0	1	0	0
u3	1	0	1	0	1	0
u4	0	1	1	0	0	1

- Cosine similarity between pages

$$sim(A,F) = \frac{0}{\sqrt{1+1}\times\sqrt{1+1}} = 0$$

$$sim(C,F) = \frac{1}{\sqrt{1+1}\times\sqrt{1+1}} = 0.5$$

	A	B	C	D	E	F
u1	0	1	0	1	0	1
u2	1	0	0	1	0	0
u3	1	0	1	0	1	0
u4	0	1	1	0	0	1

- Cosine similarity between users

$$sim(u1,u2) = \frac{1}{\sqrt{1+1+1}\times\sqrt{1+1}} \approx 0.4$$

$$sim(u1,u4) = \frac{1+1}{\sqrt{1+1+1}\times\sqrt{1+1+1}} \approx 0.7$$

Producing USER-based recommendations

- given an **active user u_a**
- find $N(u_a, k)$, the k-nearest neighbors of u_a
- compute the score of each item i viewed by its neighbors

$$\boldsymbol{score}(u_a,i) = \frac{1}{k}\times\sum_{v\in N(u_a,k)} sim(u_a,v)\times \underbrace{viewed(v,i)}_{\substack{\text{1-Yes}\\ \text{0-No}}}$$

- recommend the items with **highest score**

Producing ITEM-based recommendations

- given an **active session s_a**
- compute the score of each item i
  - find $N(i, k)$, k-nearest neighbors of i
  - consider the intersection of s_a and the neighbors of i

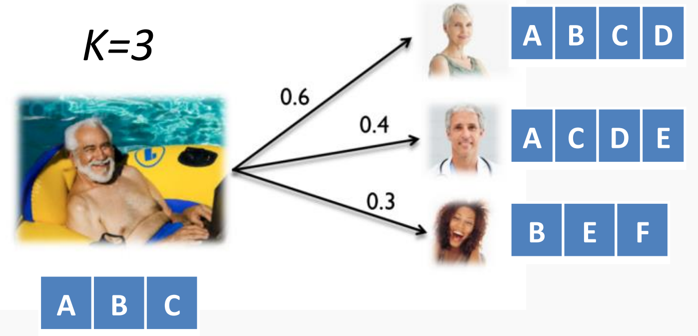
$$\boldsymbol{score}(S_a, i) = \frac{\sum_{j \in s_a \cap N(i,k)} sim(i,j)}{\sum_{j \in N(i,k)} sim(i,j)}$$

- recommend the items with **highest score**

41

Example:

Settings: knn + user-based + binary



42

Recommendations with binary ratings: Exercise

Consider the user-page access table

USER	PAGE
1	A
1	B
1	C
2	A
2	C
3	B
3	G
3	F
3	I
4	B
4	C
5	G
5	F
5	I
5	J
6	A
6	C

1. Build the **similarity cosine matrix** for:
   1. User-based approach
   2. Item-based approach
2. Compute the top2 **recommendations** for:
   1. A session <B,G>, using user-based CF
   2. User4, using item-based CF

43

Recommendations with non-binary ratings

- A user gives ratings to items. Examples:
  - 5 ⋆ scale
  - or any numeric scale *S*

- **Problem**
  - predict the rating a user $u \in U$ would give to an unseen item $i \in I$

$$f : U \times I \rightarrow S$$

44

Recommendations with non-binary ratings (cont.)

- How to recommend?
  - given an active user u_a
  - find items j from the set of **items not seen by the user** ($I \setminus I_u$) that maximize the rating function $f(u_a, j)$:

$$top\ relevant\ item = arg_{j \in I \setminus I_u} \max f(u_a, j)$$

- Methods:
  - User-based Unweighted Method
  - User-based Weighted Method
  - User-based Weighted and Mean Centered Method
- The same methods exist for item-based!

45

Recommendations with ratings

USER	ITEM	RATING
1	A	1
1	B	3
1	G	4
2	A	4
2	C	2
3	B	4
3	G	5
3	F	3
3	I	4
4	B	5
4	C	4
5	G	3
5	F	4
5	I	5
5	J	3
6	A	5
6	C	3

- How would **u2** rate **B**?

u2: A,C

Therefore, we should look at users that

- Have seen A or C and
- Have also seen B

We should look at **u1** and **u4**

From them we get the values 3 and 5

Now, what would be the rate **u2** gives **B**?

details

46

Recommendations with non-binary ratings

We are considering only similarities between users

- **User-based** Unweighted Method:
  - use $N_i(u,k)$, ***k*-nn** of u who have **rated the item *i***

$$\hat{r}_{ui} = \frac{1}{k} \sum_{v \in N_i(u,k)} r_{vi}$$

→ (3+5)/2 = 4

	A	B	C	F	G	I	J
u1	1	1	1	0	0	0	0
u2	1	0	1	0	0	0	0
u3	0	1	0	1	1	1	0
u4	0	1	1	0	0	0	0
u5	0	0	0	1	1	1	1
u6	1	0	1	0	0	0	0

`dist()`

	u1	u2	u3	u4	u5
u2	1.00				
u3	2.24	2.45			
u4	1.00	1.41	2.00		
u5	2.65	2.45	1.41	2.45	
u6	1.00	0.00	2.45	1.41	2.45

- **User-based** Weighted Method
  - use $N_i(u,k)$, ***k*-nn** of u who have **rated the item *i***
  - using the **weights w_{uv}**, with $v \in N_i(u,k)$

similarity between users

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i(u,k)} w_{uv} \times r_{vi}}{\sum_{v \in N_i(u,k)} |w_{uv}|}$$

→ (1*3 + 1.41*5)/(1 + 1.41) = 4.17

this method is not sensitive to distance

47

Recommendations with non-binary ratings

We are **considering rates as distances** between users

- **User-based** Unweighted Method:
  - use $N_i(u,k)$, ***k*-nn** of u who have **rated the item *i***

$$\hat{r}_{ui} = \frac{1}{k} \sum_{v \in N_i(u,k)} r_{vi}$$

→ (3+5)/2 = 4

	A	B	C	F	G	I	J
u1	1	3	2	0	0	0	0
u2	4	0	2	0	0	0	0
u3	0	4	0	3	5	4	0
u4	0	5	4	0	0	0	0
u5	0	0	0	4	3	5	3
u6	5	0	3	0	0	0	0

`max(dist())-dist()`

	u1	u2	u3	u4	u5
u2	5.76				
u3	2.52	0.73			
u4	7.00	3.29	1.81		
u5	1.46	1.11	4.43	0.00	
u6	4.90	8.59	0.00	2.86	0.36

- **User-based** Weighted Method
  - use $N_i(u,k)$, ***k*-nn** of u who have **rated the item *i***
  - using the **weights w_{uv}**, with $v \in N_i(u,k)$

distances between users

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i(u,k)} w_{uv} \times r_{vi}}{\sum_{v \in N_i(u,k)} |w_{uv}|}$$

→ (5.76*3 + 3.29*5)/(5.76 + 3.29) = 4.17

48

Recommendations with non-binary ratings

- User-based **Weighted** and **Mean Centered Method**
  - mean-centering is a form of **normalization**
  - use $N_i(u,k)$, k-nn of u who have rated the item i
  - the weights w_{uv}, with $v \in N_i(u,k)$
  - the **mean of ratings** made by user u, expressed as $\bar{r}_u$

$$\hat{r}_{ui} = \bar{r}_u \frac{\sum_{v \in N_i(u,k)} w_{uv} \times (r_{vi} - \bar{r}_v)}{\sum_{v \in N_i(u,k)} |w_{uv}|}$$

Recommendations with non-binary ratings: Exercise

Consider the user-page rating table

USER	PAGE	RATING
1	A	1
1	B	3
1	C	2
2	A	4
2	C	2
3	B	4
3	G	5
3	F	3
3	I	4
4	B	5
4	C	4
5	G	3
5	F	4
5	I	5
5	J	3
6	A	5
6	C	3

For two neighbors and for each of the methods:

- user-based unweighted method
- user-based weighted method
- user-based weighted and mean centered method

1. How would User2 rate page B?
   (to be seen)
2. How would User1 rate page F?

homework!

USER	PAGE	RATING
1	A	1
1	B	3
1	C	2
2	A	4
2	C	2
3	B	4
3	G	5
3	F	3
3	I	4
4	B	5
4	C	4
5	G	3
5	F	4
5	I	5
5	J	3
6	A	5
6	C	3

General steps to compute to predict User2 rating for item B using the user-based unweighted method, a neighborhood of 2 and cosine distance

1. Compute the cosine similarity between user2 and all other users who have rated item B (and share seen items).
2. Select the 2 nearest neighbors (users with the highest cosine similarity) who have rated item B.
3. Take the average of the ratings given by these 2 neighbors for item B.

User2 has rated items A and C, and we want to predict the rating for item B. We need to find users who have rated both items A and B, or items B and C. In this case, users 1 and 4 have rated item B.

User1	User2	User4
A:1	A:4	B:5
B:3	C:2	C:4
C:2		

Now, compute the cosine similarity between User2 and users 1 and 4.

The vectors for **users 1 and 2** based on common items are:
User1: (A:1, C:2)
User2: (A:4, C:2)

Cosine similarity(User1, User2) = (1 * 4 + 2 * 2) / (sqrt(1^2 + 2^2) * sqrt(4^2 + 2^2)) = 10 / (sqrt(5) * sqrt(20)) = 10 / 10 = **1**.

The vectors for **users 2 and 4** based on common items are:
User 2: (C:2)
User 4: (C:4)

Cosine similarity(User 2, User 4) = (2 * 4) / (sqrt(2^2) * sqrt(4^2)) = 8 / (2 * 4)) = 8 / 8 = **1**

Now, we have the nearest neighbors: User1 (cosine similarity = 1) and User 4 (cosine similarity = 1).

The predicted rating for item B by User2 would be the **average** of the nearest neighbors' ratings for item B: (3+5) /2 = **4**.

51

USER	PAGE	RATING
1	A	1
1	B	3
1	C	2
2	A	4
2	C	2
3	B	4
3	G	5
3	F	3
3	I	4
4	B	5
4	C	4
5	G	3
5	F	4
5	I	5
5	J	3
6	A	5
6	C	3

General steps to compute to predict User2 rating for item B using the user-based weighted method, a neighborhood of 2 and cosine distance

We have already found the nearest neighbors: **User1** (cosine similarity = 1) and **User4** (cosine similarity = 1).

Now, we'll calculate the weighted average of the nearest neighbors' ratings for item B:

Predicted rating for User2 on item B = (User1 rating * cosine similarity(User1, User2) + User4 rating * cosine similarity(User4, User2)) / (cosine similarity(User1, User2) + cosine similarity(User4, User2))

Predicted rating for user 2 on item B = (3 * 1 + 5 * 1) / (1 + 1) = (3 + 5) / 2 = 8 / 2 = **4**.

Using the user-based weighted method with a neighborhood of 2 and cosine distance, the predicted rating for item B by User2 is **4**.

52

USER	PAGE	RATING
1	A	1
1	B	3
1	C	2
2	A	4
2	C	2
3	B	4
3	G	5
3	F	3
3	I	4
4	B	5
4	C	4
5	G	3
5	F	4
5	I	5
5	J	3
6	A	5
6	C	3

General steps to compute to predict User2 rating for item B using the user-based weighted and mean-centered method, a neighborhood of 2 and cosine distance

We need to calculate the average rating of each user and adjust the ratings accordingly.

User1: (1 + 3 + 2) / 3 = 2

User2: (4 + 2) / 2 = 3

User4: (5 + 4) / 2 = 4.5

Next, we'll **adjust the ratings** by subtracting each user's average rating from their respective ratings for item B:

User1: 3 - 2 = 1

User4: 5 - 4.5 = 0.5

Now, we calculate the **weighted average of the nearest neighbors' adjusted ratings for item B**:

Predicted rating for **User2 on item B (mean-centered)** = (User1 adjusted rating * cosine similarity(User1, User2) + User4 adjusted rating * cosine similarity(User4, User2)) / (cosine similarity(User1, User2) + cosine similarity(User4, User2))

= (1 * 1 + 0.5 * 1) / (1 + 1) = (1 + 0.5) / 2 = 1.5 / 2 = 0.75

Finally, we'll add **User2 average rating** to the mean-centered predicted rating to get the final predicted rating:

Final predicted rating for User2 on item B = User2 average rating + mean-centered predicted rating = (4 + 2) / 2 + 0.75 = 3 + 0.75 = 3.75

Using the user-based weighted and mean-centered method with a neighborhood of 2 and cosine distance, the predicted rating for item b by user 2 is **3.75**.

53

Recommendations with non-binary ratings: Exercise (cont.)

USER	PAGE	RATING
1	A	1
1	B	3
1	C	2
2	A	4
2	C	2
3	B	4
3	G	5
3	F	3
3	I	4
4	B	5
4	C	4
5	G	3
5	F	4
5	I	5
5	J	3
6	A	5
6	C	3

4. Answer the same questions above, but using **item-based** distances instead of user-based distances.

- **Note**:
  item-based unweighted method uses $N_u(i,k)$, k-nn of item i rated by user u.

$$\hat{r}_{ui} = \frac{1}{k} \sum_{j \in N_u(i,k)} r_{uj}$$

more homework!

54

Challenges

- **Scalability**
- Sparsity
- Incrementality
- **Cold start**
- Considering context
- Background knowledge
- Combining content, structure and usage

55

Challenges

- **Scalability**: User-based CF can become computationally intensive with a large number of users, whereas item-based CF can be more scalable as the item-item similarity matrix can be precomputed and does not change as often.
- **Performance**: The performance of each method can vary depending on the dataset and the domain. In some cases, item-based CF can outperform user-based CF, especially when there is a large amount of user data and the user's preferences are not highly dynamic.

56

Challenges

The cold start problem in recommender systems arises from the challenge of recommending for **new users or items with little data**.

- **User-based Cold Start**: With user-based CF, the cold start problem for **new users** can be particularly challenging because the system's effectiveness largely depends on comparing the new user's interests with those of existing users. Without any data on the new user, the system cannot accurately determine which existing users are similar.
- **Item-based Cold Start**: Item-based CF encounters a cold start problem with **new items** because it lacks historical ratings or interactions from users. Without this data, the system cannot determine which items are similar to the new one and thus cannot recommend it to users who might have shown an interest in similar items.

Solutions to the cold start problem often involve:

- **Hybrid Approaches**: Combining collaborative filtering with content-based filtering, where recommendations are based on the content features of items or user profiles, rather than just historical interaction data.
- **Early User Profiling**
- **Utilizing Demographic Data**
- **Encouraging Early Rating**

57