A Brief Introduction to Recommender Systems

Presenter: Liang Hu

Outline

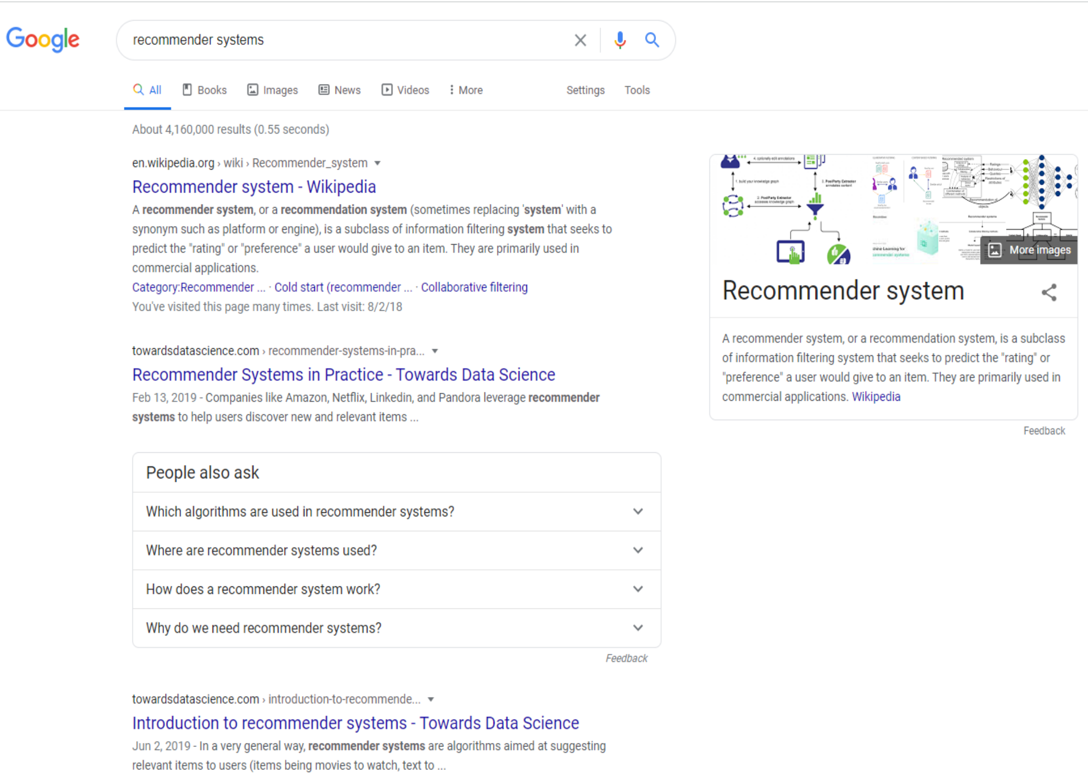
- Recommendation age
- A case study of recommender systems
- The 3C principle to build recommender systems
- Classic recommender systems
- Evaluation metrics for recommender systems

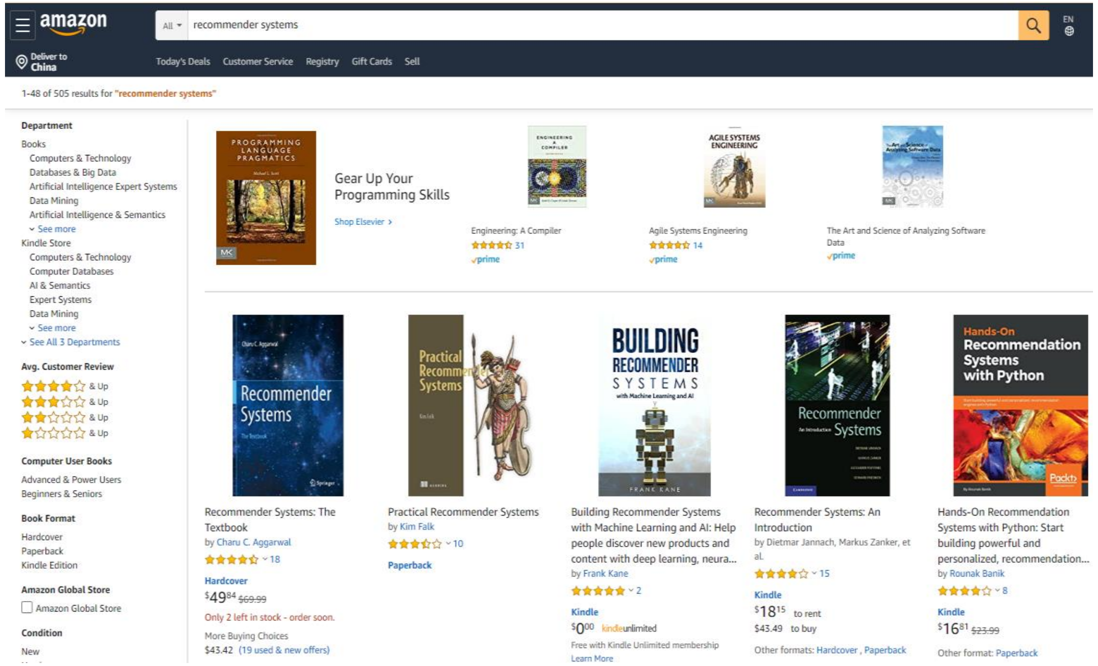
Content

- Recommendation age
- A case study of recommender systems
- The 3C principle to build recommender systems
- Classic recommender systems
- Evaluation metrics for recommender systems

Information age

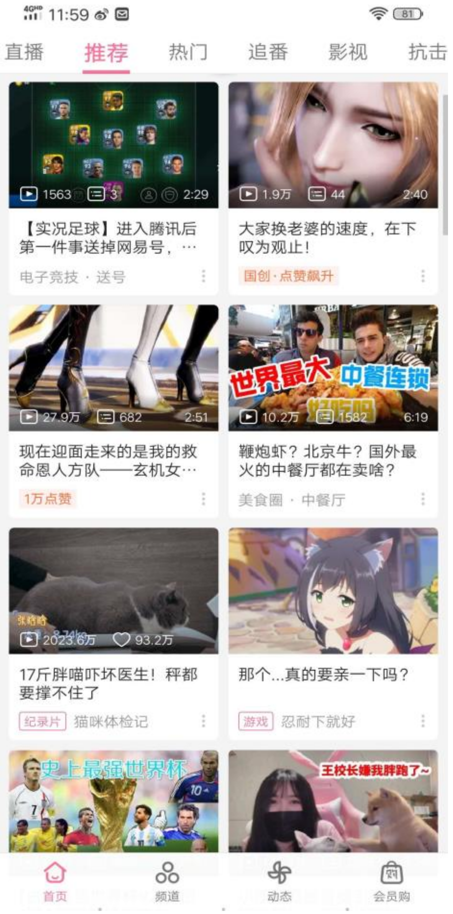
Connecting world with searches





Time changes

Connecting world with clicks



From information age to recommendation age

- Recommender systems (push information) are the evolution of information retrieval systems (pull information).

Information Age → Recommendation Age

Pull mode (IRS):
Query → Matched Results → Manual Filtering

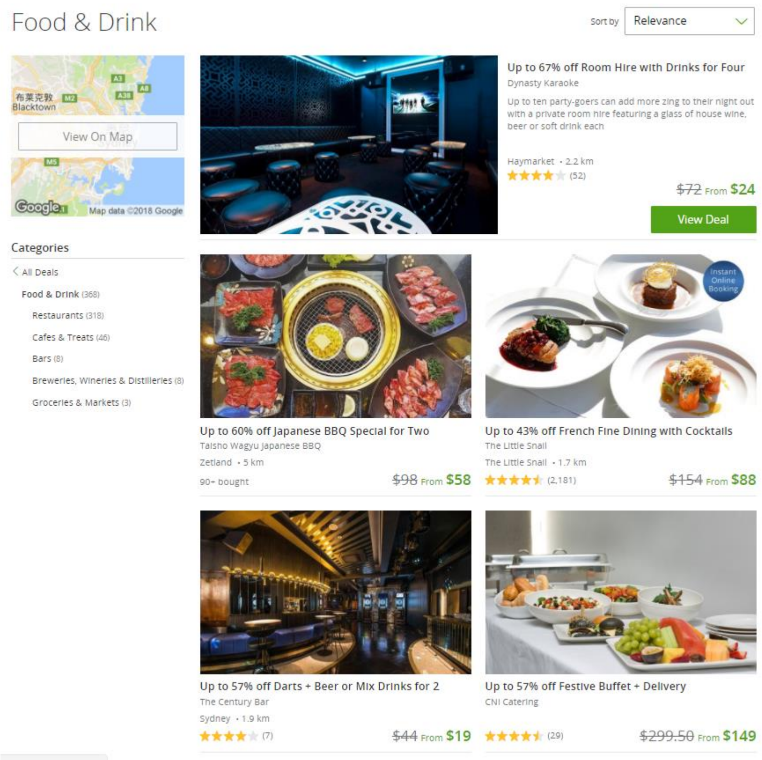
Push mode (RS):
Potential Requirement → Machine Filtering → Recommendation

Anderson, C. (2006). *The long tail: Why the future of business is selling less of more*

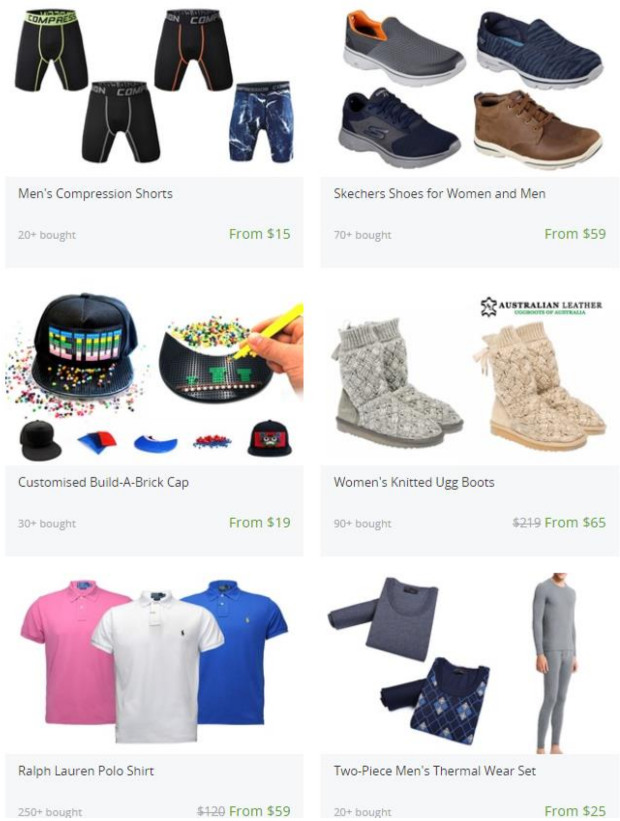
Personalization

Recommender systems have occupied our life

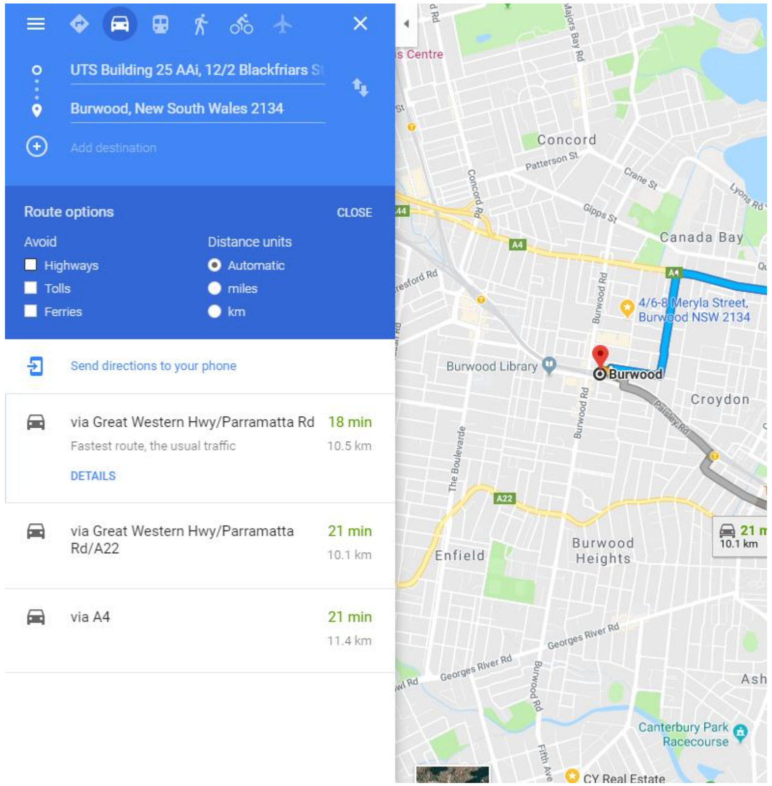
What to eat



Which to dress



Where to go



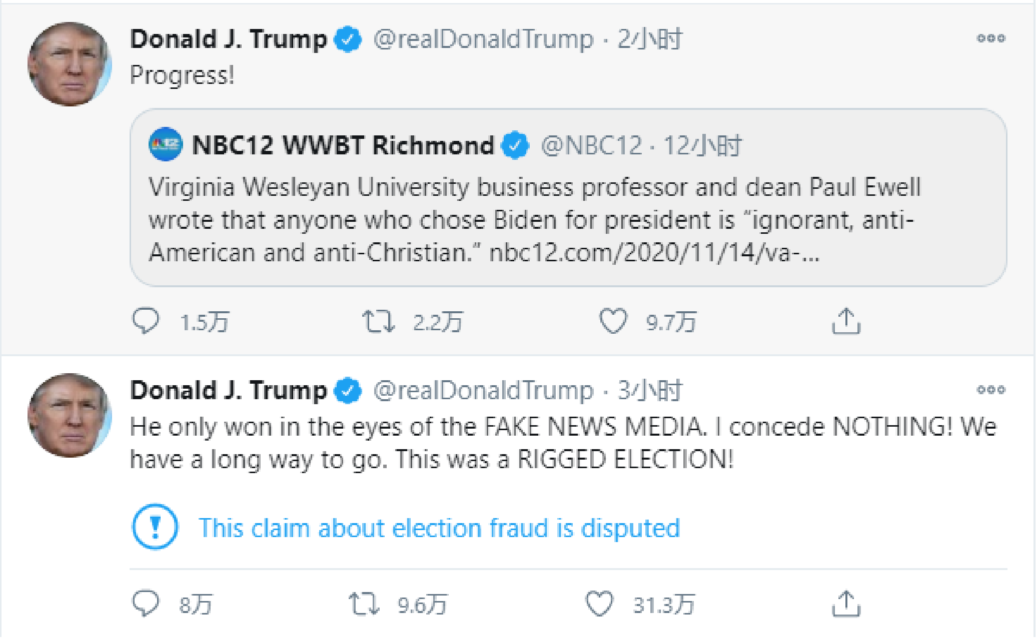
Can RS be used for election?

Data won the U.S. election

Data scientist Rayid Ghani helped persuade voters to reelect President Obama.

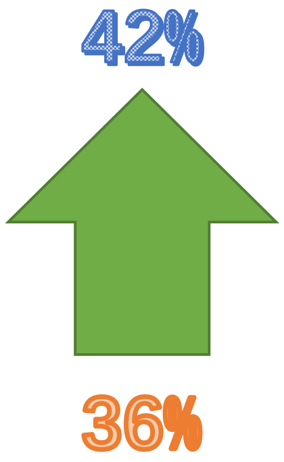
https://www.technologyreview.com/2013/05/29/178302/data-won-the-us-election-now-can-it-save-the-world/
https://site.douban.com/153042/widget/notes/7994379/note/262654369/

Mining public opinions



Improve poll ratings

- The tweet has attracted more than 500,000 retweets, along with over 1M likes.



https://www.forbes.com/sites/nicholasreimann/2022/06/04/biden-posts-his-top-tweet-as-president-thanks-to-bts/?sh=41f7841274fb

Laws for Recommendation

全 国 人 民 代 表 大 会
The National People's Congress of the People's Republic of China

首页 | 宪法 | 人大机构 | 栗战书委员长 | 代表大会会议 | 常委会会议 | 委员长会议 | 权威发布 | 立法 | 监督 | 代表
对外交往 | 选举任免 | 法律研究 | 理论 | 机关工作 | 地方人大 | 图片 | 视频 | 直播 | 专题 | 资料库 | 国旗 | 国歌 | 国徽

当前位置： 首页

中华人民共和国数据安全法

（2021年6月10日第十三届全国人民代表大会常务委员会第二十九次会议通过）

来源： 中国人大网 浏览字号： 大 中 小 2021年06月10日 19:58:46

目 录

全 国 人 民 代 表 大 会
The National People's Congress of the People's Republic of China

首页 | 宪法 | 人大机构 | 栗战书委员长 | 代表大会会议 | 常委会会议 | 委员长会议 | 权威发布 | 立法 | 监督 | 代表
对外交往 | 选举任免 | 法律研究 | 理论 | 机关工作 | 地方人大 | 图片 | 视频 | 直播 | 专题 | 资料库 | 国旗 | 国歌 | 国徽

当前位置： 首页

中华人民共和国个人信息保护法

（2021年8月20日第十三届全国人民代表大会常务委员会第三十次会议通过）

来源： 中国人大网 浏览字号： 大 中 小 2021年08月20日 16:53:44

目 录

国家市场监督管理总局
State Administration for Market Regulation

请输入要查询的内容

首页 机构 新闻 政务 服务 互动 专题

你的位置:首页 > 新闻 > 媒体聚焦

互联网信息服务算法推荐管理规定

发布时间：2022-01-04 12:58 信息来源：中国网信网

国家互联网信息办公室
中华人民共和国工业和信息化部
中华人民共和国公安部
国家市场监督管理总局
令

第9号

《互联网信息服务算法推荐管理规定》已经2021年11月16日国家互联网信息办公室2021年第20次室务会议审议通过，并经工业和信息化部、公安部、国家市场监督管理总局同意，现予公布，自2022年3月1日起施行。

国家互联网信息办公室主任 庄荣文
工业和信息化部部长 肖亚庆
公安部部长 赵克志
国家市场监督管理总局局长 张 工
2021年12月31日

互联网信息服务算法推荐管理规定

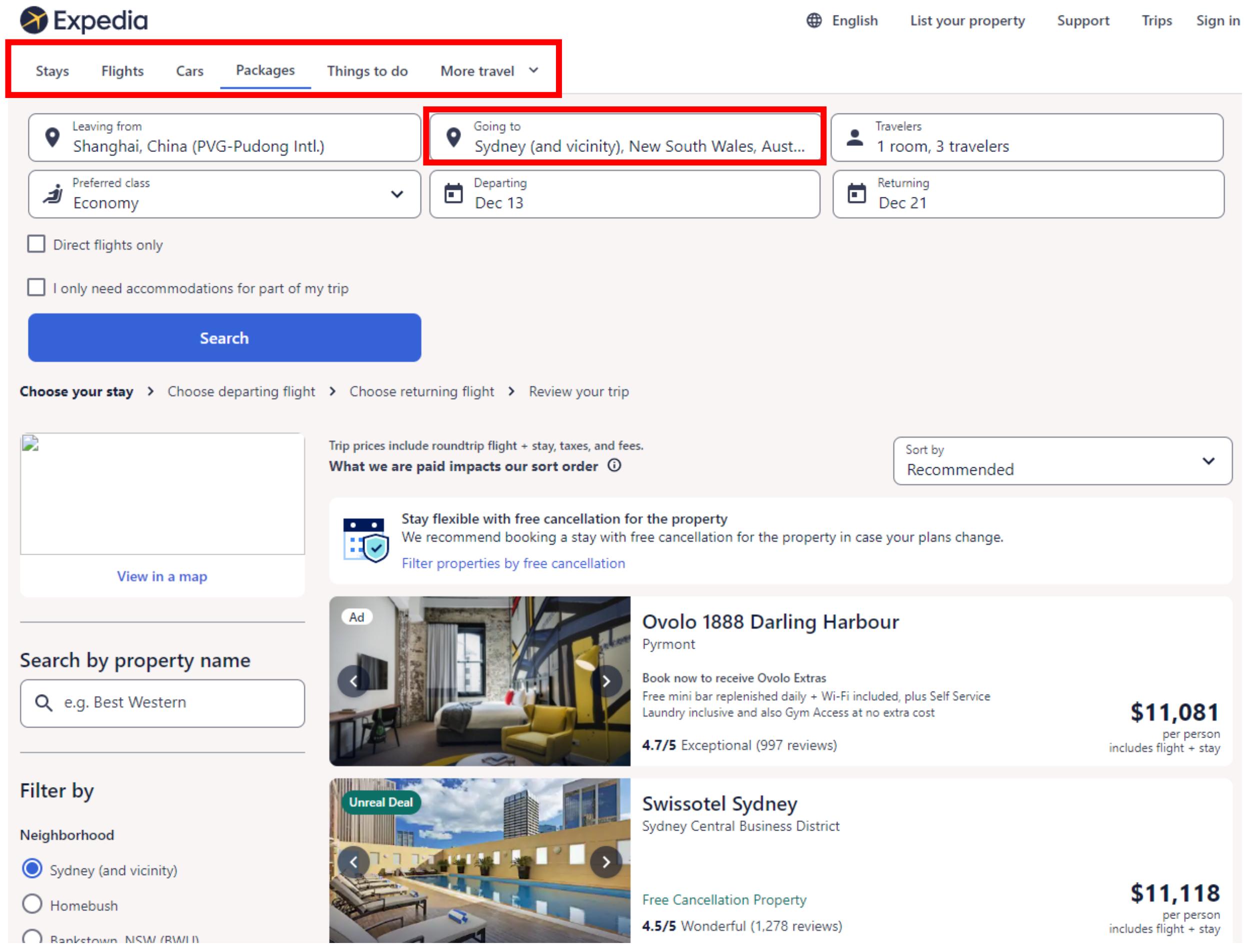
https://www.samr.gov.cn/xw/mtjj/202201/t20220104_338828.html

Content

- Recommendation age
- A case study of recommender systems
- The 3C principle to build recommender systems
- Classic recommender systems
- Evaluation metrics for recommender systems

Item domain recommendation

- Item types
  - E.g., flights, stays, cars
- Item relations:
  - E.g., destination (flight), location (hotel)



User domain recommendation

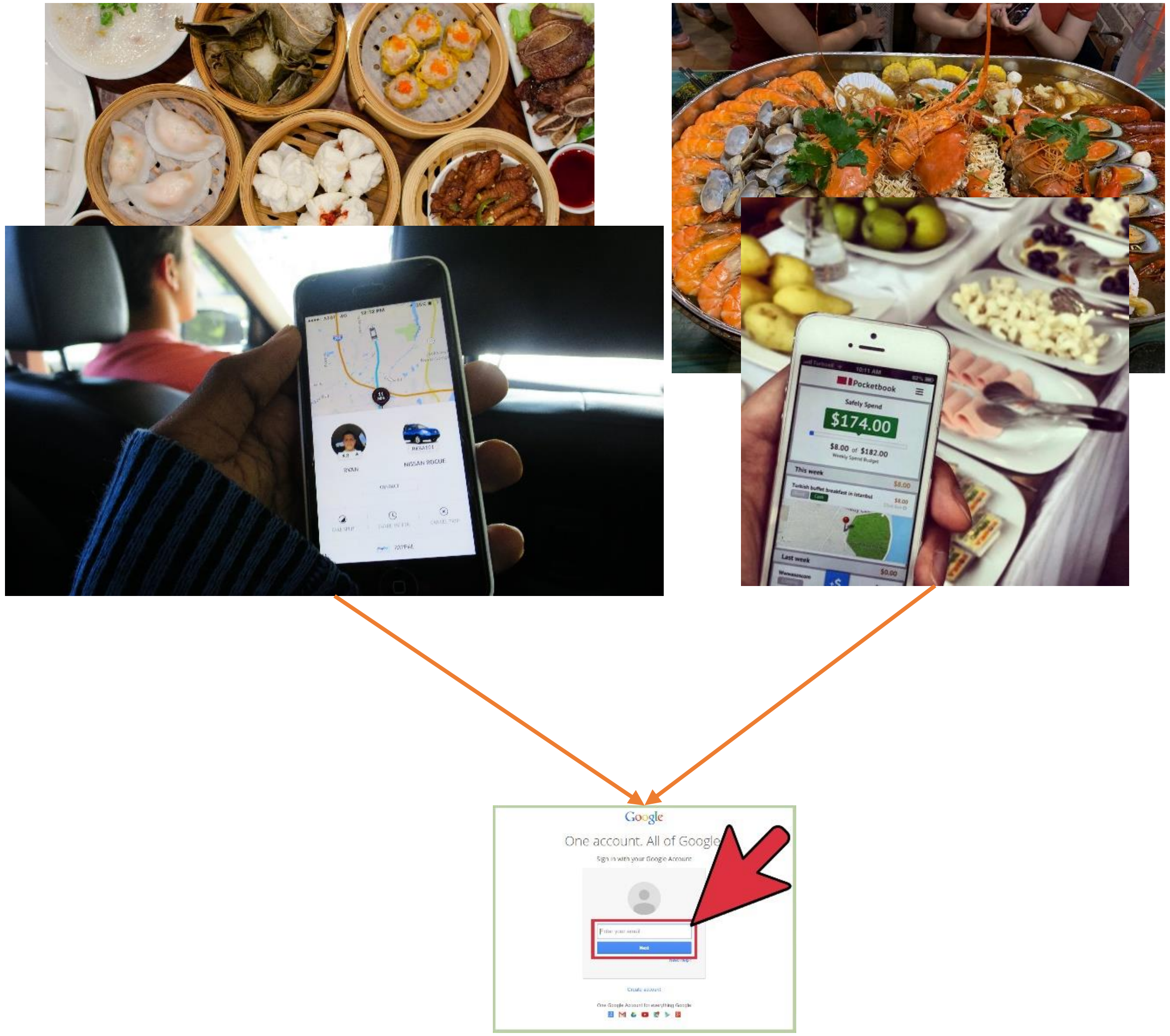
- User groups
  - E.g., family vs. colleagues for stays
- Social relations:
  - E.g., local food recommendations according to your local friends



Spatial domain recommendation

- Geographic points
  - E.g., Chinese food restaurants (Hongkong), Seafood restaurants (Sydney)

- Systems/Sites:
  - E.g., Uber (google account), Eats Apps (google account)



Temporal domain recommendation

- Sessions:
  - E.g. a shopping transaction may help travel package recommendation
- Time period:
  - E.g., preferences of breakfast (morning) help to recommend dinner (evening)

Recent transaction

Travel package

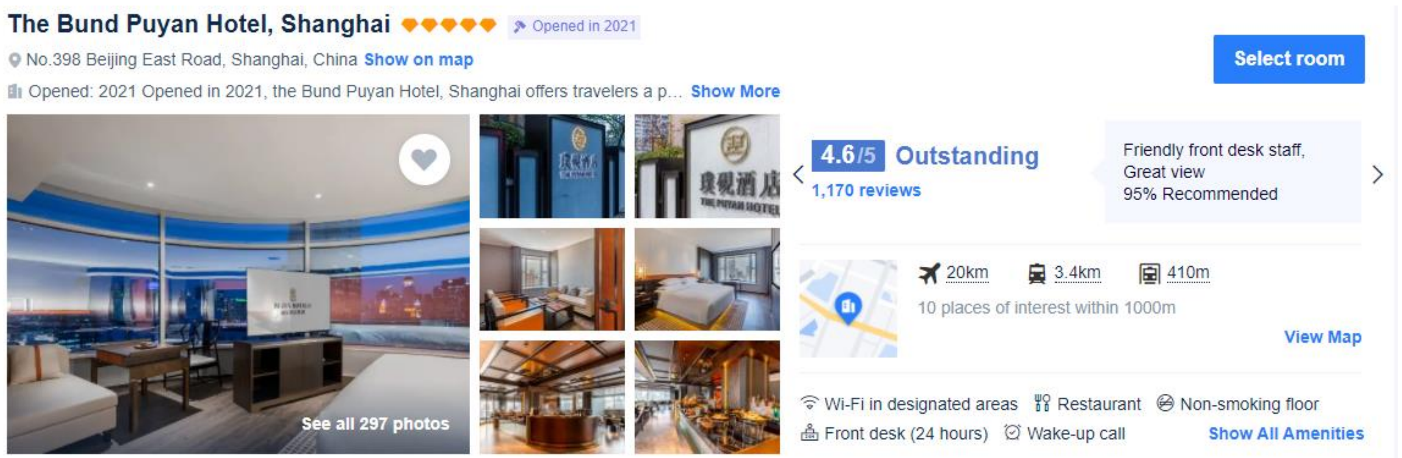
Breakfast

Dinne r

Data domain recommendation

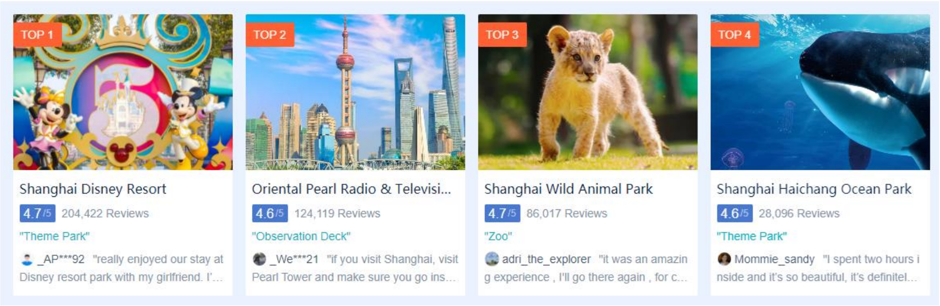
- Modalities:
  - E.g. rating, description, photos, geo information

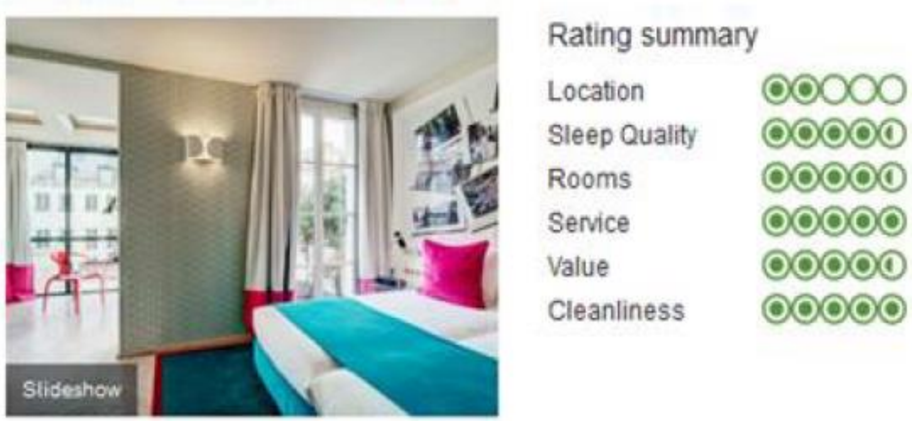
- Distributions:
  - Ratings have different distributions for different domains, e.g., food vs stays
  - The content of image distribution is also different, e.g., color distributions between food and stays.



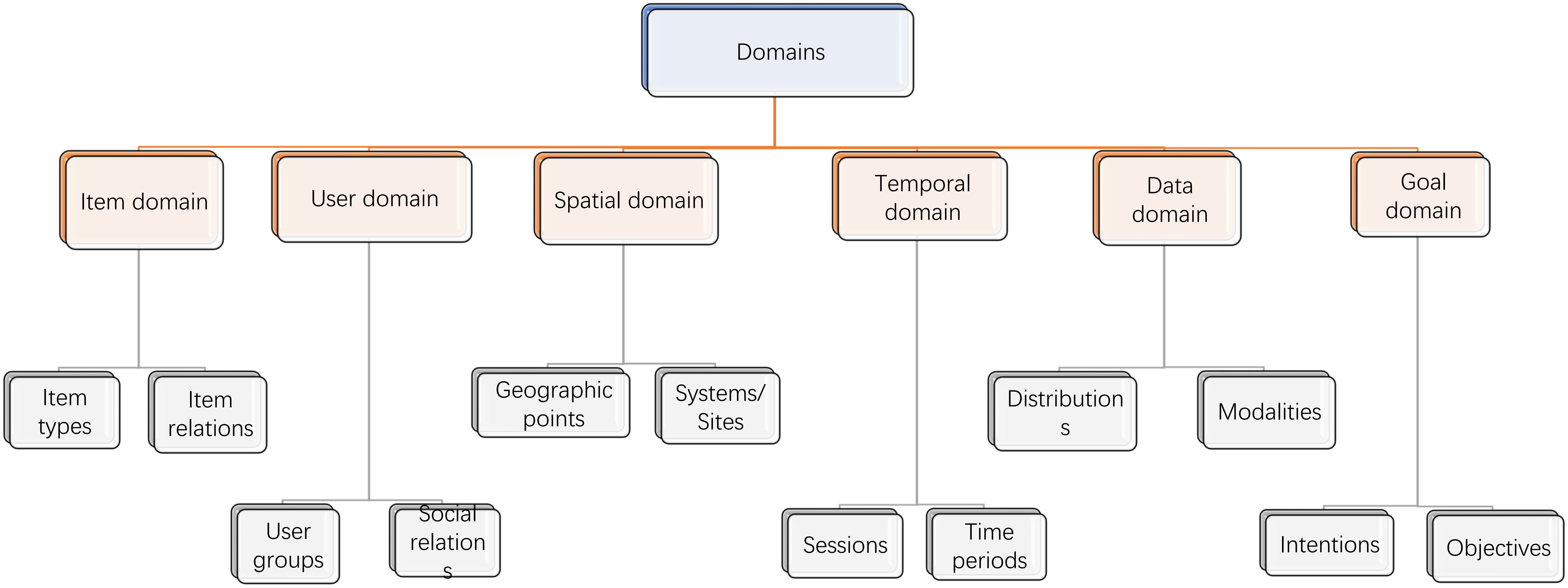
Goal domain recommendation

- Intentions:
  - A travel often consists of multiple intentions：
  - E.g. visiting attractions, enjoy local food.
- Objectives:
  - Multiple criteria:
    - E.g. location, cleanliness, services
  - Multiple tasks:
    - E.g. booking the cheapest flight and a sea view hotel





Modeling RS from different domains



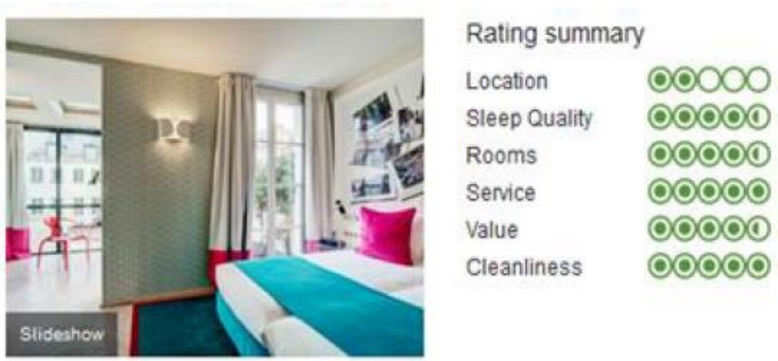
Content

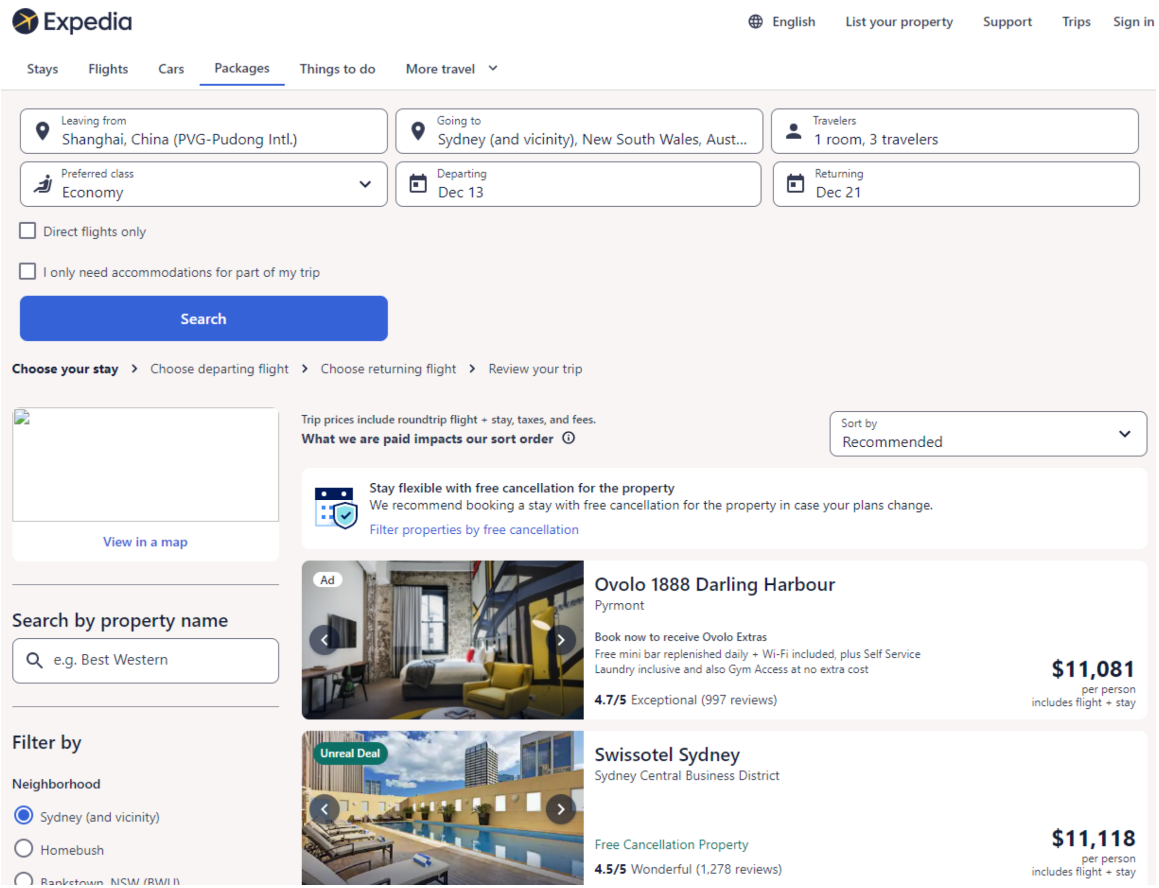
- Recommendation age
- A case study of recommender systems
- The 3C principle to build recommender systems
- Classic recommender systems
- Evaluation metrics for recommender systems

3C: Complement, Composite, Context

- **Complement**
  - To provide a complete view for recommendation by leveraging complement information over multiple domains.
- **Composite**
  - To provide comprehensive views for recommendation by integrating composite information over multiple domains.
- **Context**
  - To provide a conscious view for recommendation by considering context information over multiple domains.

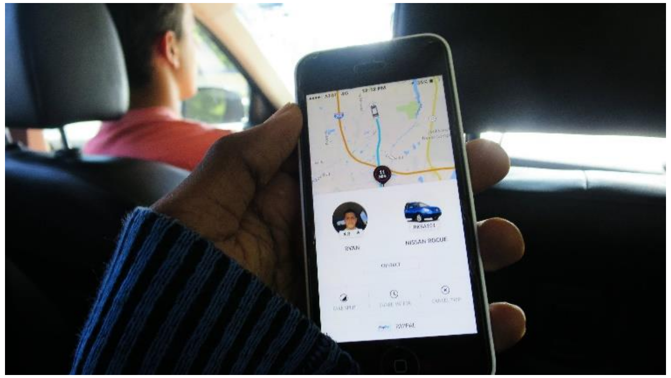
Case study: recommendations for a travel



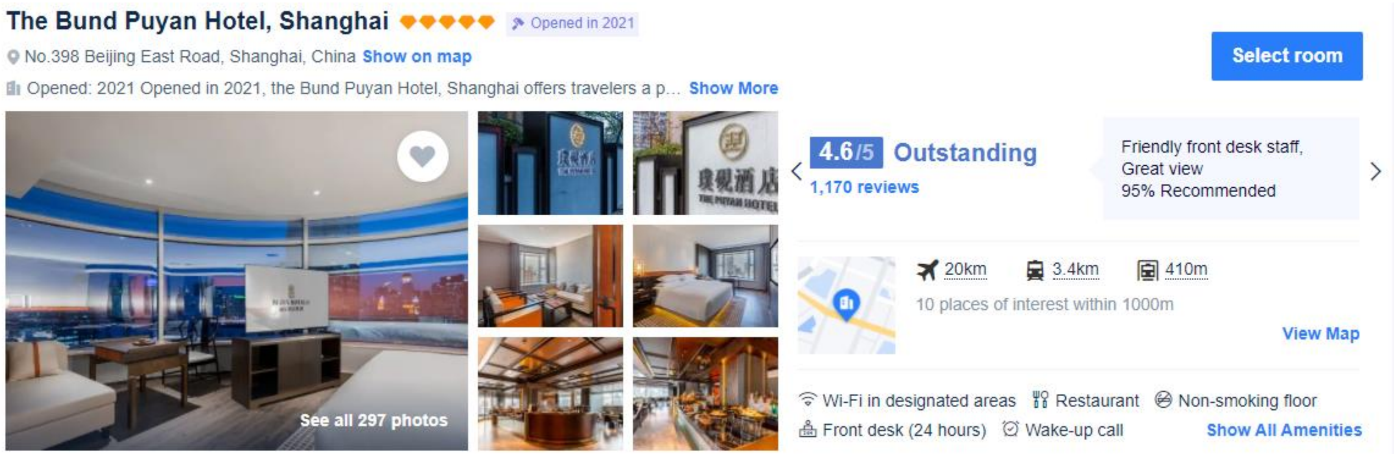


Examples: complement information

- Transfer Uber histories for local restaurant recommendation

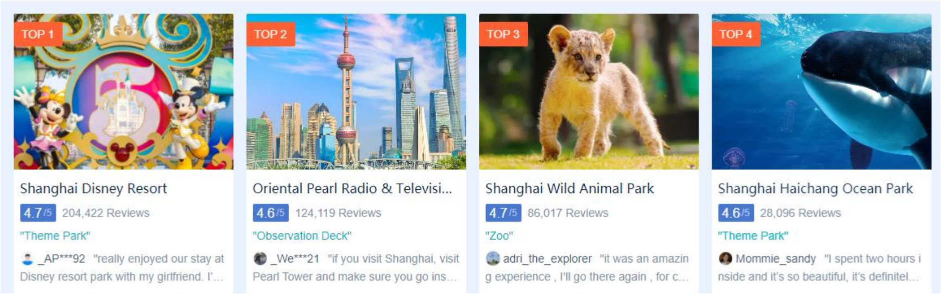


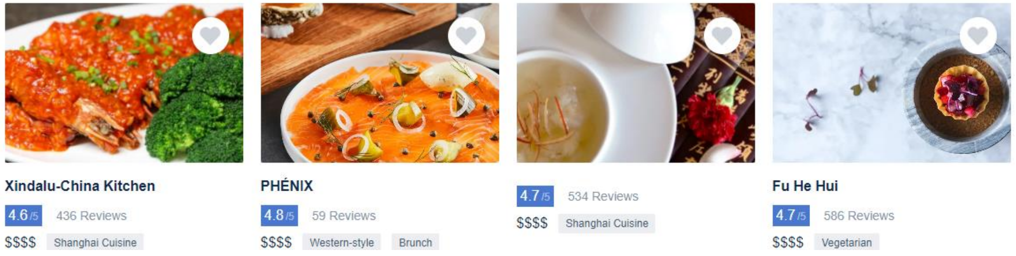
- Multimodal modeling for describing a hotel



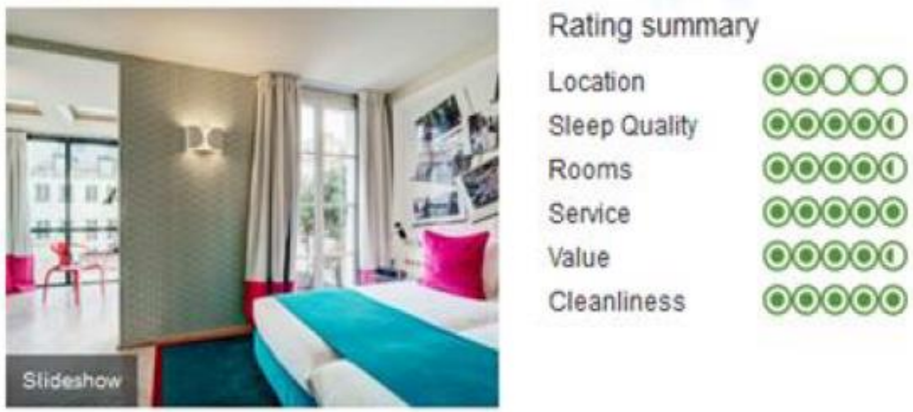
Examples: composite information

- A travel consists of multiple requirements, including attraction, food, and living





- A recommendation for a hotel needs to comprehensively consider multiple aspects



Examples: context information

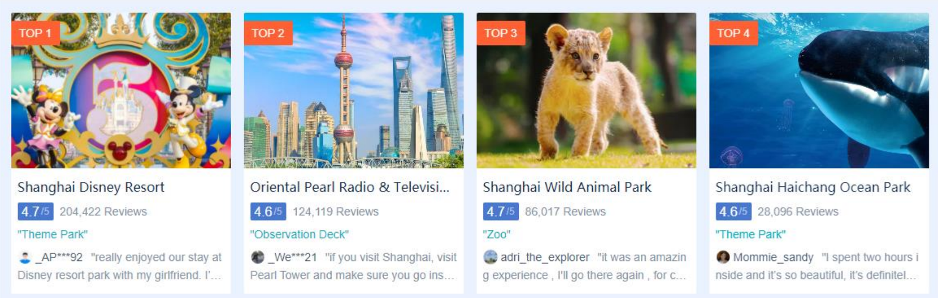
- A recommendation needs to consider the time and location

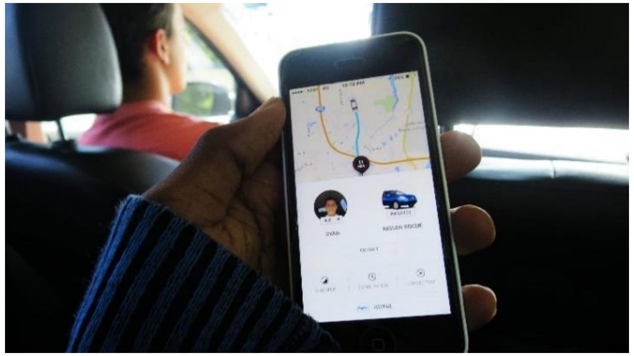
Breakfast

Dinne r

- A recommendation needs to consider the preferences of all members in a user group

The way to the metaverse of RSs

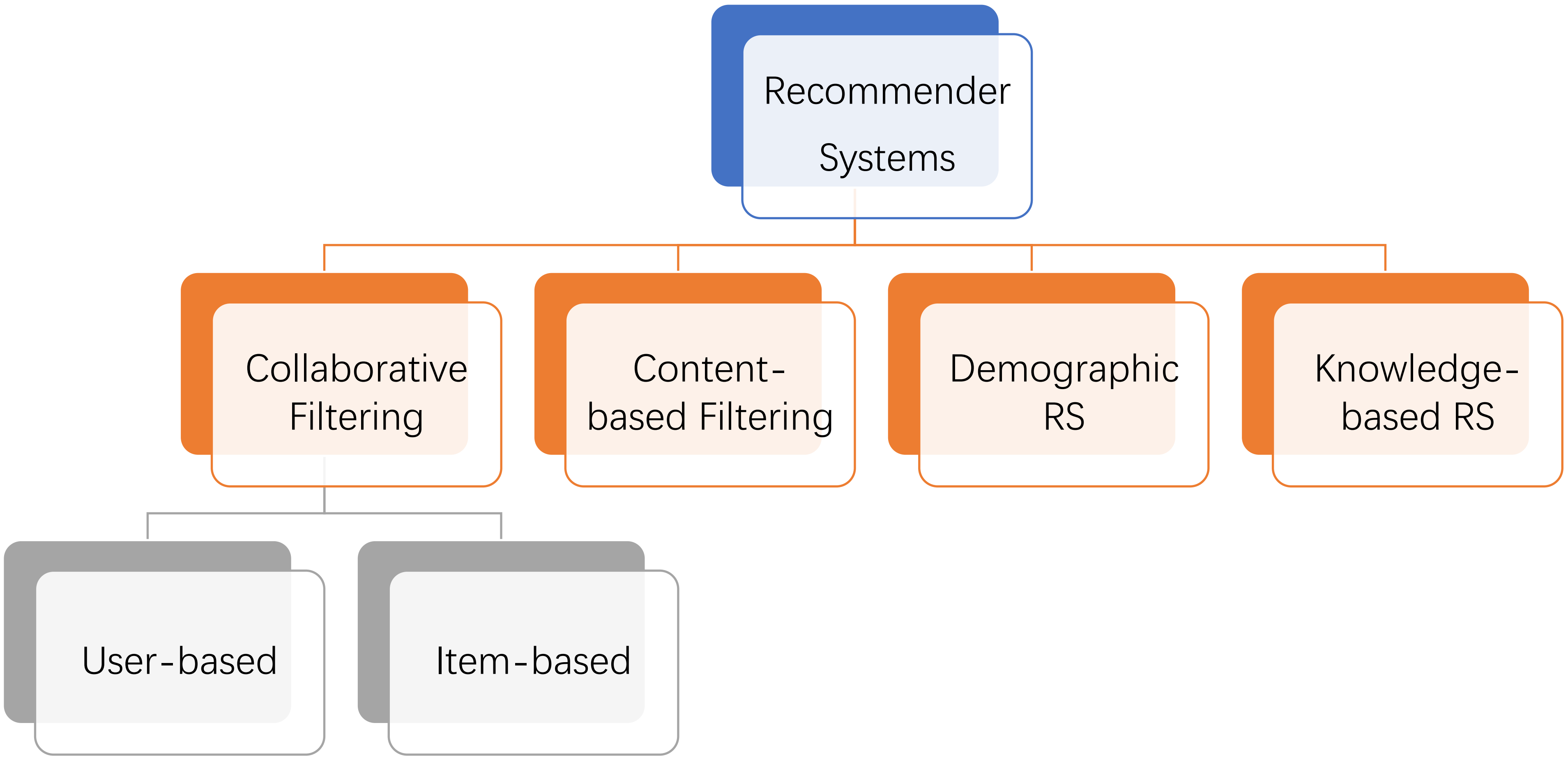




Content

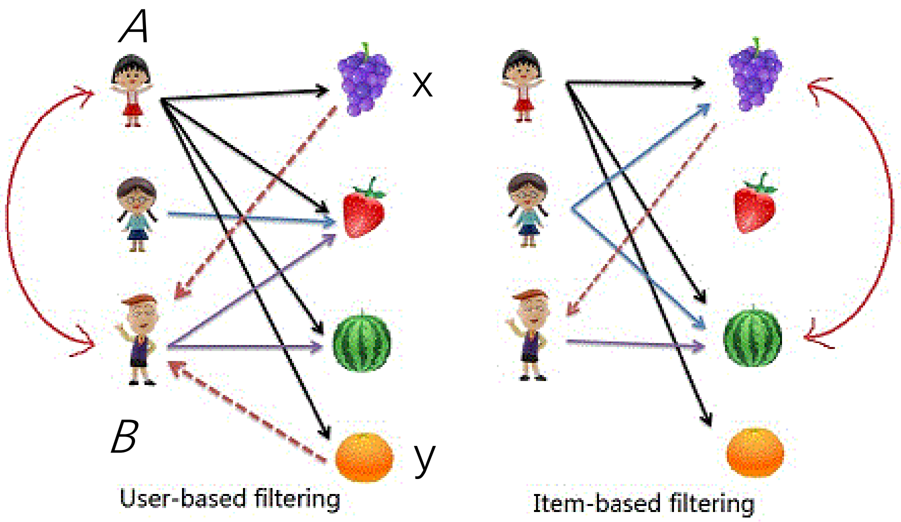
- Recommendation age
- A case study of recommender systems
- The 3C principle to build recommender systems
- Classic recommender systems
- Evaluation metrics for recommender systems

Classic recommender systems



Collaborative Filtering (CF)

- Intuition (user-based filtering): If user ***A*** related to user ***B*** and ***A*** bought x and y, then ***B*** bought x tend to buy y.
- Famous examples(item-based filtering): Amazon.com's recommender system
- Facebook, MySpace, LinkedIn use collaborative filtering to recommend new friends, groups, and other social connections.



Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms.* In Proceedings of the 10th international conference on World Wide Web, Hong Kong.

Memory-based CF

- The memory-based approach uses user rating data to compute the similarity between users or items.
- Typical examples of this approach are neighborhood-based CF and item-based/user-based top-N recommendations.
- User-based CF:

$$r_{\langle u,i\rangle} = \frac{1}{Z}\sum_{v\in\Gamma(u)} sim(u,v)\, r_{\langle v,i\rangle}$$

- Item-based CF:

$$r_{\langle u,i\rangle} = \frac{1}{Z}\sum_{j\in\Gamma(i)} sim(i,j)\, r_{\langle u,j\rangle}$$

CF for friend recommendation

- Similarity (Jaccard)
  - $s_{\langle u_1, u_2 \rangle} = \frac{|\{u_2, u_5, u_6\} \cap \{u_1, u_3, u_6\}|}{|\{u_2, u_5, u_6\} \cup \{u_1, u_3, u_6\}|} = \frac{1}{5}$
  - $s_{\langle u_1, u_3 \rangle} = \frac{1}{4}, \quad s_{\langle u_1, u_4 \rangle} = \frac{1}{4}$
  - $s_{\langle u_1, u_5 \rangle} = \frac{1}{4}, \quad s_{\langle u_1, u_6 \rangle} = \frac{2}{5}$
- Ranking
  - $r_{\langle u_1, u_3 \rangle} = s_{\langle u_1, u_2 \rangle} * 1_{\langle u_2, u_3 \rangle} + s_{\langle u_1, u_3 \rangle} * 0_{\langle u_3, u_3 \rangle} + s_{\langle u_1, u_4 \rangle} * 1_{\langle u_4, u_3 \rangle} + s_{\langle u_1, u_5 \rangle} * 0_{\langle u_5, u_3 \rangle} + s_{\langle u_1, u_6 \rangle} * 0_{\langle u_6, u_3 \rangle} = \frac{9}{20}$
  - $r_{\langle u_1, u_4 \rangle} = s_{\langle u_1, u_2 \rangle} * 0_{\langle u_2, u_4 \rangle} + s_{\langle u_1, u_3 \rangle} * 1_{\langle u_3, u_4 \rangle} + s_{\langle u_1, u_4 \rangle} * 0_{\langle u_4, u_4 \rangle} + s_{\langle u_1, u_5 \rangle} * 0_{\langle u_5, u_4 \rangle} + s_{\langle u_1, u_6 \rangle} * 1_{\langle u_6, u_4 \rangle} = \frac{13}{20}$

	u_1	u_2	u_3	u_4	u_5	u_6
u_1	0	1	0	0	1	1
u_2	1	0	1	0	0	1
u_3	0	1	0	1	0	0
u_4	0	0	1	0	0	1
u_5	1	0	0	0	0	1
u_6	1	1	0	1	1	0

CF for rating prediction

- Similarity (Pearson correlation)

$$s_{\langle u_1,u_2\rangle} = \frac{\sum_{i\in I(u_1,u_2)}(r_{\langle u_1,i\rangle} - \bar{r}_{u_1})(r_{\langle u_2,i\rangle} - \bar{r}_{u_2})}{\sqrt{\sum_{i\in I(u_1,u_2)}(r_{\langle u_1,i\rangle} - \bar{r}_{u_1})^2}\sqrt{\sum_{i\in I(u_1,u_2)}(r_{\langle u_2,i\rangle} - \bar{r}_{u_2})^2}}$$

- Rating Prediction

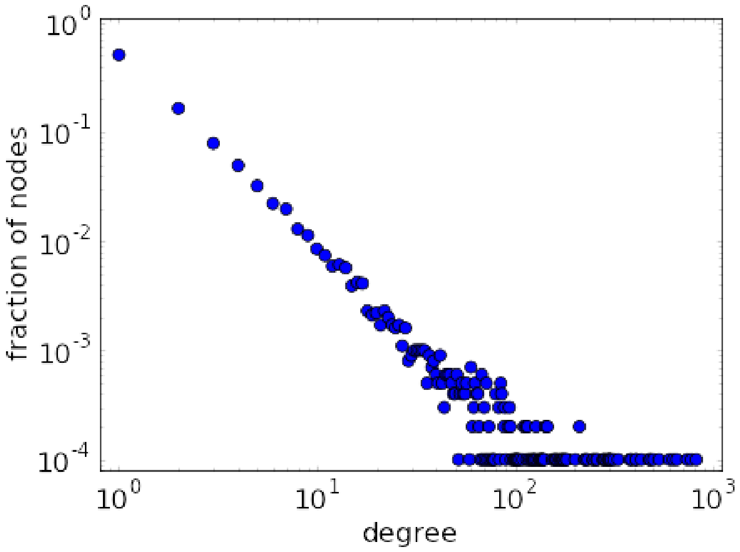
$$r_{\langle u,i\rangle} = \bar{r}_u + \frac{\sum_{v\in\Gamma(u)} s_{\langle u_1,v\rangle}(r_{\langle v,i\rangle} - \bar{r}_v)}{\sum_{v\in\Gamma(u)} s_{\langle u_1,v\rangle}}$$

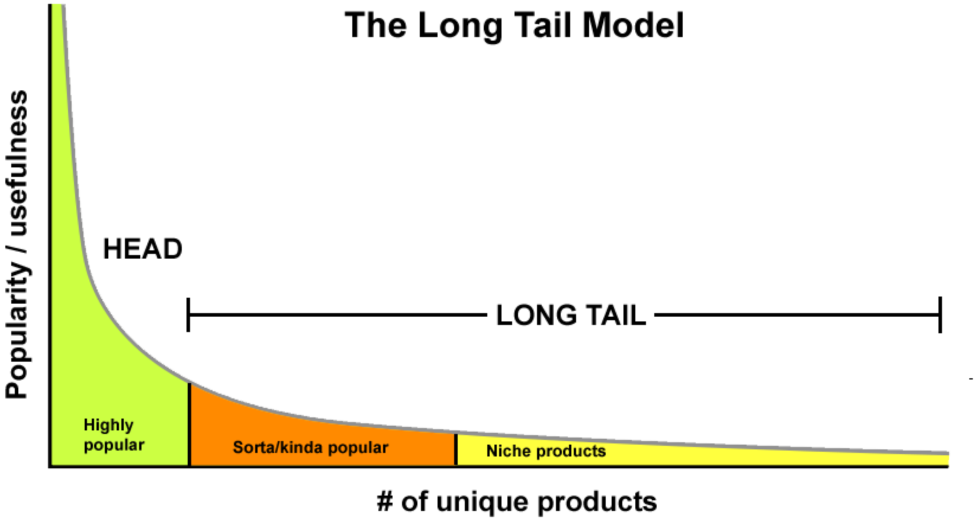
- Example:
  - $\bar{r}_{u_1} = 4,\ \bar{r}_{u_2} = 3.5,\ \bar{r}_{u_3} = 4,\ \bar{r}_{u_4} = 5,\ \bar{r}_{u_5} = 1,\ \bar{r}_{u_6} = 3$
  - $s_{\langle u_1,u_2\rangle} = 0,\quad s_{\langle u_1,u_3\rangle} = 0.7071,\quad s_{\langle u_1,u_4\rangle} = 0$
  - $s_{\langle u_1,u_5\rangle} = 0,\quad s_{\langle u_1,u_6\rangle} = 1$
  - $r_{\langle u_1,i_1\rangle} = 4 + \frac{0.7071*(4-4)+1*(2-3)}{0.7071+1} = 4 - 0.5858 = 3.4142$

	i_1	i_2	i_3	i_4	i_5
u_1		5		4	3
u_2	3		4		
u_3	4	5	3	4	4
u_4			5		
u_5	1				
u_6	2	4		3	

Data characteristics in recommender systems

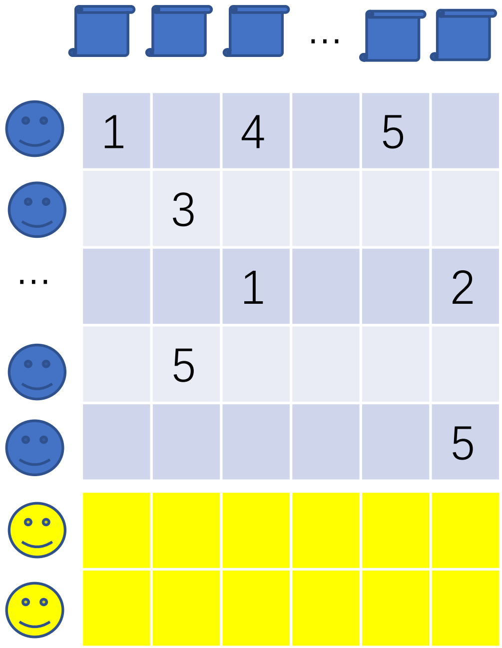
- Power law or Long tail distribution
  - Data associated with the majority of users are insufficient and even absent in real world.
  - In most recommender systems, the majority of users/items only associated with very few data while only minority of users/items have sufficient data





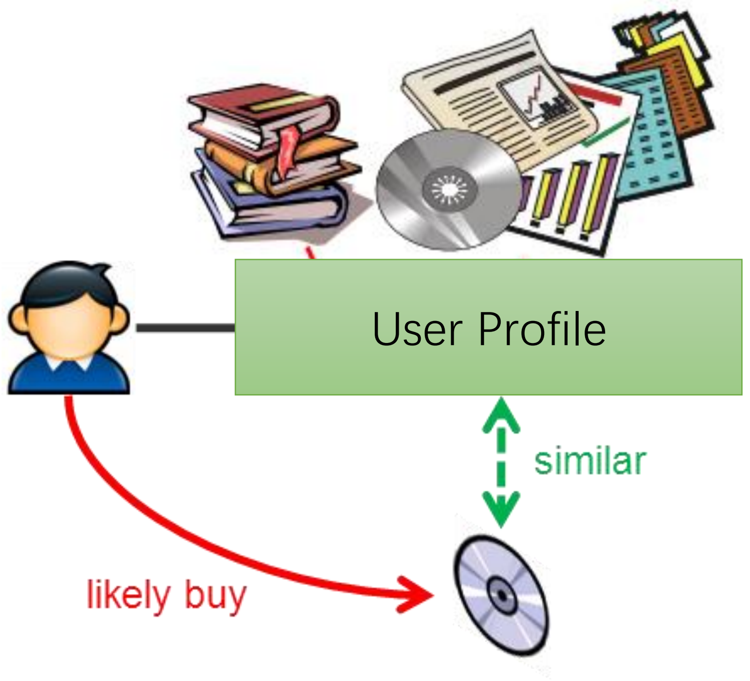
Challenges in collaborative filtering

- **Data Sparsity**
  - In real-world recommender systems, the user-item matrix is very sparse.
- **Cold Start**
  - When new users or new items are added, the system cannot recommend to these users and these items.
- **Scalability**
  - There are millions of users and products in real systems.
    - Large amount of computation
    - Large storage



Content-based Filtering (CBF)

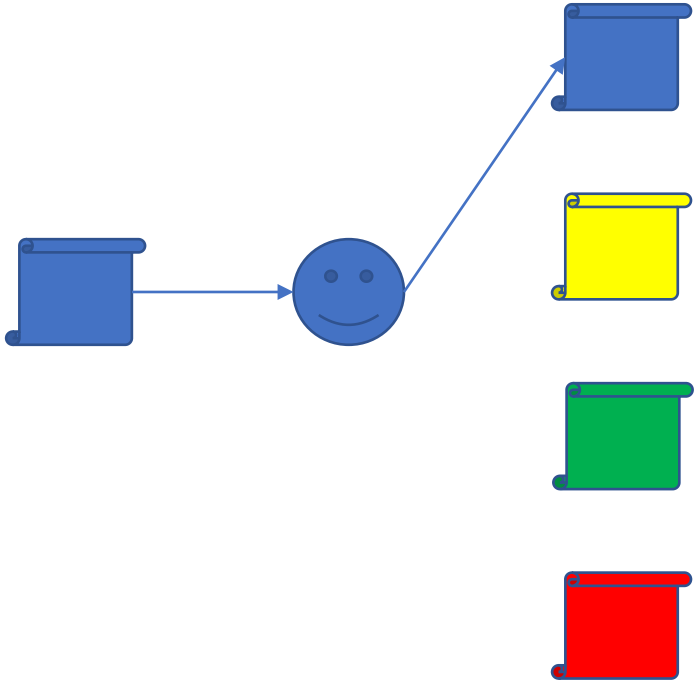
- CBF is based on the features of items
  - Attributes of items
  - Description of items
  - Text of an article
- User profile is built with the features of historical items
- Recommend items according to user profile



Challenges in content-based filtering

- **Limited Content Analysis**
  - System has a limited amount of information on its users or the content of its items.

- **Over-specialization**
  - The system can only recommend items that highly similar with user's profile, the user is limited to be recommended items similar to those already rated.



Demographic RS

- The demographic RS provides suggestions based on the characteristics of the specific audience segments.
- This type of recommendation engine takes available user data (age, gender, location, etc.), classifies it into specific audience segments, and then puts in a bigger picture to fill the gaps in the data.

- **Pros:** Non-existence of cold start
- Cons: Non-personalized recommendations

Name	Gender	Occupation	Country	Age
John	M	Student	France	13
Paul	M	Doctor	France	34
Sarah	F	Student	USA	12
Mike	M	Teacher	France	27

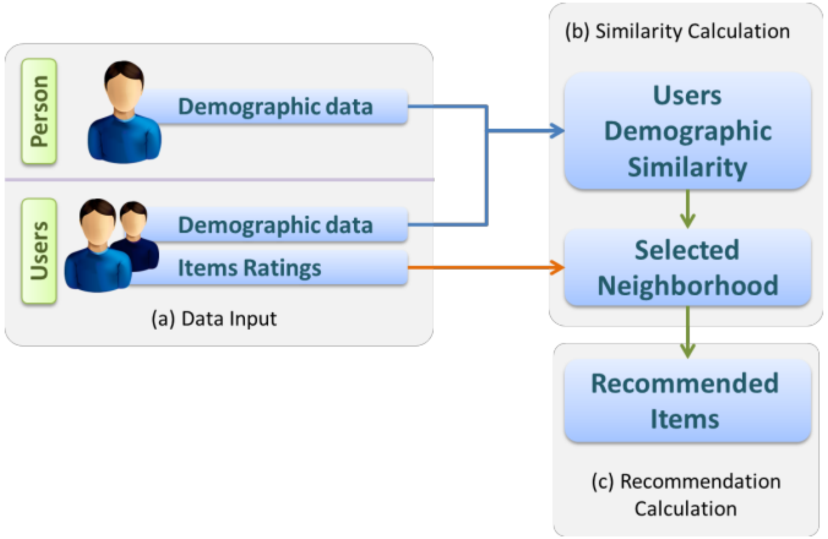
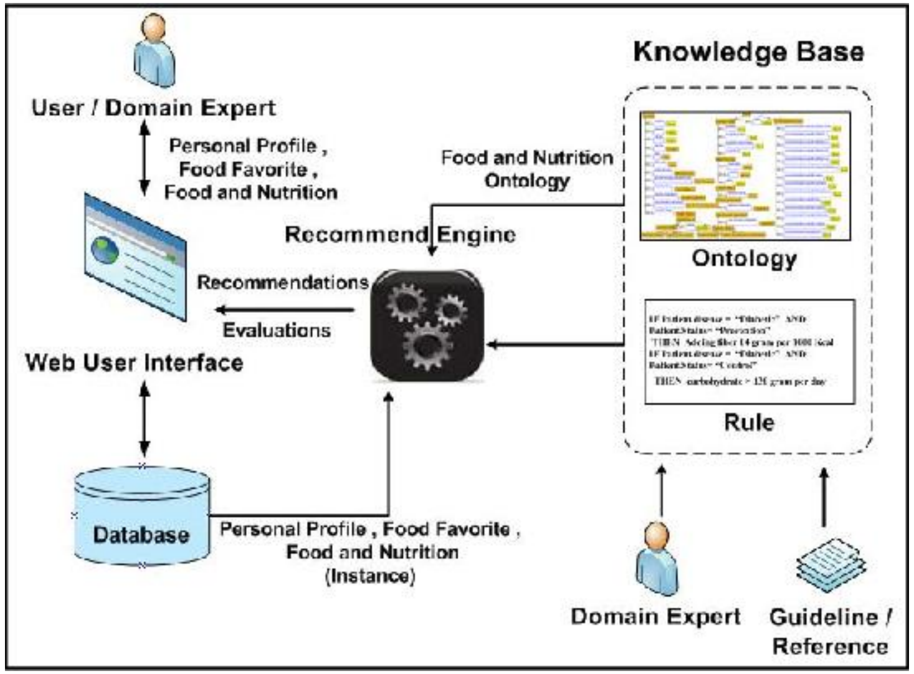


Fig. 2. Demographic-based approach for new users.

Knowledge-based RS

- Knowledge-based RS are based on explicit knowledge about the item assortment, user preferences, and recommendation criteria (i.e., which item should be recommended in which context).

- **Pros:** Non-existence of cold start
- Cons: Need to define recommendation knowledge in an explicit fashion

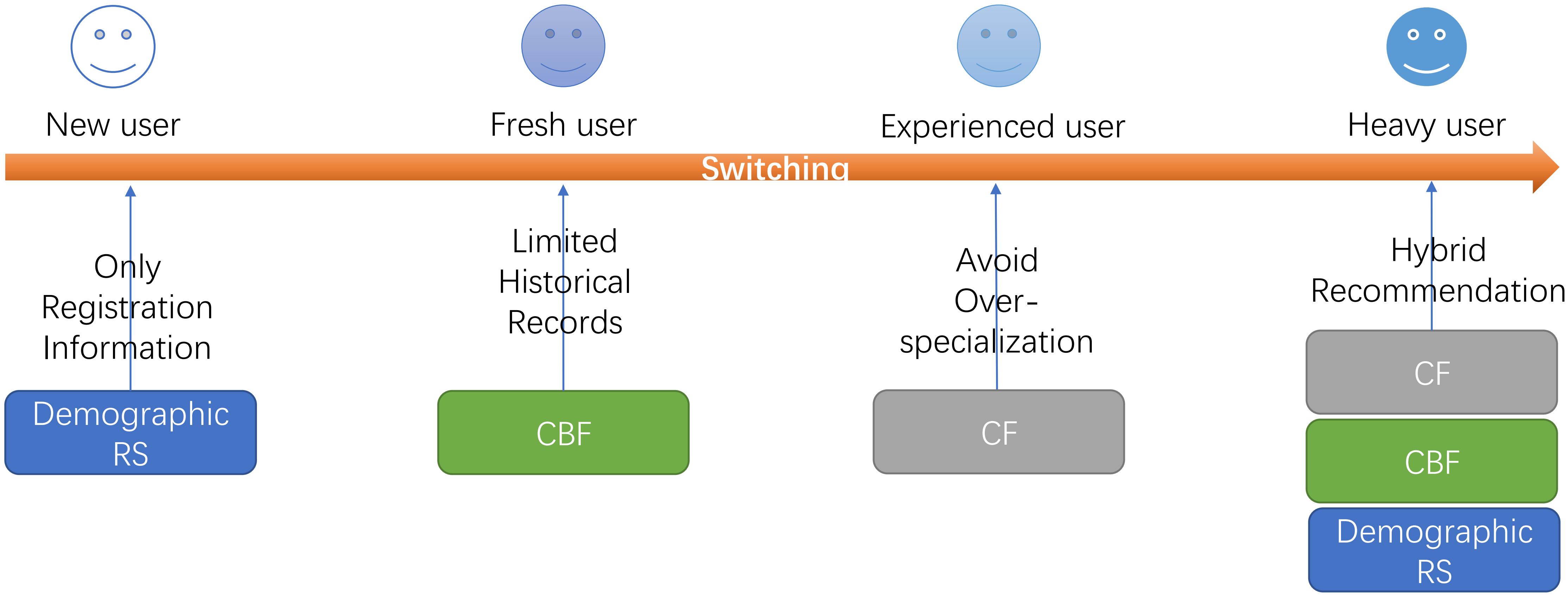


Hybrid RS

- Most RSs use a hybrid approach, combining CF, CBF, and other approaches.
- Some hybridization techniques include:
  - **Weighted**: Combining the score of different recommendation components numerically.
  - **Switching**: Choosing among recommendation components and applying the selected one.
  - **Mixed**: Recommendations from different recommenders are presented together to give the recommendation.
  - **Feature Combination**: Features derived from different knowledge sources are combined together and given to a single recommendation algorithm.
  - **Feature Augmentation**: Computing a feature or set of features, which is then part of the input to the next technique.
  - **Cascade**: Recommenders are given strict priority, with the lower priority ones breaking ties in the scoring of the higher ones.
  - **Meta-level**: One recommendation technique is applied and produces some sort of model, which is then the input used by the next technique.

https://en.wikipedia.org/wiki/Recommender_system#Hybrid_recommender_systems

Example: Hybrid RS

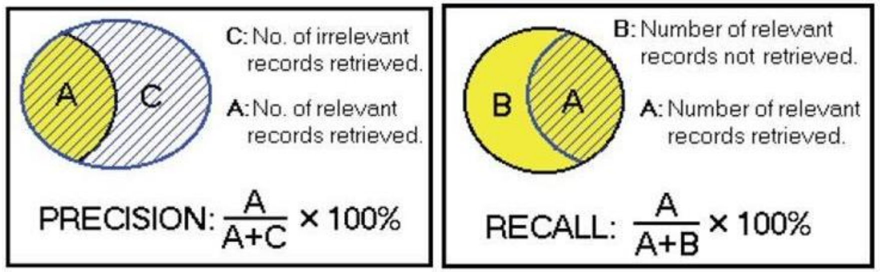


Content

- Recommendation age
- A case study of recommender systems
- The 3C principle to build recommender systems
- Classic recommender systems
- Evaluation metrics for recommender systems

Evaluation metric types

- **Relevance-based evaluation**



- **Ranking-based evaluation**

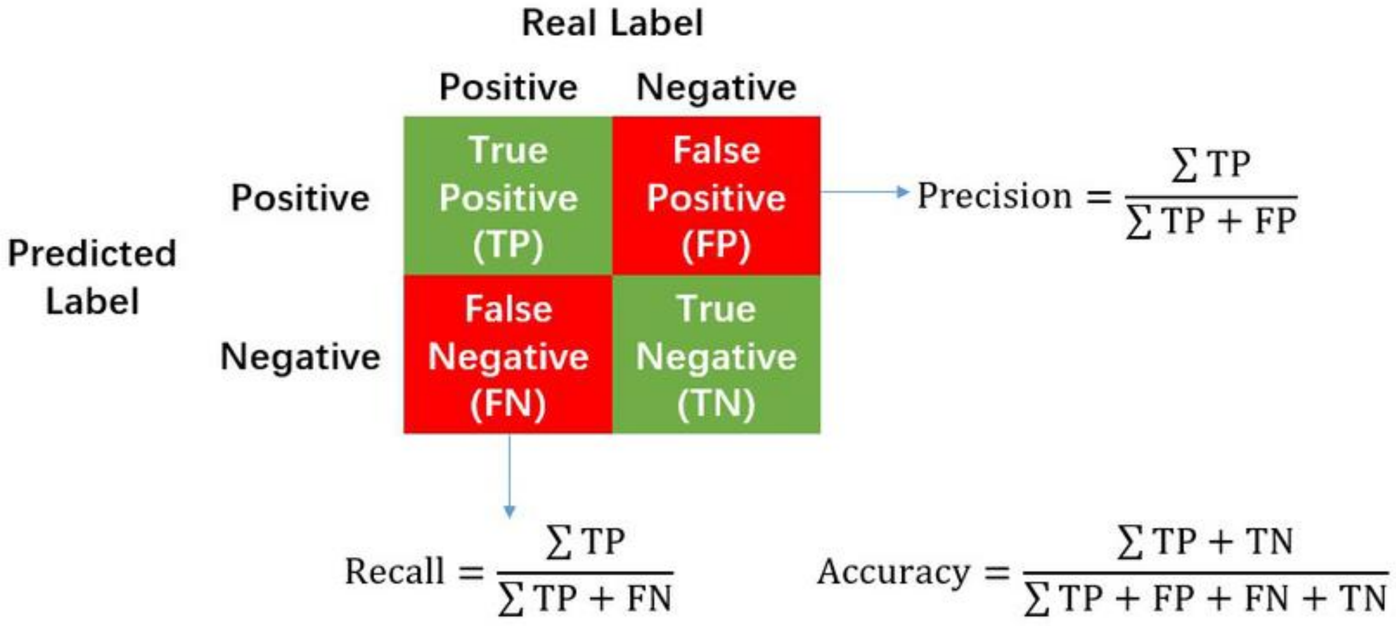
- **Rating-based evaluation**

	Item 1	Item 2	Item 3	...	Item n
User 1	2	3	?	...	5
User 2	?	4	3	...	?
User 3	3	2	?	...	3
...	...	...	...	...	...
User m	1	?	5	...	4

Relevance-based evaluation

- Recommendation can be viewed as a special information retrieval task to recommend the items which relevant to users' preference.

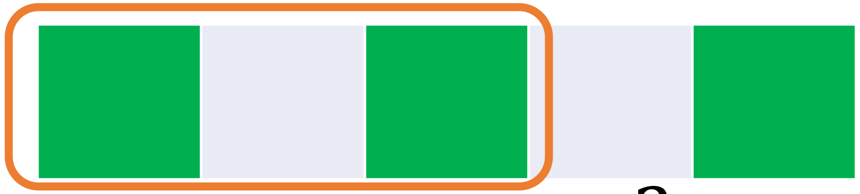
  - ***Recall***
  - ***Precision***
  - ***Hit Ratio***



Recall@K

- The fraction of relevant items over all N relevant items

$$Recall@K = \frac{\sum_{k=1}^{K} TP_k}{N}$$

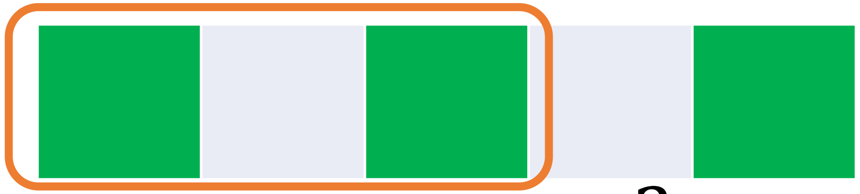


$$Recall@3 = \frac{2}{5}$$

Precision@K

- The fraction of relevant items over top K recommended items

$$Precison@K = \frac{\sum_{k=1}^{K} TP_k}{K}$$

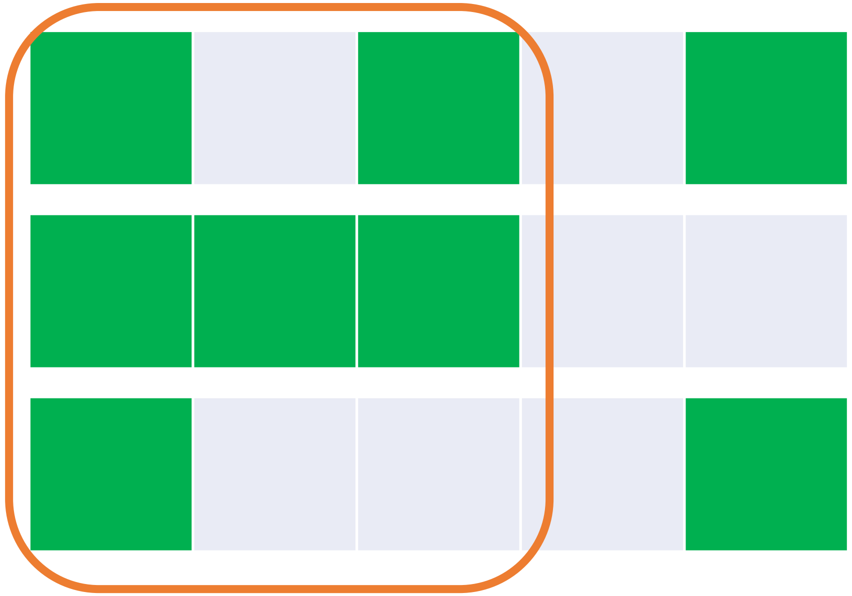


$$Recall@3 = \frac{2}{3}$$

Hit Ratio (HR@K)

- The ratio of hit (predicted) true results w.r.t. the total ground truth results.

$$HR@K = \frac{\sum_{q=1}^{Q} \sum_{k=1}^{K} TP_{q,k}}{\sum_{q=1}^{Q} \min(N_q, K)}$$



$$HR@3 = \frac{2+3+1}{3+3+2} = \frac{6}{8}$$

Ranking-based evaluation

- The most common way to assess the recommendation performance is to measure whether relevant items are placed in the top positions of a recommendation list.

  - ***MAP:*** *Mean Average Precision*
  - ***MRR:*** *Mean Reciprocal Rank*
  - ***nDCG:*** *Normalized Discounted Cumulative Gain*

AP@K

- Average Precision (AP) is a ranked precision metric that places emphasis on highly ranked correct predictions (hits).
- AP@K is the average result over Precision@1~K, which is defined as

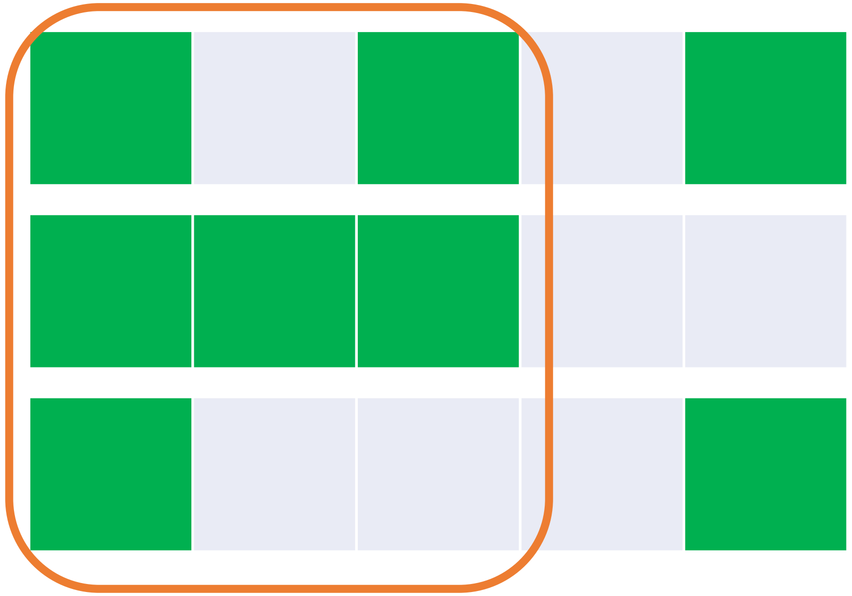
$$AP@K = \frac{\sum_{k=1}^{K} relevant(k) * Precision@k}{\min(K, N)}$$

$$AP@4 = \frac{1 + 0 + \frac{2}{3} + 0}{\min(2,4)} = \frac{5}{6}$$

MAP@K

- Mean Average Precision (MAP) is the mean of AP over all the recommended lists:

$$MAR@K = \frac{\sum_{q=1}^{Q} AP_q@K}{Q}$$



$$AP_1@3 = \frac{\left(1 + \frac{2}{3}\right)}{3} = \frac{5}{9}$$

$$AP_2@3 = \frac{(1 + 1 + 1)}{3} = 1$$

$$AP_3@3 = \frac{1}{2}$$

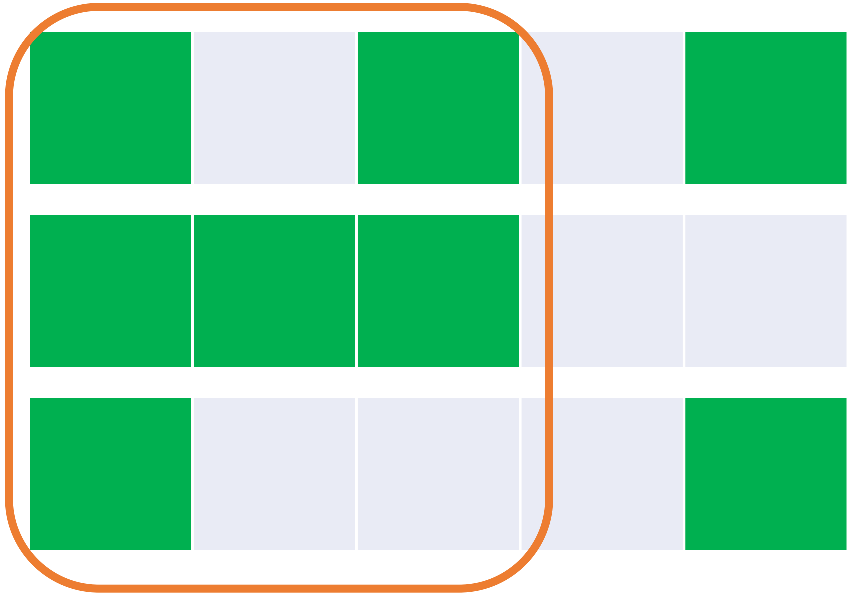
$$MAR@3 = \frac{1}{3}\left(\frac{5}{9} + 1 + \frac{1}{2}\right)$$

MRR@K

- Mean Reciprocal Rank evaluates any process that produces a list of possible responses to Q testing users, ordered by probability of correctness., which is defined as:

$$RR@K = \sum_{k=1}^{K} \frac{relevant(k)}{k}$$

$$MRR@K = \frac{\sum_{q=1}^{Q} RR_q@K}{Q}$$



$$RR_1@3 = \frac{1}{1} + \frac{1}{3} = \frac{4}{3}$$

$$RR_2@3 = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} = \frac{4}{3}$$

$$RR_3@3 = \frac{1}{1}$$

$$MRR@3 = \frac{1}{3}\left(\frac{4}{3} + \frac{4}{3} + 1\right)$$

nDCG@K

- Normalized Discounted Cumulative Gain is a measure of ranking quality which places strong emphasis on relevant items

$$nDCG@K = \frac{DCG@K}{IDCG@K}$$

- where

$$DCG@K = \sum_{i=1}^{K} \frac{2^{rel(i)} - 1}{\log_2(i+1)}, \qquad IDCG@K = \sum_{i=1}^{K} \frac{1}{\log_2(i+1)}$$

$$DCG@4 = 1 + 0 + \frac{1}{\log_2(3+1)} + 0 = \frac{3}{2}$$

$$IDCG@K = \log_2 2 + \log_2 3 + \log_2 4 + \log_2 5$$

Rating-based evaluations

- Rating-based evaluation aims to evaluate the prediction error between the true rating values and the predicted ones.
- Two commonly used rating-based evaluation metrics:
  - **MAE**: Mean Absolute Error
  - **RMSE**: Root Mean Squared Error

	Item 1	Item 2	Item 3	...	Item n
User 1	2	3	?	...	5
User 2	?	4	3	...	?
User 3	3	2	?	...	3
...	...	...	...	...	...
User m	1	?	5	...	4

MAE & RMSE

- Metrics measuring error rate
  - **Mean Absolute Error (MAE)** computes the deviation between predicted ratings and actual ratings

$$MAE = \frac{\sum_{i=1}^{N} |r_i - p_i|}{N}$$

  - **Root Mean Square Error (RMSE**) is similar to MAE, but places more emphasis on larger deviation

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (r_i - p_i)^2}{N}}$$

Rating prediction: an example

# ID	User ID	Movie ID	Rating (r_i)	Prediction (p_i)	$\|r_i - p_i\|$	$(r_i - p)^2$
1	1	134	5	4.5	0.5	0.25
2	1	238	4	5	1	1
3	1	312	5	5	0	0
4	2	134	3	5	2	4
5	2	767	5	4.5	0.5	0.25
6	3	68	4	4.1	0.1	0.01
7	3	212	4	3.9	0.1	0.01
8	3	238	3	3	0	0
9	4	68	4	4.2	0.2	0.04
10	4	112	5	4.8	0.2	0.04
Sum					4.6	5.6

MAE = 4.6/10=0.46 RMSE = $\sqrt{5.6/10}$

Summary

- We have entered recommendation age
- There are more and more emerging needs for recommender systems
- Different recommendation approaches have their pros and cons
- The recommendation results should be evaluated with diverse metrics

Thinking

- Try to find recommender systems in your daily life.
- Are the recommender systems only used in e-commerce?
- According to you, what are the difficulties to build recommend systems?

Assignment

MovieLens 100K Dataset

MovieLens 100K movie ratings. Stable benchmark dataset. 100,000 ratings from 1000 users on 1700 movies.

Task: Apply memory-based CF to predict movie ratings.

- Evaluation metrics: MAE, RMSE

Experimental dataset: https://grouplens.org/datasets/movielens/100k/

- Training set: ua.base
- Testing set: ua.test