Practical Data Science: Recommender Systems II

Dr. Yongli Ren

(yongli.ren@rmit.edu.au)

Computer Science & IT

School of Science

Outline

- Part 1: What is Collaborative Filtering?
- Part 2: How to Do Collaborative Filtering?
- Part 3: How to Evaluate Collaborative Filtering?

Practical Data Science – COSC2670

PART 1: WHAT IS COLLABORATIVE FILTERING?

Collaborative Filtering - At Amazon

Roll over image to zoom in

See all 14 customer images

Share your own customer images

Apple iPhone 5 16GB (Black)

by Apple

(151 customer reviews)

List Price: ~~$750.00~~

Price: $696.95

You Save: $53.05 (7%)

Only 12 left in stock.

Ships from and sold by PRIME ELECTRONICS.

125 new from $480.00 65 used from $500.00

Thinking of Selling Your iPho

Now Amazon has three easy wa
for Cash as an Individual Seller,

http://www.tatvic.com/blog/an-introduction-to-collaborative-filtering/

The principle is like this:

if several members of my community owned and liked the latest Apple gadget, then it is highly likely that I will too.

Customers who bought this item also bought

iPhone 5s case iPhone SE case iPhone 5 case by Ailun Shock-Absorption Bumper TPU Clear...

1,135

$5.89 prime

iPhone 5S Screen Protector,iPhone SE Screen Protector,[2 Packs]by Ailun,2.5D Edge...

1,240

$5.89 prime

Balee iphone 5s Screen Protector,Ultra Thin Anti-Scratch Tempered Glass Screen Protector for...

1,411

$5.96 prime

iPhone SE Case, for iPhone 5s 5 SE Limecase [NGS Series] Slim Fit Heavy Duty Protection Case...

2,664

$7.99 prime

T-Mobile Prepaid Complete SIM Starter Kit - No Contract Network Connection (Universal:...

1,178

$9.99 prime

https://www.sitepoint.com/ux-lessons-from-amazon-4-hacks-guaranteed-to-boost-conversions/

What is Collaborative Filtering?

- Collaborative filtering (CF)
  - is a technique used by recommender systems
    - the most successful recommendation technique to date
  - is the idea of recommending an item depending on other like minded individuals
  - consists of
    - set of users,
    - set of items, and
    - set of opinions about the item: ratings, reviews or purchases.

User-Item Rating Matrix

	Troy	The Godfather	Titanic	Forrest Gump
Fahime	5	-	5	1
Musi	5	-	-	1
Hamza	4	4	5	1
Paul	4	-	5	5
Adam	1	2	-	5

Note '-' means there is no rating.

What is Collaborative Filtering?

- In the narrower sense,
  - collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user
    - by collecting preferences or taste information from many users (collaborating).
  - The underlying assumption is that
    - if person A has the same opinion as person B on an issue,
    - A is more likely to also have B's opinion on a different issue than that of a randomly chosen person.
  - This differs from *MovieAvg* (or *TopPop*)
    - The simpler approach of giving an average score (or popularity) for each item of interest.

https://en.wikipedia.org/wiki/Collaborative_filtering

Collaborative Filtering
- Neighbourhood-based Methods

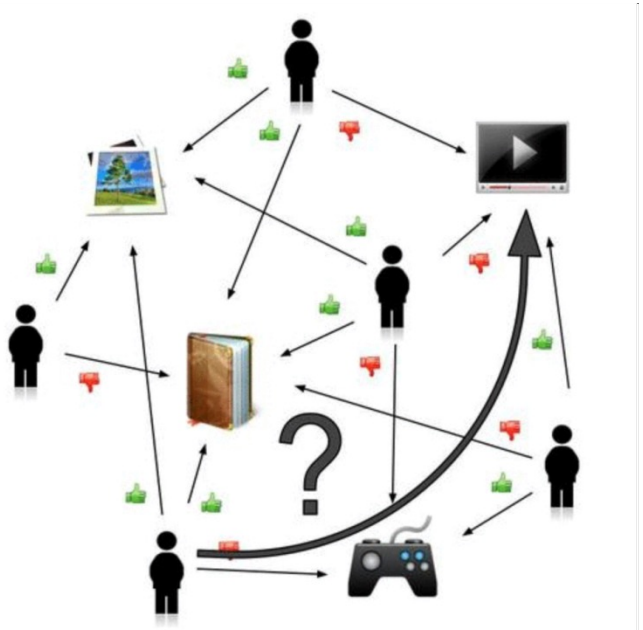
- Consider the active user x;
- find set N of other users whose ratings are "similar" to x's ratings;
- Estimate x's ratings based on ratings of users in N

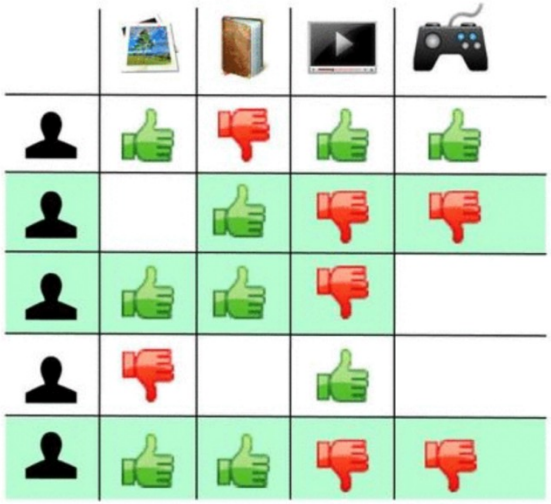
https://www.youtube.com/watch?v=h9gpufJFF-0

Practical Data Science – COSC2670

PART 2:
HOW TO DO COLLABORATIVE FILTERING?

The Process of Collaborative Filtering





This image shows an example of predicting of the user's rating using collaborative filtering.

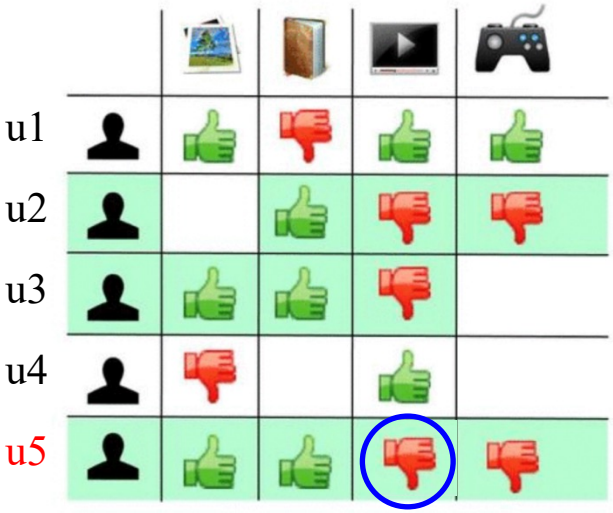
1. People rate different items.
2. The system is making predictions about the active user's rating for an item, which the user hasn't rated yet.
3. These predictions are built upon the existing ratings of other users, who have similar ratings with the active user.

For instance, in our case the system has made a prediction, that the active user won't like the video.

https://en.wikipedia.org/wiki/Collaborative_filtering

kNN-based Collaborative Filtering

- Input: the User-Item rating matrix.
- Output: the rating of the active user on the target item



	u1	u2	u3	u4
u5	1	2	2	0

The number of items they have common opinions

For the active user

Similarities to other users

Find the k-nearest neighbours

Aggregate neighbours' ratings

Predict rating on target item

https://en.wikipedia.org/wiki/Collaborative_filtering

Similar Users

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

- Consider users A and B with rating vectors (row A and B)
- We need a similarity metric $sim\ (A,B)$:
  - Capture intuition that $sim(A,B) > sim(A,C)$

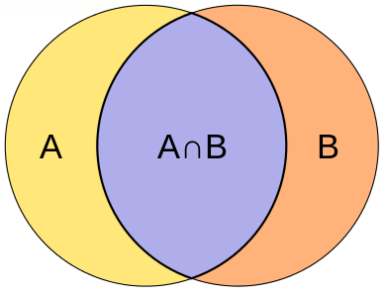
Similar Users
-Jaccard Similarity

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

$$sim(A,B) = \frac{|I(A) \cap I(B)|}{|I(A) \cup I(B)|}$$

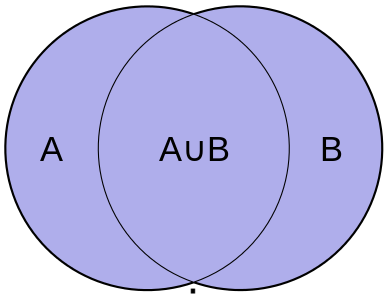
Co-rated item set

- $I(A) = \{HP1, TW, SW1\}$
- $I(B) = \{HP1, HP2, HP3\}$



Intersection

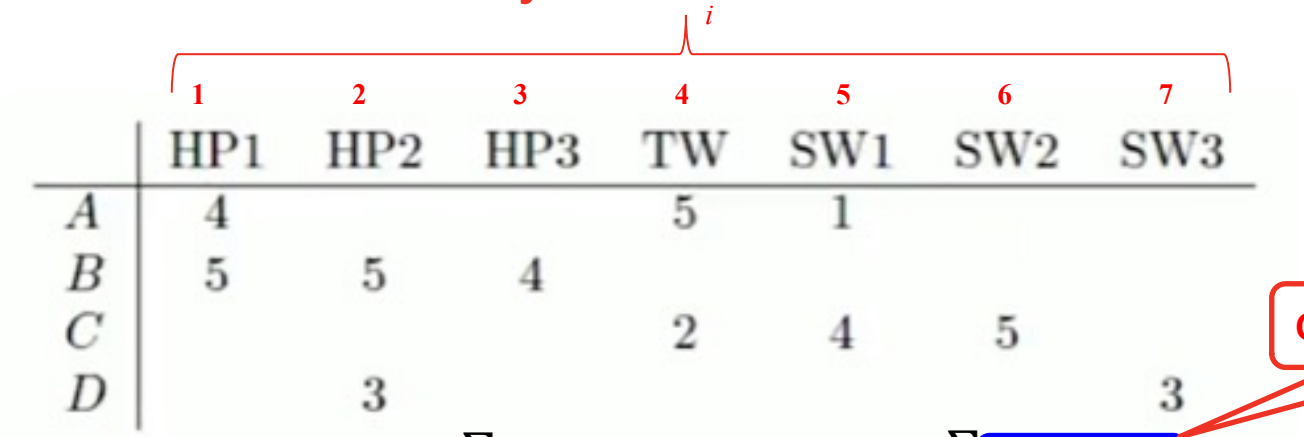
- $sim(A,B) = 1/5;$
- Similarly, $sim(A,C) = 2/4$, where
  - $I(C) = \{TW, SW1, SW2\}$
- $sim(A,B) < sim(A,C)$
- Problem: ignores rating values



union

https://en.wikipedia.org/wiki/Jaccard_index https://www.youtube.com/watch?v=h9gpufJFF-0

Similar Users
-Cosine Similarity



$$sim(A,B) = \cos(A,B) = \frac{\sum_i r_{a,i} r_{b,i}}{\sqrt{\sum_i r_{a,i}^2}\sqrt{\sum_i r_{b,i}^2}} = \frac{\sum_{i \in I(A) \cap I(B)} r_{a,i} r_{b,i}}{\sqrt{\sum_{i \in I(A) \cap I(B)} r_{a,i}^2}\sqrt{\sum_{i \in I(A) \cap I(B)} r_{b,i}^2}}$$

Co-rated item set

- i is the index for item
- $r_{a,i}$ is the rating from user A on item i, and similarly $r_{b,i}$ is the rating from user B on item i; e.g. $r_{a,1} = 4, r_{b,2} = 5, r_{c,4} = 2, r_{d,7} = 3$

- If missing values are treated as zero
  - $sim(A,B) = 1, sim(A,C) = 0.61;$
  - $sim(A,B) > sim(A,C)$, but not by much
- Problem: treats missing ratings as negative

e.g. sim(A,C) = (5*2 + 1*4)/(sqrt(25+1) sqrt(4 + 16)) =0.6139406, which is around 0.61

Similar Users
-Centered Cosine Similarity

- Normalize ratings by subtracting row mean

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

(4+5+1)/3
(5+5+4)/3
(2+4+5)/3
(3+3)/2

	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

$$sim(A,B) = \frac{\sum_{i \in I(A) \cap I(B)} (r_{a,i} - \bar{r_a})(r_{b,i} - \bar{r_b})}{\sqrt{\sum_{i \in I(A) \cap I(B)} (r_{a,i} - \bar{r_a})^2} \sqrt{\sum_{i \in I(A) \cap I(B)} (r_{b,i} - \bar{r_b})^2}}$$

Similar Users
-Centered Cosine Similarity

	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

- $sim(A,B) = cos(A, B) = 1$
- $sim(A, C) = -0.73$
- $sim(A,B) > sim(A,C)$
- Captures intuition better
  - Missing ratings treated as "average"
  - Handles "tough raters" and "easy raters"
- Also known as "Pearson Correlation Coefficient (PCC)"

https://www.youtube.com/watch?v=h9gpufJFF-0

Similar Users

-Co-rated Item Set in Centered Cosine Similarity

	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

- $sim(A,B) = cos(A, B) = 1$
- $sim(A, C) = -0.73$
- $sim(A,B) > sim(A,C)$

Co-rated item set

$$sim(A,B) = \frac{\sum_{i \in I(A) \cap I(B)} (r_{a,i} - \bar{r}_a)(r_{b,i} - \bar{r}_b)}{\sqrt{\sum_{i \in I(A) \cap I(B)} (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{i \in I(A) \cap I(B)} (r_{b,i} - \bar{r}_b)^2}}$$

- If the size of co-rated item set is too small, the corresponding similarity is likely not that reliable. Thus, significance weighting is introduced:

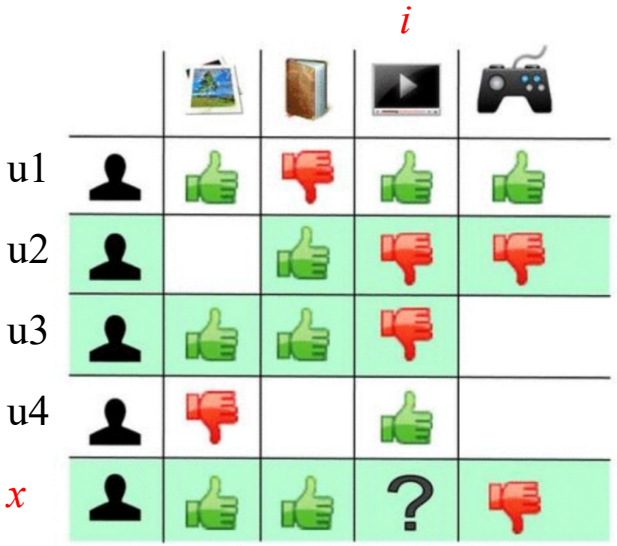
$$sim(A,B) \leftarrow \frac{\min(|I(A) \cap I(B)|, \gamma)}{\gamma} sim(A,B)$$

  - where γ is a pre-defined a parameter.

https://www.youtube.com/watch?v=h9gpufJFF-0

Rating Predictions

- Let $r_{x,i}$ be the rating of the active user x on item i
- let $S(x)$ be the set of k users most similar to x who have also rated item i
  - $S(x) = \{u2, u3\}$
- Prediction for $r_{x,i}$
  - option 1: average rating in the neighbourhood;
  - (👎 + 👎) / 2 = 👎
  - option 2: weighted average rating in the neighbourhood
  - (2 * 👎 + 2 * 👎) / (2+2) = 👎
  - Option 3: weight the difference to the average rating by the corresponding similar users:



	u1	u2	u3	u4
x	**1**	**2**	**2**	**0**

$$P(r_{x,i}) = \bar{x} + \frac{\sum_{u \in S(x)} sim(x,u)(r_{u,i} - \bar{u})}{\sum_{u \in S(x)} sim(x,u)}$$

https://www.youtube.com/watch?v=h9gpufJFF-0

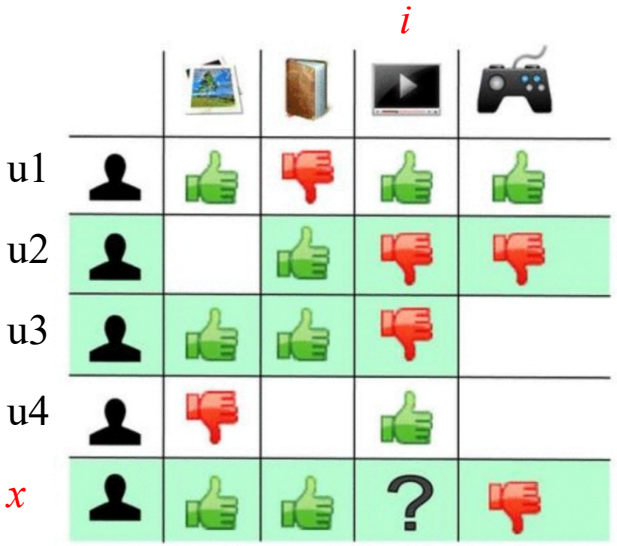
Rating Predictions

- Let r_x be the vector of user x's ratings
- let $S(x)$ be the set of k users most similar to x who have also rated item i
  - $S(x) = \{u2, u3\}$
- Prediction for users x and item i
  - Option 3: weight the the difference to the average rating by the corresponding similar users:
    - 👍: 1; 👎: -1
  - Then,

$$\bar{x} = \frac{(1+1-1)}{3} = \frac{1}{3}$$

$$\overline{u2} = \frac{(1-1-1)}{3} = -\frac{1}{3}$$

$$\overline{u3} = \frac{(1+1-1)}{3} = \frac{1}{3}$$



	u1	u2	u3	u4
x	**1**	**2**	**2**	**0**

$$P(r_{x,i}) = \frac{1}{3} + \frac{2 * \left(-1 - \left(-\frac{1}{3}\right)\right) + 2\left(-1 - \frac{1}{3}\right)}{2+2} = -\frac{2}{3}$$

https://www.youtube.com/watch?v=h9gpufJFF-0

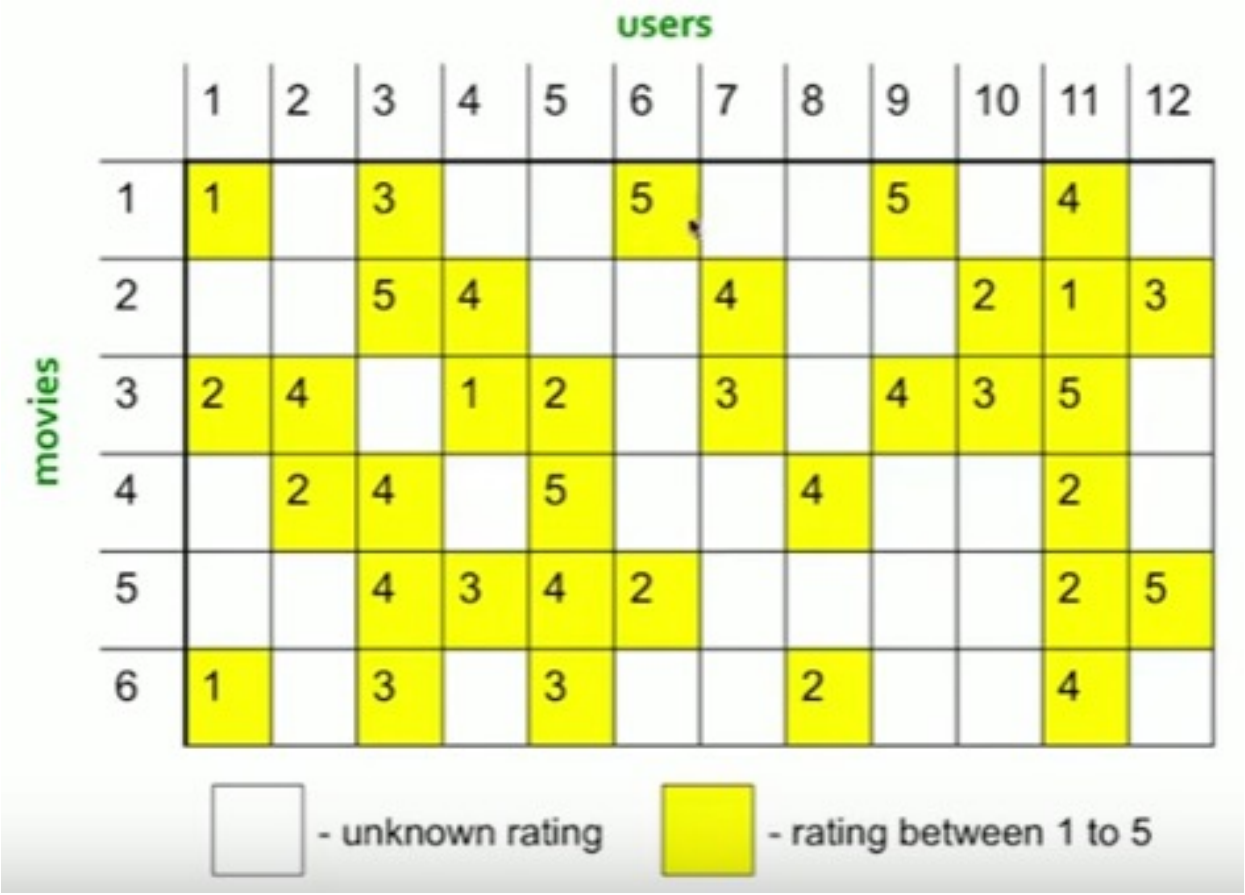
Item-item Collaborative Filtering

- So far: user-user collaborative filtering
- Another view: item-item
  - for item i, find other similar items
  - estimate rating for item i based on ratings for similar items;
  - can use same similarity metrics and prediction functions as in user-user model

https://www.youtube.com/watch?v=h9gpufJFF-0

Item-item Collaborative Filtering (|N| = 2)

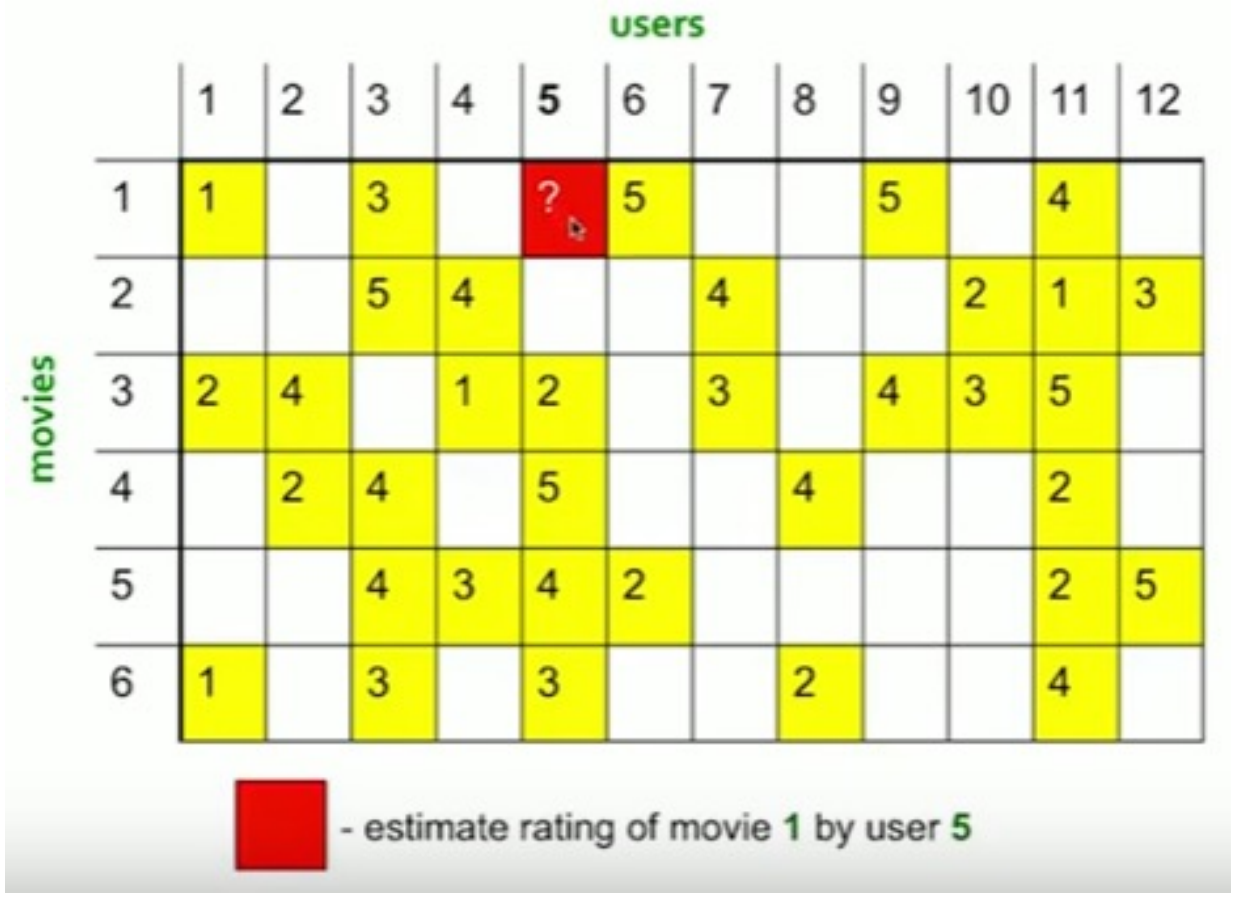
The input will be the Transpose[1] of the User-Item Rating Matrix



1. https://en.wikipedia.org/wiki/Transpose

https://www.youtube.com/watch?v=h9gpufJFF-0

Item-item Collaborative Filtering (|N| = 2)



	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

https://www.youtube.com/watch?v=h9gpufJFF-0

Item-item Collaborative Filtering (|N| = 2)

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		0.41
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		0.59

Neighbour selection:
Identify movies similar to movie 1, rated by user 5

Here, we use Centered Cosine as similarity:

1) Subtract mean rating from each movie:
   meanRating1 = (1+3+5+5+4)/5 = 3.6
   row 1 = [-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0.4)
2) Compute cosine similarities between rows

Predict by taking the weighted difference to the average rating by the corresponding similar users(Option 3):
3.6 + (0.41 * (2 – 3) + 0.59*(3-2.6))/(0.41 + 0.59) = 3.426

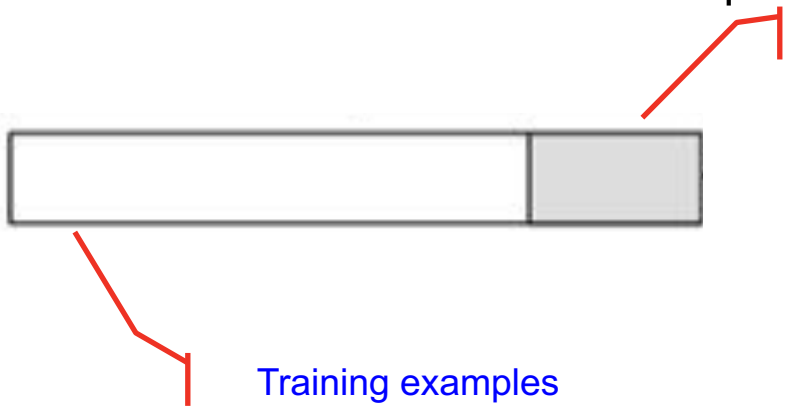
https://www.youtube.com/watch?v=h9gpufJFF-0

Practical Data Science – COSC2670

PART 2:
HOW TO EVALUATE COLLABORATIVE FILTERING?

Evaluation
- Validate Strategy

- Training Test Split.
  - *Randomly dividing your data into a training set with X% of the observations and keeping the rest as a holdout data set*
    - (a data set that's never used for model creation)
  - This is the most common technique.




```
from sklearn.model_selection import train_test_split
train_df, test_df = train_test_split(df, test_size=0.2, random_state=0)
train_df, test_df
```

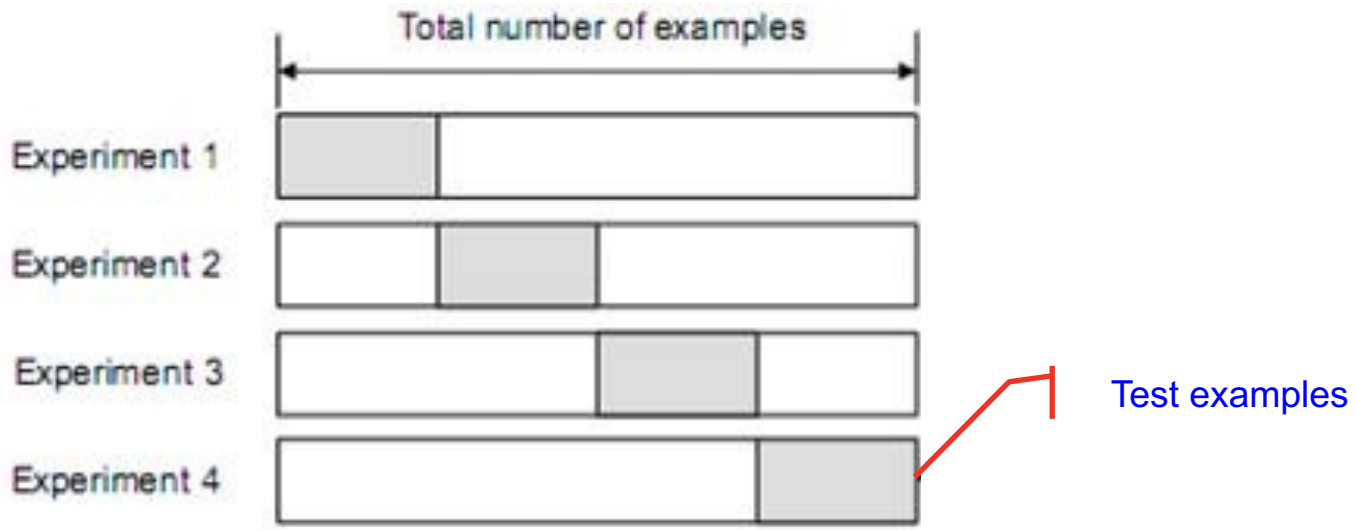
```
(       user_id  item_id  rating  timestamp
 10382       70      298       5  884064134
 73171      215      172       4  891435394
 30938      488      210       4  891294896
 99310      916      156       5  880844016
 58959      292      197       5  881105246
 ...        ...      ...     ...        ...
 21243      379      187       5  880525031
 45891      526      750       4  885681886
 42613       11      717       2  891902815
 43567       94      328       3  891724990
 68268      474      186       4  887925977

 [80000 rows x 4 columns],
        user_id  item_id  rating  timestamp
 3582        23      528       4  874786974
 60498      695      242       5  888805837
 53227      774       28       3  888556698
 21333      417      550       3  879649178
 3885       234     1035       3  892335142
 ...        ...      ...     ...        ...
 60116      659        4       3  891383917
 2415        14      709       5  879119693
 43763      629      660       5  880117772
 71345      892      214       2  886608897
 77687      417      663       3  879647040

 [20000 rows x 4 columns])
```

Evaluation
- Validate Strategy

- *k-folds cross validation*
  - This strategy divides the data set into k parts and uses each part one time as a test data set while using the others as a training data set.
  - This has the advantage that you use all the data available in the data set.



D. Cielen et. Al., Chapter 3

Evaluation
- MAE and RMSE

- **Mean-Absolute-Error (MAE) and Root-mean-square error** (**RMSE**) measure the differences between the predicted ratings by a recommender system and the actual ratings observed.
  - https://en.wikipedia.org/wiki/Mean_absolute_error

$$MAE = \frac{\sum_{r_{x,i} \in T} |\widehat{r_{x,i}} - r_{x,i}|}{n}$$

  - https://en.wikipedia.org/wiki/Root-mean-square_deviation

$$RMSE = \sqrt{\frac{\sum_{r_{x,i} \in T} (\widehat{r_{x,i}} - r_{x,i})^2}{n}}$$

  - where n is the size of the test set T; $\widehat{r_{x,i}}$ is the predicted rating for the actual observed rating $r_{x,i}$

K Nearest Neighbour-based Collaborative Filtering

- Please check the KNN_based_CF.ipynb for details
  - The file is available in this week's module in the course Canvas.

References and Further Reading

- Xiaoyuan Su and Taghi M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 19 pages, 2009. doi:10.1155/2009/421425
- Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. on Knowl. and Data Eng. 17, 6 (June 2005), 734-749. DOI: https://doi.org/10.1109/TKDE.2005.99
- Yongli Ren, Gang Li, Wanlei Zhou: A survey of recommendation techniques based on offline data processing. Concurrency and Computation: Practice and Experience 27(15): 3915-3942 (2015)

Data
Science

Thanks!