Lecture 20: Recommendation Systems

COMP90049
Knowledge Technology

Sarah Erfani and Vinh Nguyen, CIS

Semester 2, 2018

From Search to Recommendation

“The Web, they say, is leaving the era of search and entering one of discovery.

What’s the difference? Search is what you do when you’re looking for something.

Discovery is when something wonderful that you didn’t know existed, or didn’t know how to ask for, finds you.”

Jeffrey O'Brien, Fortune Magazine, The race to create a 'smart' Google

Information overload

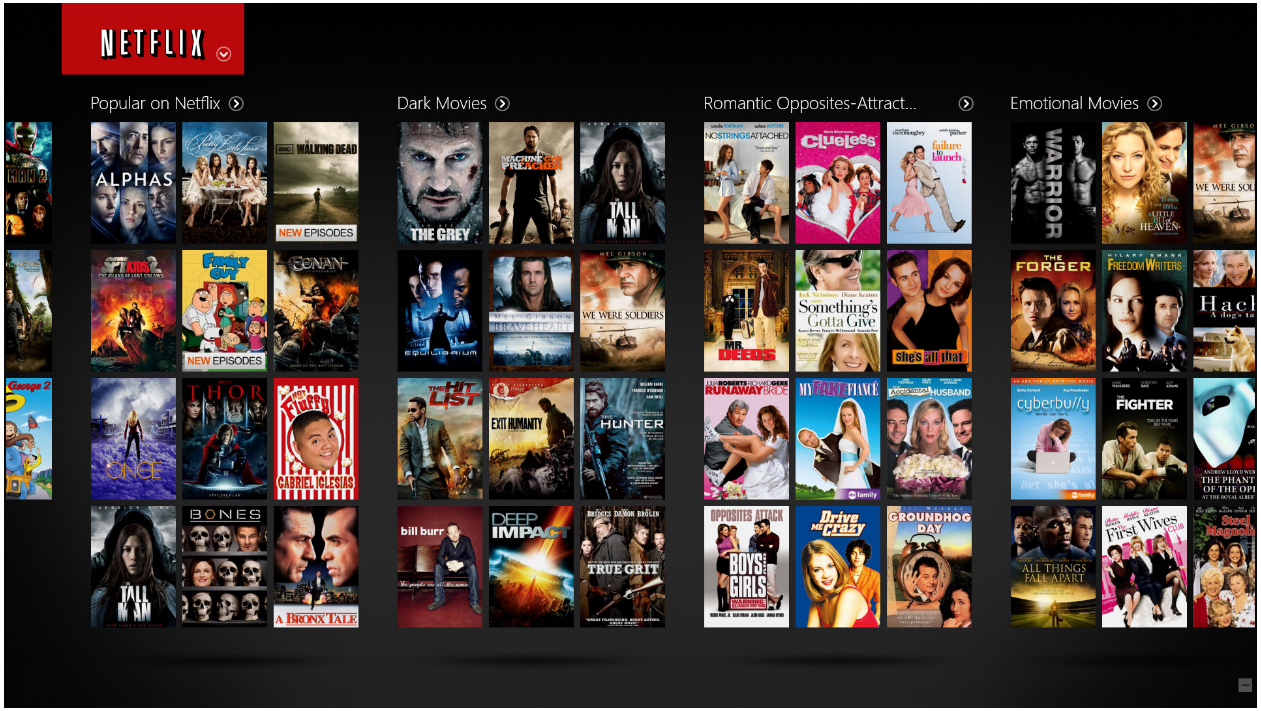
- Consumers are overloaded with products and services
- We only need and like to consume a few

Search vs. Recommendation

- Search: when users know what to look for

Search vs. Recommendation

- Recommendation:
  - – Relevant items that users don't know
    - how to look for
    - Its existence



Examples

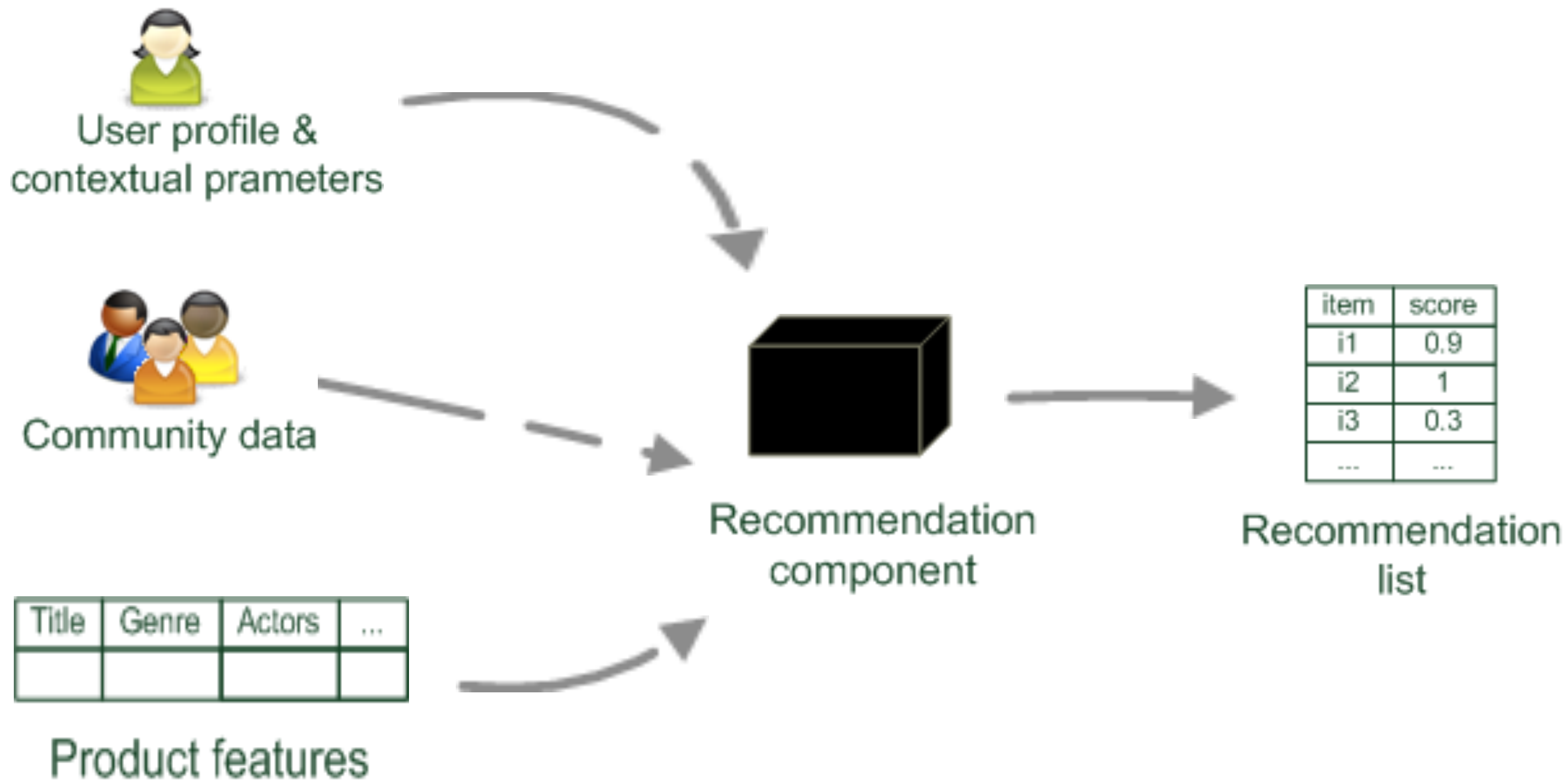
- Movie recommendation (Netflix)
- Related product recommendation (Amazon)
- Social recommendation (Facebook, LinkedIn)
- News content recommendation (Yahoo, Google News)

The value of recommendations

- Netflix: 2/3 of the movies watched are recommended
- Google News: recommendations generate 38% more click-throughs
- Amazon: 35% sales from recommendations

Problem Formulation

- Estimate a utility function to predict how a user will like an item



Operational goals

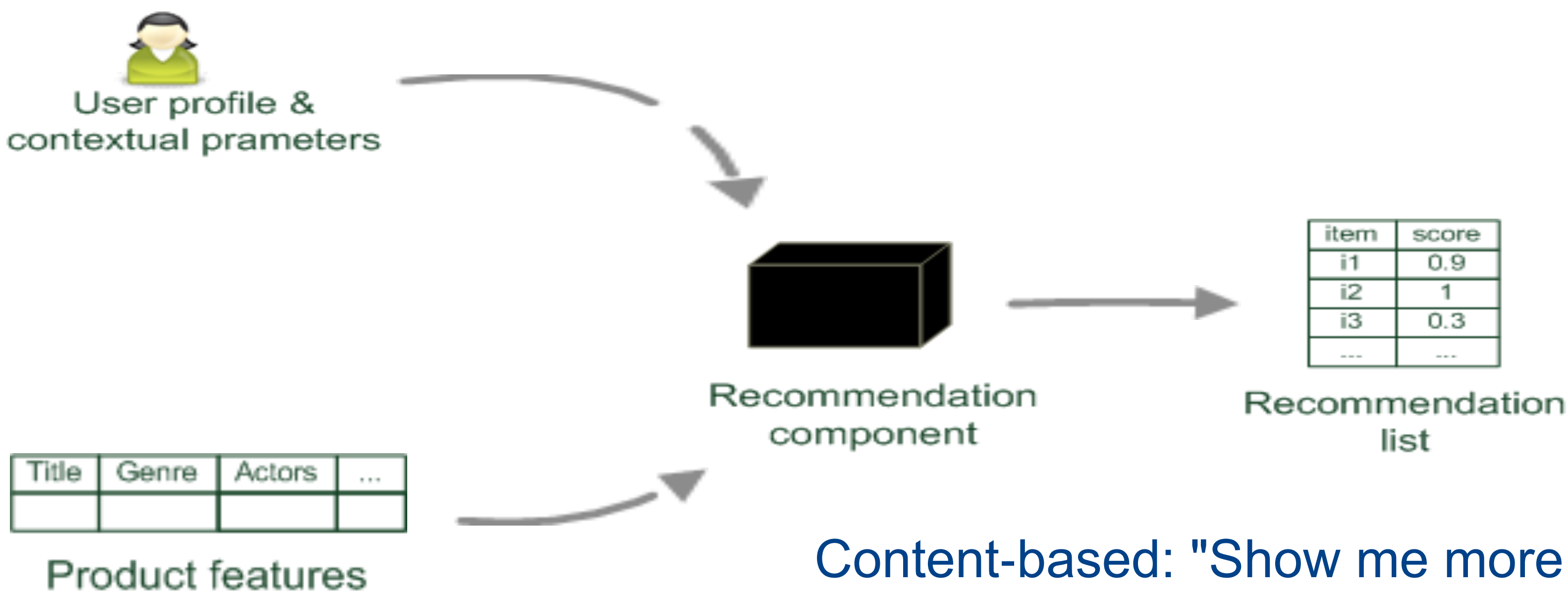
- **Relevance:** Users are more likely to consume items they find *Relevance*
- **Novelty:** Recommender systems are truly helpful when the recommended item is something that the user has *not seen in the past*
- **Serendipity:** the items recommended are somewhat *unexpected*
  - Recommend a new Indian restaurant opens in a neighbourhood to a user who normally eats Indian food is novel but not necessarily serendipitous.
  - Recommend Ethiopian food, and it was unknown to the user that such food might appeal to her, then the recommendation is serendipitous.
- **Increasing recommendation diversity:** Diversity has the benefit of ensuring that the user does not get bored by repeated recommendation of similar items

Recommendation Systems Strategies

- Content-based recommendation
- Collaborative filtering
  - User-based CF
  - Item-based CF

1. Content-based recommendation

- Recommendations are based on information on the content of items
- Uses a machine learning algorithm to induce a profile of the users preferences from examples based on a featural description of content.



Content-based: "Show me more of the same what I've liked"

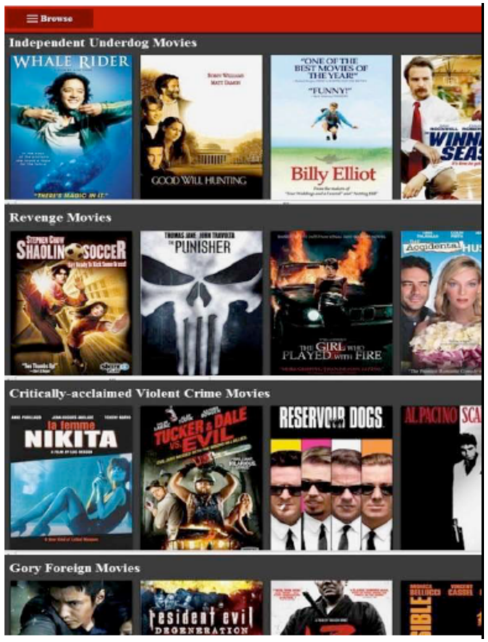
1. Content-based recommendation

- **What do we need:**
  - Some information about the available items such as the genre ("content")
  - Some sort of *user profile* describing what the user likes (the preferences): items users liked in the past or explicit interests that he defined

- **The task:**
  - Learn user preferences
  - Locate/recommend items that are "similar" to the user preferences

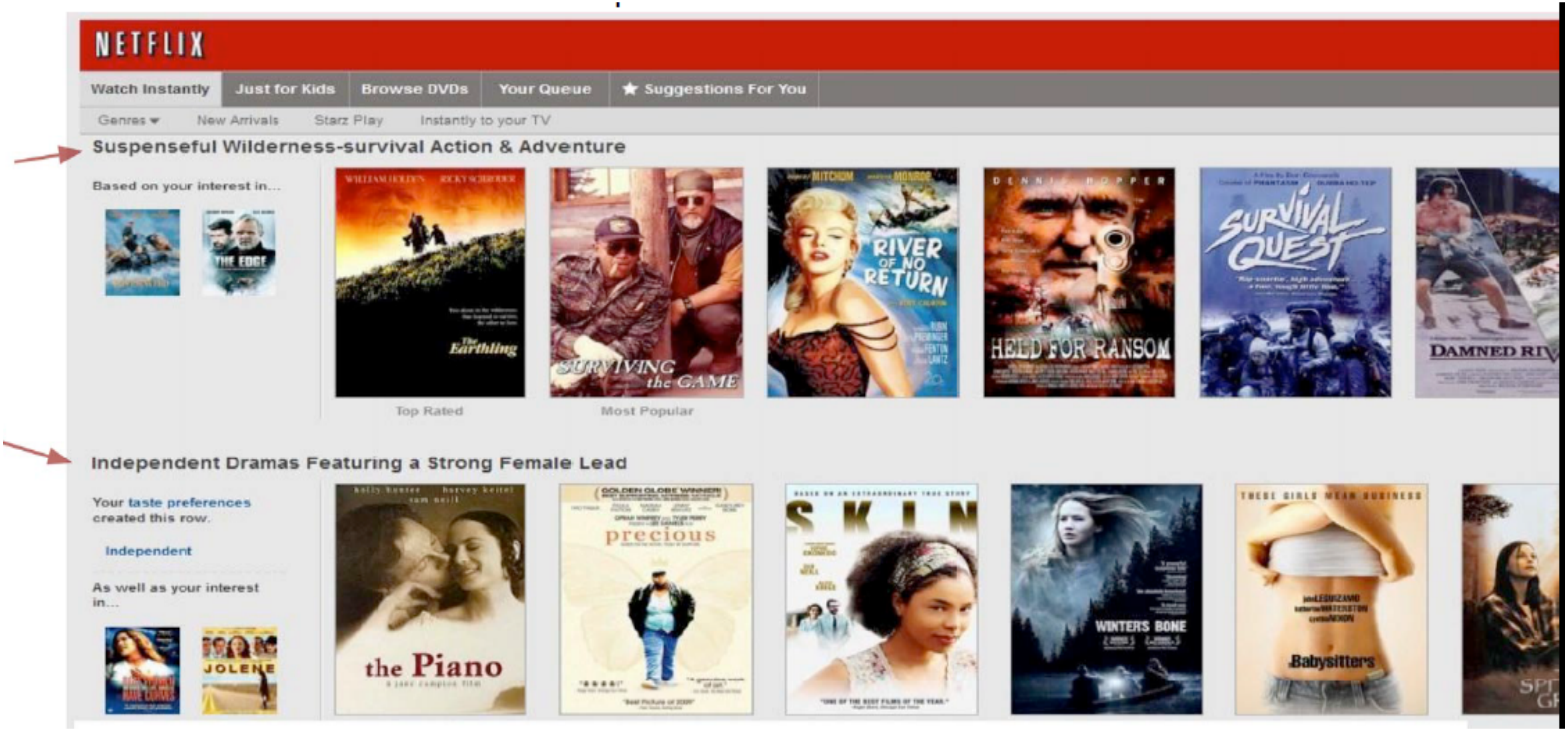
What is the "content"?

- Explicit attributes or characteristics, e.g., for a movie:
  - Genre: Action / adventure
  - Actors: Bruce Willis
  - Year: 1995
- Textual content, e.g., for a book:
  - Title,
  - Description,
  - Table of content



1. Content-Based Recommendations

- The recommended items for a user are based on the *profile* built up by analysing the content of the items the user has liked in the past



1. Content-Based Recommendations

- Suitable for text-based products (web pages, books)
- Items are “described” by their features (e.g. keywords)
- Users are described by the keywords in the items they bought
- Recommendations based on the match between the content (item keywords) and user keywords

Item profiles

- *ItemProfile(s):*= profile of item s

$$w_s = (w_{1s}, . . . , w_{ks})$$

weight w_{is} measures the 'Importance" (or "informativeness") of word k_i in item s

- *term frequency/inverse document frequency (TF-IDF)* is a popular weighting technique in IR

User profiles

- *UserProfile(c):*= profile of user c

$$w_c = (w_{1c}, \ldots, w_{kc})$$

- a vector of weights, where w_{ic} denotes the importance of keyword k_i to user c
- profiles are obtained by analysing the content of the previous items using keyword analysis techniques

Similarity Measurements

- Utility function *u(c,s)* usually represented by some scoring heuristic defined in terms of vectors, such as the cosine similarity measure.

$$u(c,s) = cos(w_c, w_s) = \frac{w_c \times w_s}{||w_s|| \, ||w_c||} = \frac{\sum_{i=1}^{K} w_{ic} w_{is}}{\sum_{i=1}^{K} w_{ic}^2 \sum_{i=1}^{K} w_{is}^2}$$

Example: How to compute Recommendations of books based only on their title?

- A customer buys the book: *Building data mining applications for CRM*

- 7 Books are possible candidates for a recommendation:
  - *Accelerating Customer Relationships: Using CRM and Relationship Technologies*
  - *Mastering Data Mining: The Art and Science of Customer Relationship Management*
  - *Data Mining Your Website*
  - *Introduction to marketing*
  - *Consumer behaviour*
  - *Marketing research, a handbook*
  - *Customer knowledge management*

Frequency count

COUNT	a	Accelerating	and	applications	art	behavior	Building	Consumer	CRM	customer	data	for	Handbook	Introduction	Knowledge	Management	Marketing	Mastering	mining	of	relationship	Research	science	technology	the	to	using	website	your
Building data mining applications for CRM				1			1		1		1	1							1										
Accelerating customer relationships: using CRM and relationship technologies		1	1						1	1											2			1			1		
Mastering Data Mining: the art and science of Customer Relationship Management			1		1					1	1					1		1	1	1	1		1		1				
Data Mining your website											1								1									1	1
Introduction to Marketing														1			1									1			
Consumer behavior						1		1																					
Marketing Research: a Handbook	1												1				1					1							
Customer Knowledge Management										1					1	1													

TF-IDF

TFIDF Normed Vectors	a	Accelerating	and	applications	art	behavior	Building	Consumer	CRM	customer	data	for	Handbook	Introduction	Knowledge	Management	Marketing	Mastering	mining	of	relationship	Research	science	technology	the	to	using	website	your
Building data mining applications for CRM				0.502			0.502		0.344		0.251	0.502							0.251										
Accelerating customer relationships: using CRM and relationship technologies		0.432	0.296						0.296	0.216											0.468			0.432			0.432		
Mastering Data Mining: the art and science of Customer Relationship Management			0.256		0.374					0.187	0.187					0.256		0.374	0.187	0.374	0.256		0.374		0.374				
Data Mining your website											0.316								0.316									0.632	0.632
Introduction to Marketing														0.636			0.436									0.636			
Consumer behavior						0.707		0.707																					
Marketing Research: a Handbook	0.537												0.537				0.368					0.537							
Customer Knowledge Management										0.381					0.736	0.522													

Recommendation

- **Computes distances** between the book that the user has bought & all others
- Recommends the **closest** books:
  - #1: Data Mining Your Website
  - #2: Accelerating Customer Relationships: Using CRM and Relationship Technologies
  - #3: Mastering Data Mining: The Art and Science of Customer Relationship Management

Advantages/disadvantages of content based approaches

- Advantages:
  - Can recommend new items

- Disadvantages
  - Feature extraction can be difficult (music, movies)

2. Collaborative Filtering

The task of predicting user preferences on new items (filtering) by collecting taste information from many users (collaborative).

2. Collaborative Filtering

The task of predicting user preferences on new items (filtering) by collecting taste information from many users (collaborative).

- **User-based models:** Similar users have similar ratings on the same item.
  - If Alice and Bob have rated movies in a similar way in the past, then one can use Alice's observed ratings on the movie *Terminator* to predict Bob's unobserved ratings on this movie.

- **Item-based models:** Similar items are rated in a similar way by the same user
  - Bob's ratings on similar science fiction movies like *Alien* and *Predator* can be used to predict his rating on *Terminator*.

Problem setting

Each user has expressed an opinion for some items:

- Explicit opinion: rating score
- Implicit: purchase records or listen to tracks

2			4	5	
5		4			1
		5		2	
	1		5		4
		4		2	
4	5		1		

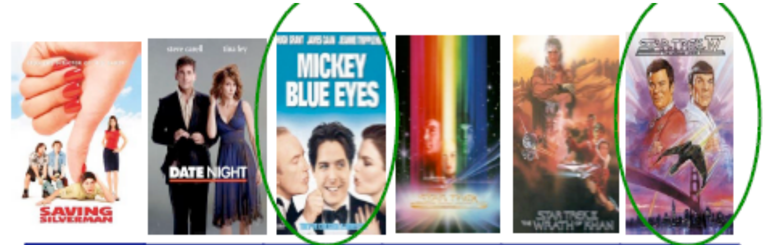
Problem setting

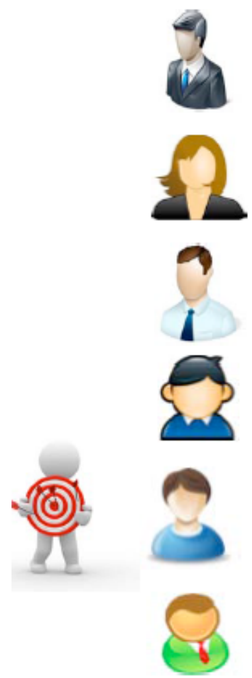
Target (or active) user for whom the CF recommendation task is performed.

2			4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

User based collaborative filtering

1. **Identify** set of **items** rated by the target user



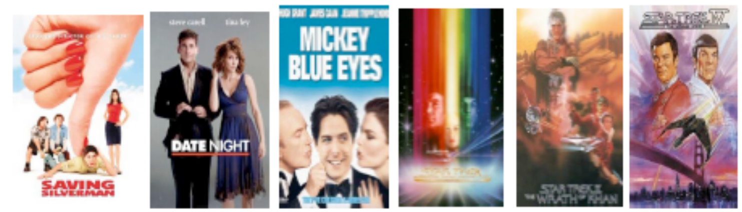


2			4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

User based collaborative filtering

1. **Identify** set of **items** rated by the target user

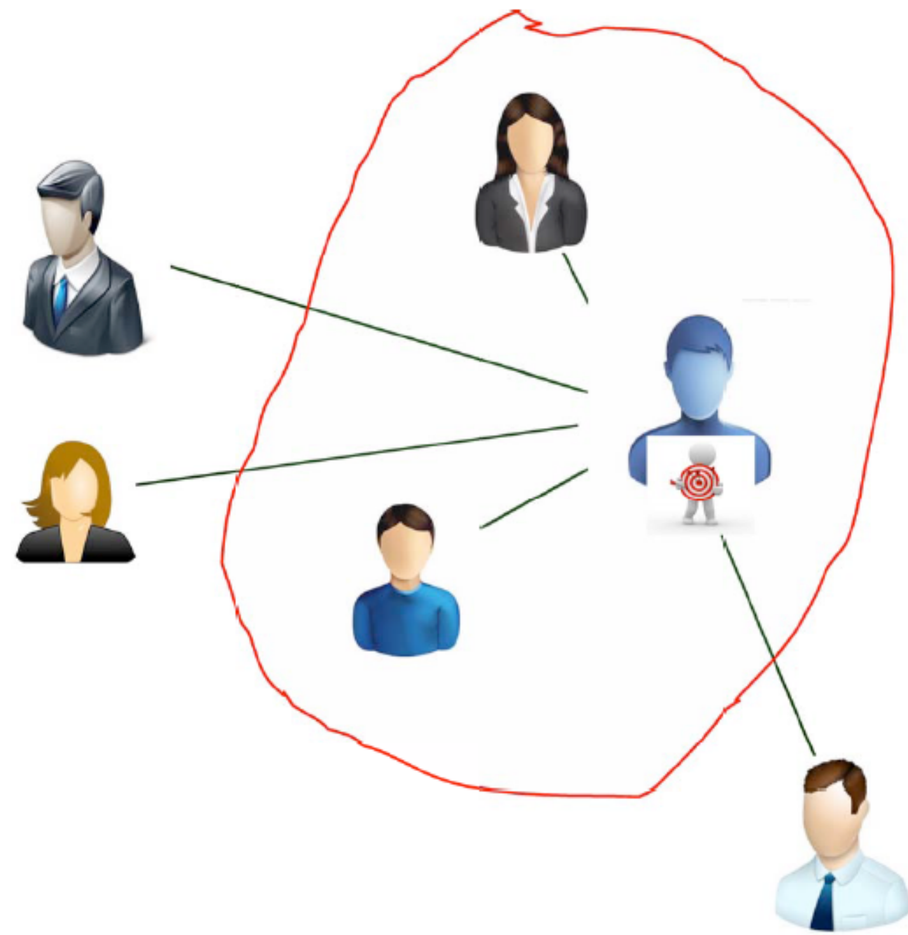
2. **Identify** which other **users** rated 1+ items in this set (neighborhood formation)



2			4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

User similarity

1. **Identify** set of **items** rated by the target user
2. **Identify** which other **users** rated 1+ items in this set (*neighborhood formation*)
3. **Compute** how **similar** each neighbor is to the target user (*similarity function*)
4. **Select k most similar** neighbors



Computing user similarity

- Rating matrix R: $r_{u,k}$ - rating by user u for item k
- *Pearson correlation* between users u & v

$$\mathrm{Sim}(u,v) = \mathrm{Pearson}(u,v) = \frac{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u) \cdot (r_{vk} - \mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v} (r_{uk} - \mu_u)^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} (r_{vk} - \mu_v)^2}}$$

$$\mu_u = \frac{\sum_{k \in I_u} r_{uk}}{|I_u|}$$ Mean rating of user u

I_u Set of items rated by user u

Strictly speaking, the traditional definition of Pearson(u,v) mandates that the values of μ_u and μ_v should be computed *only* over the items that are rated *both* by users u and v.

User based collaborative filtering

5. **Predict** ratings for the target user's unrated items (prediction function)

Rating of user u for item j:

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v \in P_u(j)} \text{Sim}(u, v) \cdot (r_{vj} - \mu_v)}{\sum_{v \in P_u(j)} |\text{Sim}(u, v)|}$$

$P_u(j)$ Set of nearest users of user u who rated item j

6. **Recommend** to the user the top N products based on predicted rating

Example: user-based CF

- Predict r_{31} and r_{36}

Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6	Mean Rating	Cosine$(i,3)$ (user-user)	Pearson$(i,3)$ (user-user)
1	7	6	7	4	5	4	5.5	0.956	0.894
2	6	7	?	4	3	4	4.8	0.981	0.939
3	?	3	3	1	1	?	2	1.0	1.0
4	1	2	2	3	3	4	2.5	0.789	-1.0
5	1	?	1	2	3	3	2	0.645	-0.817

$$\text{Cosine}(1,3) = \frac{6*3+7*3+4*1+5*1}{\sqrt{6^2+7^2+4^2+5^2}\cdot\sqrt{3^2+3^2+1^2+1^2}} = 0.956$$

$$\text{Pearson}(1,3) =$$

$$= \frac{(6-5.5)*(3-2)+(7-5.5)*(3-2)+(4-5.5)*(1-2)+(5-5.5)*(1-2)}{\sqrt{1.5^2+1.5^2+(-1.5)^2+(-0.5)^2}\cdot\sqrt{1^2+1^2+(-1)^2+(-1)^2}}$$

$$= 0.894$$

Prediction

$$\hat{r}_{31} = 2 + \frac{1.5 * 0.894 + 1.2 * 0.939}{0.894 + 0.939} \approx 3.35$$

$$\hat{r}_{36} = 2 + \frac{-1.5 * 0.894 - 0.8 * 0.939}{0.894 + 0.939} \approx 0.86$$

Item based CF

The basic steps:

- Identify set of users who rated the target item i
- Identify which other items (neighbours) were rated
- Compute similarity between each neighbour and target item (*similarity function*)
- Select k most similar neighbours
- Predict ratings for the target item (*prediction function*)

Item similarity

Similarity measure

- Compute the similarity between columns (items)
- Pearson and cosine similarity can be used
- Adjusted cosine measure yields superior results

$$\text{AdjustedCosine}(i,j) = \frac{\sum_{u \in U_i \cap U_j} s_{ui} \cdot s_{uj}}{\sqrt{\sum_{u \in U_i \cap U_j} s_{ui}^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} s_{uj}^2}}$$

$s_{uj} = r_{uj} - \mu_u$ Mean-centred rating

Example of item based CF

- Mean-centred voting matrix

Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6
1	1.5	0.5	1.5	-1.5	-0.5	-1.5
2	1.2	2.2	?	-0.8	-1.8	-0.8
3	?	1	1	-1	-1	?
4	-1.5	-0.5	-0.5	0.5	0.5	1.5
5	-1	?	-1	0	1	1
Cosine$(1, j)$ (item-item)	1	0.735	0.912	-0.848	-0.813	-0.990
Cosine$(6, j)$ (item-item)	-0.990	-0.622	-0.912	0.829	0.730	1

$$\text{AdjustedCosine}(1,3) = \frac{1.5 * 1.5 + (-1.5) * (-0.5) + (-1) * (-1)}{\sqrt{1.5^2 + (-1.5)^2 + (-1)^2} \cdot \sqrt{1.5^2 + (-0.5)^2 + (-1)^2}} = 0.912$$

Item based prediction

Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6
1	1.5	0.5	1.5	-1.5	-0.5	-1.5
2	1.2	2.2	?	-0.8	-1.8	-0.8
3	?	1	1	-1	-1	?
4	-1.5	-0.5	-0.5	0.5	0.5	1.5
5	-1	?	-1	0	1	1
Cosine(1, j) (item-item)	1	0.735	0.912	-0.848	-0.813	-0.990
Cosine(6, j) (item-item)	-0.990	-0.622	-0.912	0.829	0.730	1

$$\hat{r}_{ut} = \frac{\sum_{j \in Q_t(u)} \text{AdjustedCosine}(j,t) \cdot r_{uj}}{\sum_{j \in Q_t(u)} |\text{AdjustedCosine}(j,t)|}$$

$$\hat{r}_{31} = \frac{3 * 0.735 + 3 * 0.912}{0.735 + 0.912} = 3$$

$$\hat{r}_{36} = \frac{1 * 0.829 + 1 * 0.730}{0.829 + 0.730} = 1$$

Challenges

- Many items to choose from
- Very few recommendations to propose
- Few data per user
- No data for new user
- Very large datasets

Performance

- Bottleneck: similarity computation

  Time complexity, highly time consuming with millions of users and items in the database

- Two step process:
  - Offline components/model

    Similarity computation precomputed and stored
  - Online component: prediction on process

References

- Recommendation systems – The text book (chapter 1 & 2)