Machine Learning with Graphs (MLG)

Recommender Systems (RecSys)

Basic & Collaborative Filtering

Cheng-Te Li (李政德)

Institute of Data Science

National Cheng Kung University

chengte@mail.ncku.edu.tw

YouTube

推薦影片

GS Warriors vs Houston Rockets - Full Game...

Rapid Highlights

觀看次數：15萬・2 小時前

Golden State Warriors vs Houston Rockets Full Game...

MLG Highlights

觀看次數：34萬・2 小時前

MAYDAY五月天【終於結束的起點 Beginning of the End】-...

相信音樂BinMusic

觀看次數：86萬・5 個月前

于文文《體面》動態歌詞版【前任3:再見前任 插曲】

PIGFISH Music Channel

觀看次數：3,443萬・3 個月前

西洋流行音樂電台24小時不中斷 | POP Music 24/7...

Top Music KKBOX

2,002 人正在觀看

直播中

柯文哲 我的故事 (全)~柯P來輔大

fonzae

觀看次數：7.9萬・1 個月前

田馥甄 - 愛了很久的朋友 (lyrics)

XIANGLIN HUANG

觀看次數：12萬・1 個月前

Lebron James 'Long Live The King' Motivational Workout

basketballprosworkouts

觀看次數：638萬・11 個月前

PROFESSOR VS CONMAN

Tom "Conman" Connors

觀看次數：61萬・2 週前

Michael Jordan Crazy Tomahawk Leaner Dunk vs...

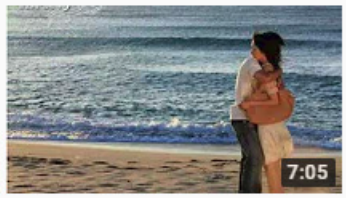
YouuToobe

觀看次數：239萬・4 個月前

Federer's Historical French Open Title - Best Points

Valzy

觀看次數：5.5萬・2 天前



海角七號(野玫瑰） 范逸臣&中孝介

De Moore

觀看次數：69萬・9 年前

'Still KD' Kevin Durant Workout With Steve Nash...

basketballprosworkouts

觀看次數：15萬・6 個月前

2018新歌排行榜（2018最新歌曲,歌曲排行榜2018）201...

Jean Huang

觀看次數：478萬・4 個月前

政大@我在台灣留學的大學 NCCU My university during...

Julie Flower

觀看次數：2.6萬・3 週前

Golden State Warriors vs Houston Rockets Full Game...

Ximo Pierto

觀看次數：320・1 小時前

Mayday五月天[步步Step by Step]MV官方高音質HD版-電...

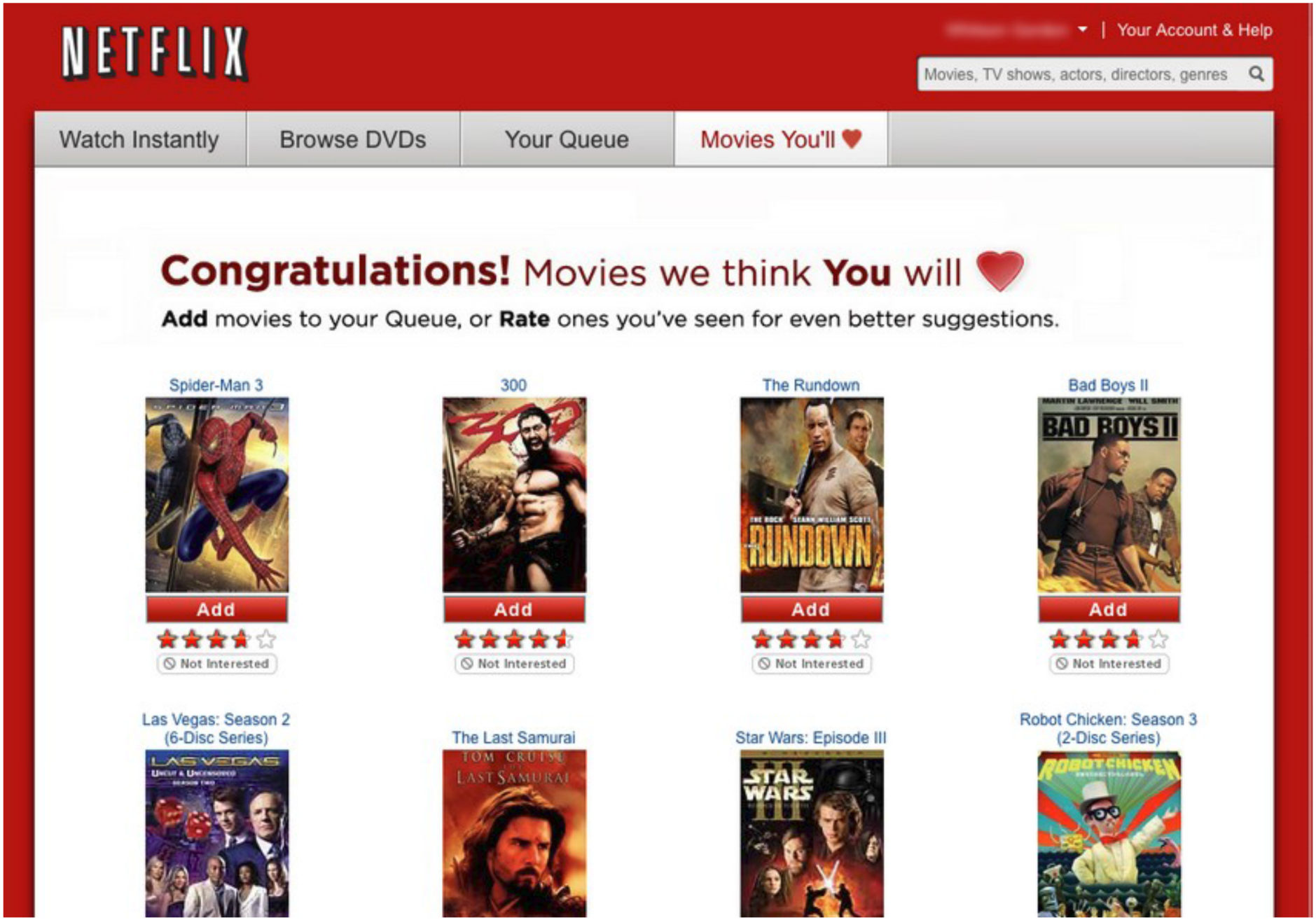
相信音樂BinMusic

觀看次數：3,127萬・4 年前

司马懿功阻曹叡弑母

China Zone – 大军师司马懿之...

觀看次數：2.2萬・2 個月前


NETFLIX
Your Account & Help
Movies, TV shows, actors, directors, genres
Watch Instantly
Browse DVDs
Your Queue
Movies You'll
Congratulations! Movies we think You will
Add movies to your Queue, or Rate ones you've seen for even better suggestions.
Spider-Man 3
300
The Rundown
Bad Boys II
Add
Not Interested
Las Vegas: Season 2 (6-Disc Series)
The Last Samurai
Star Wars: Episode III
Robot Chicken: Season 3 (2-Disc Series)

facebook

You are surrounded by recommendations

朋友推薦

塗鴉牆貼文推薦

粉絲頁推薦

打卡地點推薦

遊戲推薦

社團推薦

廣告推薦

Recommender Systems (RecSys)

- RS analyzes **patterns of user interest** in items to provide personalized recommendations
  - Many users view the same movie and each user is likely to view numerous different movies
  - Huge volume of data arise from user feedbacks that can be analyzed to provide recommendations
- Goal of RecSys: predict the rating or preference that user would give to an item
  - RecSys is very useful for commercial items such as movies, music, books, apps, and TV shows

Recommendations

Search

Recommendations

Items

Products, web sites, blogs, news items, …

Types of Recommendations

- Editorial and hand curated
  - List of favorites
  - Lists of “essential” items
- Simple aggregates
  - Top 10, Most Popular, Recent Uploads
- Tailored to individual users ⟵ Our Target!
  - Amazon, Netflix, YouTube, Instagram, Facebook …

Key Problems

- (1) Gathering "known" ratings for matrix
  - How to collect the data about user-item interactions?
- (2) Predicting unknown ratings from known ones
  - Mainly interested in high unknown ratings
    - Not interested in what you don't like but what you like
- (3) Evaluating extrapolation methods
  - How to measure success/performance of recommendation methods

Gathering Ratings

- Explicit
  - Ask people to rate items
    - Do not work well in practice -- people can't be bothered
  - Crowdsourcing: pay people to label items
  - Incentive (e.g., discounts, coupons) in the websites
- Implicit
  - Learn ratings from user actions
    - E.g., purchase implies high ratings
    - E.g., browsing history (page view)
    - E.g., clicks

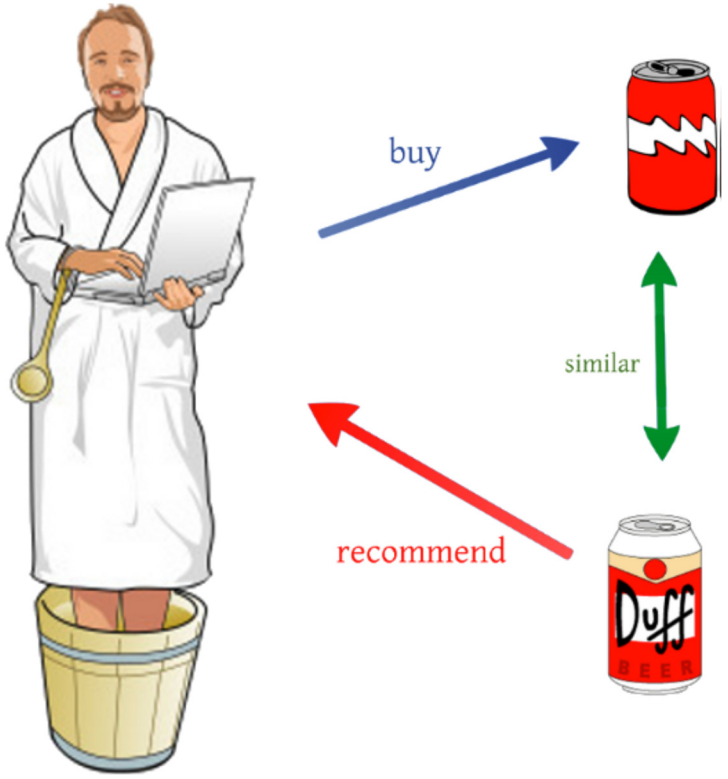
Predicting Ratings

- Key problem: Rating matrix R is **sparse**
  - Most people have not rated most items
  - Cold start
    - New items have no ratings
    - New users have no history
- Three approaches to recommender systems:
  - 1) Content-based Recommendation (CR)
  - 2) Collaborative Filtering (CF)
  - 3) Latent Factor Model (e.g., MF, FM)

Content-based Recommendation

CR

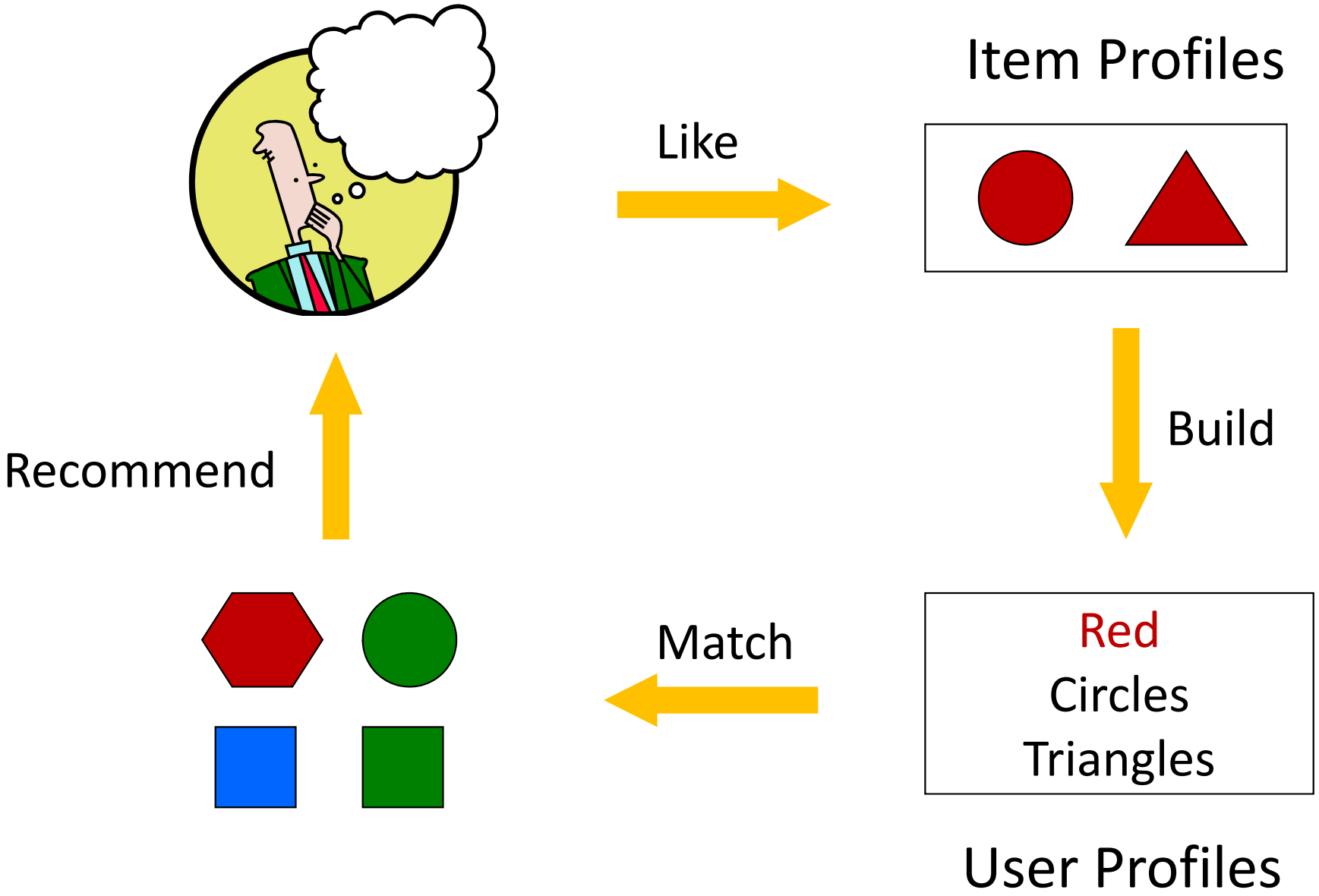
- Main idea: Recommend items to customer u similar to previous items rated highly by u
- E.g., Movie recommendations
  - Recommend movies with the same actor(s), director, genre, etc.
- E.g., Websites, blogs, news
  - Recommend other sites with “similar” content



seafood

CR

Plan of Actions



CR

Item Profiles

- For each item, create an **item profile**
- Profile is a vector of features
  - **Movies:** author, title, actor, director,…
  - **Text:** Set of “important” words in document
- How to pick important features?
  - Usual heuristic from text mining is **TF-IDF**
    (Term Frequency × Inverse Document Frequency)
    - Term → Feature
    - Document → Item

CR

User Profiles and Prediction

- User profile possibilities:
  - Average of rated item profiles
  - Weighted by difference from average rating for item
- Prediction heuristic:

  Cosine similarity of user and item profiles
  - Given user profile $\boldsymbol{x_u}$ and item profile $\boldsymbol{x_i}$, estimate

$$cos(\boldsymbol{x_u}, \boldsymbol{x_i}) = \frac{\boldsymbol{x_u} \cdot \boldsymbol{x_i}}{\|\boldsymbol{x_u}\| \|\boldsymbol{x_i}\|}$$

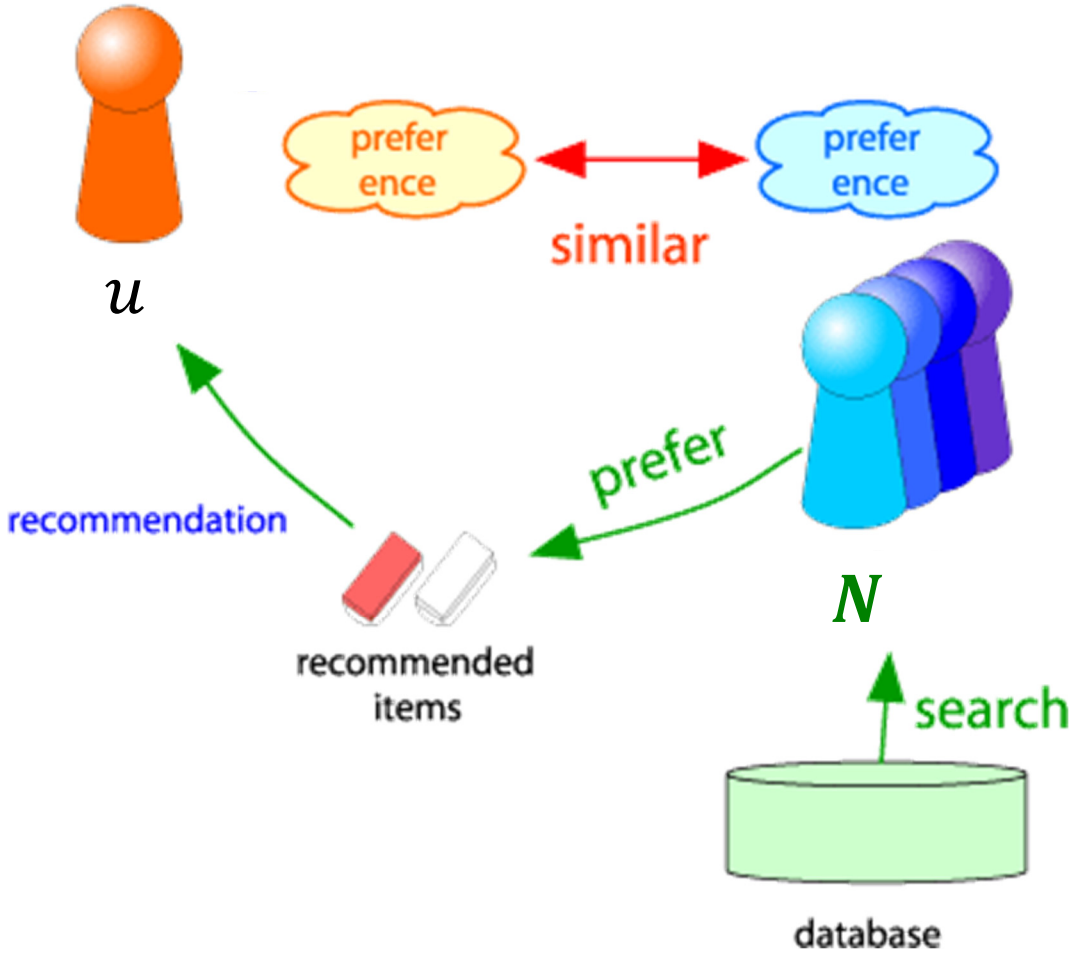
Pros/Cons of CR

CR

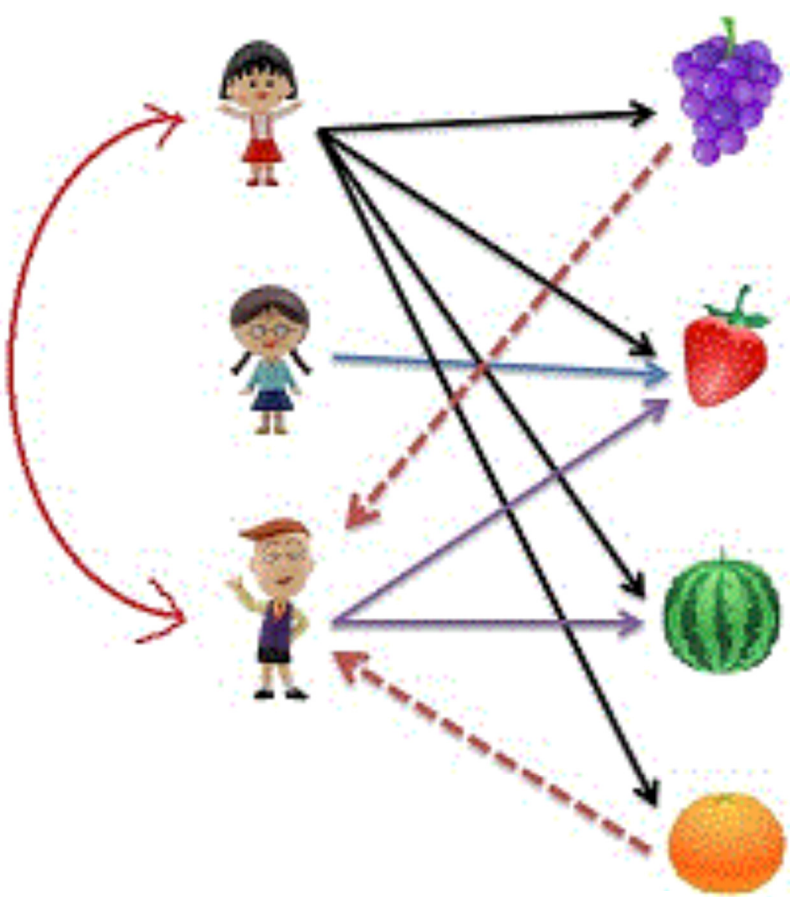
- +: No need for data on other users
  - No cold-start or sparsity problems
- +: Able to recommend to users with unique tastes
- +: Able to recommend new & unpopular items
  - No first-rater problem
- +: Able to provide explanations
  - Can provide explanations of recommended items by listing content-features that caused an item to be recommended
- –: Finding the appropriate features is hard
  - E.g. images, movies, music
- –: Recommendations for new users
  - How to build a user profile?
- –: Overspecialization
  - Never recommends items outside user's content profile
  - People might have multiple interests
  - Unable to exploit quality judgments of other users

Collaborative Filtering (CF)

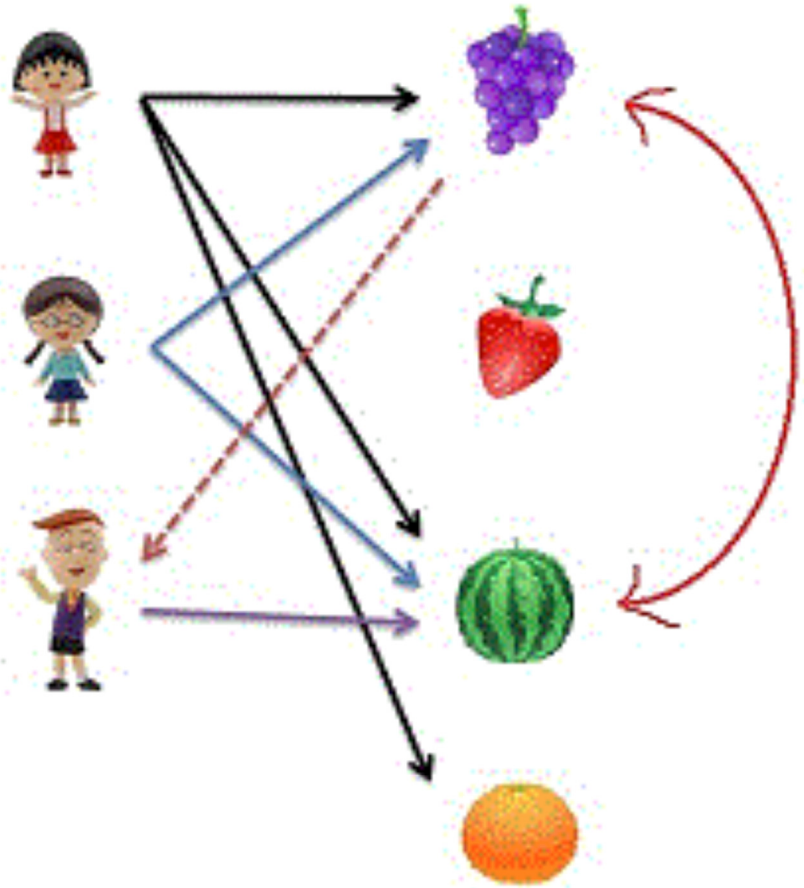
- Consider user u
- Find set N of other users whose ratings are similar to u's ratings
- Estimate u's ratings based on ratings of users in N



Collaborative Filtering (CF)



User-based CF



Item-based CF

User-based CF: Finding **Similar** Users

CF

- Let r_u be the vector of user u's ratings
- Cosine similarity measure
  - $\text{sim}(u, v) = \cos(r_u, r_v) = \frac{r_u \cdot r_v}{\|r_u\| \|r_v\|}$
  - **Problem:** treat missing ratings as "negative" (dissimilar)

$\boldsymbol{r_u}$ = {1, 0, 0, 1, 3}
$\boldsymbol{r_v}$ = {1, 0, 2, 2, 0}

- Pearson Correlation Coefficient
  - S_{uv}: items rated by both users u and v

$$sim(u,v) = \frac{\sum_{i \in S_{uv}} (r_{ui} - \overline{r_u})(r_{vi} - \overline{r_v})}{\sqrt{\sum_{i \in S_{uv}} (r_{ui} - \overline{r_u})^2} \sqrt{\sum_{i \in S_{uv}} (r_{vi} - \overline{r_v})^2}}$$

$\overline{r_u}, \overline{r_v}$: the average ratings of users u and v

Cosine Similarity Metric

	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: **sim(*A*, *B*) > sim(*A*, *C*)**
- **Cosine similarity:** 0.380 > 0.322
  - Considers missing ratings as “negative”
  - **Solution: subtract the (row) mean**

sim A,B vs. A,C:
0.092 > -0.559

	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

Rating Predictions

From similarity metric to recommendations:

- Let r_u be the vector of user u's ratings
- Let N be the set of k users most similar to u who have rated item i
- Prediction for item i of *user* u:
  - $r_{ui} = \frac{1}{k} \sum_{v \in N} r_{vi}$
  - Or even better: $r_{ui} = \frac{\sum_{v \in N} s_{uv} \cdot r_{vi}}{\sum_{v \in N} s_{uv}}$

Shorthand:
$s_{xy} = sim(x, y)$

Item-Item Collaborative Filtering

CF

- So far: User-User Collaborative Filtering
- Another view: Item-Item CF
  - 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 CF model

$$r_{ui} = \frac{\sum_{j \in N(i;u)} s_{ij} \cdot r_{uj}}{\sum_{j \in N(i;u)} s_{ij}}$$

s_{ij}: similarity of items i and j
r_{uj}: rating of user u on item j
$N(i;u)$: set of items rated by u similar to i

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

users

movies	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	

- unknown rating

- rating between 1 to 5

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

users

movies

	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	

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

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

Compute similarity weights:

$s_{1,3}$=0.41, $s_{1,6}$=0.59

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**

Neighbor selection:

Identify movies similar to movie **1**, **rated by user 5**

Here we use Pearson correlation as similarity:

1) Subtract mean rating m_i from each movie ***i***

$m_1 = (1+3+5+5+4)/5 = 3.6$

row 1: *[-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0.4, 0]*

2) Compute cosine similarities between rows

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

Compute similarity weights:

$s_{1,3}$=0.41, $s_{1,6}$=0.59

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**

Predict by taking weighted average:

$r_{1.5}$ = (0.41*2 + 0.59*3) / (0.41+0.59) = 2.6

$$r_{ui} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{uj}}{\sum_{j \in N(i;x)} s_{ij}}$$

CF Common Practice

- Define similarity s_{ij} of items i and j
- Select k nearest neighbors $N(i;u)$
  - Items most similar to i that were rated by u
- Estimate rating r_{ui} as the weighted average:

$$r_{ui} = \frac{\sum_{j \in N(i;u)} s_{ij} \cdot r_{uj}}{\sum_{j \in N(i;u)} s_{ij}}$$

$$r_{ui} = b_{ui} + \frac{\sum_{j \in N(i;u)} s_{ij} \cdot \left(r_{uj} - b_{uj}\right)}{\sum_{j \in N(i;u)} s_{ij}}$$

baseline estimate for r_{ui}

$$b_{ui} = \mu + b_u + b_i$$

- μ = overall mean item rating
- b_x = rating deviation of user x
  = (*avg. rating of user x*) – μ
- b_i = rating deviation of item i
  = (*avg. rating of item i*) – μ

Item-based CF vs. User-based CF

CF

	Avatar	LOTR	Matrix	Pirates
Alice	1		0.8	
Bob		0.5		0.3
Carol	0.9		1	0.8
David			1	0.4

- In practice, it has been observed that item-based CF often works better than user-based CF
- Why? Items are simpler, users have multiple tastes

Two similar items (e.g., “Action” movies) tend to have the same group of users

Two similar users could have different tastes (e.g., {Action, Drama, Dramatic}, {Action, Historical, Comedy}

Pros/Cons of CF

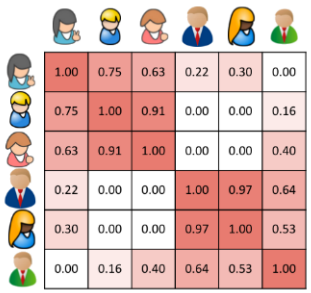
CF

- + Work for any kind of item
  - No feature selection needed
- - Cold Start:
  - Need enough users in the system to find a match
- - Sparsity:
  - The user/ratings matrix is sparse
  - Hard to find users that have rated the same items
- - First rater:
  - Cannot recommend an item that has not been previously rated
  - New items, esoteric (極少數的) rated items
- - Popularity bias:
  - Tend to recommend popular items
  - Cannot recommend items to someone with unique taste

Hybrid Methods

Hybrid Method	Description
Weighted	The scores (or votes) of several recommendation techniques are combined (e.g. average) together to produce a single recommendation. Another approach is to use Linear model.
Switching	The system switches between recommendation techniques depending on the current situation (e.g. cold-start)
Mixed	Recommendations from several different recommenders are presented at the same time
Feature Combining	Features from different recommendation data sources are thrown together into a single recommendation algorithm
Cascade	One recommender refines the recommendations given by another
Feature Argumentation	Output from one technique is used as an input feature to another

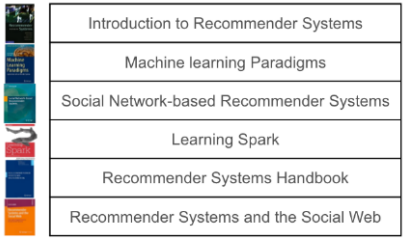
USER-BASED COLLABORATIVE FILTERING



1.00	0.75	0.63	0.22	0.30	0.00
0.75	1.00	0.91	0.00	0.00	0.16
0.63	0.91	1.00	0.00	0.00	0.40
0.22	0.00	0.00	1.00	0.97	0.64
0.30	0.00	0.00	0.97	1.00	0.53
0.00	0.16	0.40	0.64	0.53	1.00

INPUT

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



Introduction to Recommender Systems
Machine learning Paradigms
Social Network-based Recommender Systems
Learning Spark
Recommender Systems Handbook
Recommender Systems and the Social Web

ITEM-BASED COLLABORATIVE FILTERING

1.00	0.27	0.79	0.32	0.98	0.00
0.27	1.00	0.00	0.00	0.34	0.65
0.79	0.00	1.00	0.69	0.71	0.18
0.32	0.00	0.69	1.00	0.32	0.49
0.98	0.34	0.71	0.32	1.00	0.00
0.00	0.65	0.18	0.49	0.00	1.00

CONTENT-BASED FILTERING

1.00	0.00	0.58	0.00	0.67	0.58
0.00	1.00	0.00	0.41	0.00	0.00
0.58	0.00	1.00	0.00	0.58	0.75
0.00	0.41	0.00	1.00	0.00	0.00
0.67	0.00	0.58	0.00	1.00	0.58
0.58	0.00	0.75	0.00	0.58	1.00

HYBRID

OUTPUTS