

Lec01 Introduction to Recommender Systems

This is what I recorded after listening to Professor Lee Jong-wook's Introduction to Recommendation System (**ESW4024_41**) class in the first semester of 2023 at Sungkyunkwan University.

Contents

1. What are recommender systems?
 2. How to design recommender models?
 3. References
-

1. What are Recommender systems?

추천 시스템이란, 유저의 선호도 및 과거 행동을 바탕으로 개인에 맞는 관심사를 제공하는 분야를 말한다.

그렇다면 왜, 추천시스템이 요즘 유행할까?

Information Overload

- The explosion of data result in approximately 40 trillion gigabytes in 2020.
 - 1.7 MB of data are created every second for every person
- Google gets over 3.5 billion searches daily.
 - 1.2 trillion searches yearly and more than 40,000 queries per second.

추천시스템은 어떤 형태로 나눌수 있을까? 아니 내가 찾고자 하는 아이템과 관련성 있는 아이템들에 어떻게 접근할 수 있을까?

How to Access to Relevant Items?

- How can we help users get access to **relevant items**?

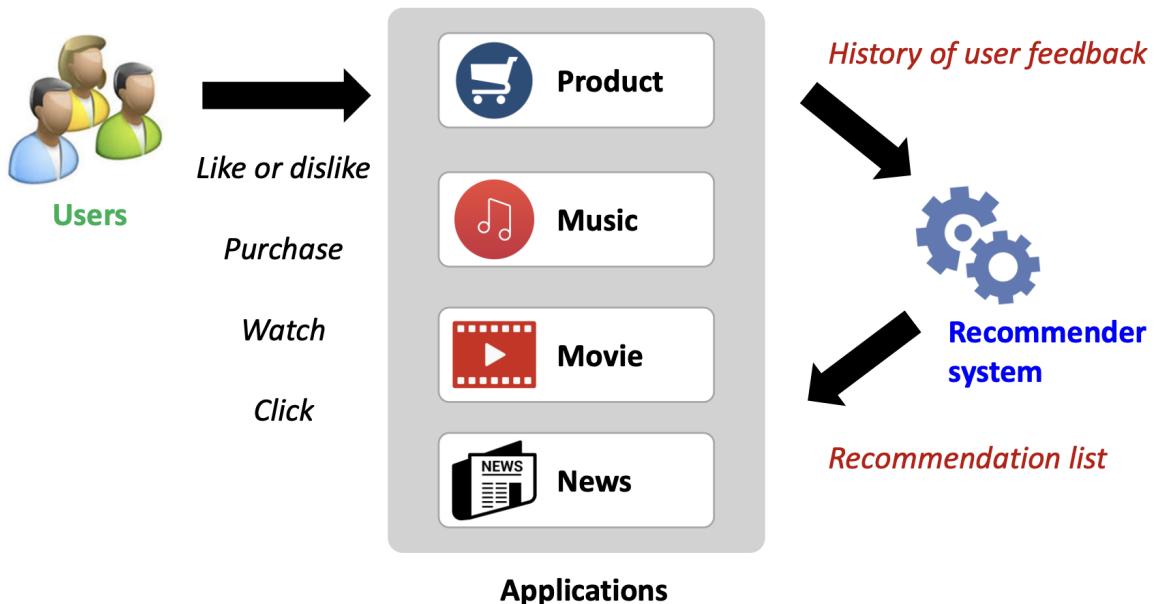
- **Pull mode (search engines)**
 - **Users** take the initiative.
 - Ad-hoc information need
- **Push mode (recommender systems)**
 - **Systems** take the initiative.
 - Systems have the user's potential information need.

Types of Recommendations

- Editorial and hand-curated
 - Lists of favorites
 - Lists of essential items
- **Simple aggregates**
 - Most popular, recent uploads
- Tailored to individual users

What are Recommender Systems?

Information filtering systems to predict the user's hidden preference for items



추천시스템 어떤 종류가 있지?

Product Recommendation



Your recently viewed items and featured recommendations

This screenshot shows a grid of product recommendations. An orange arrow points from the text 'Inspired by your browsing history' to one of the recommended items. The items shown are Moko cases for Kindle Voyage and Kindle 8th Generation. Below the grid is a row of small thumbnail images representing other products.

Amazon product recommendation

Customers who viewed this item also viewed these products

This screenshot shows four product recommendations based on previous viewing history. Each item includes a thumbnail, the product name, price, and a 'View options' button.

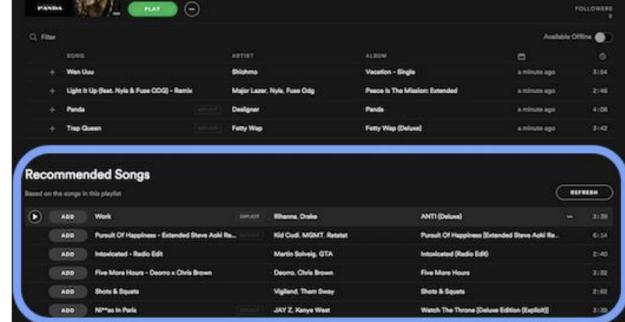
Product	Price	Action
Dualit Food XL1500 Processor	\$560	Add to cart
Kenwood kMix Manual Espresso Machine	\$250	Select options
Weber One Touch Gold Premium Charcoal Grill-57cm	\$225	Add to cart
NoMU Salt Pepper and Spice Grinders	\$3	View options

Movie & Music Recommendation



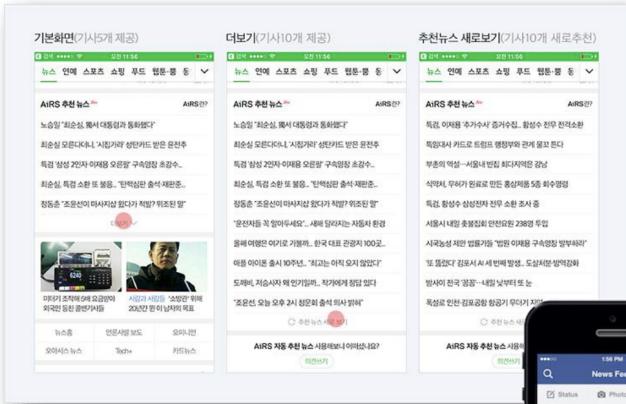
Netflix movie recommendation

Spotify music recommendation



10

News Recommendation



Naver News recommendation

Facebook News recommendation



11

Other Various Recommendations



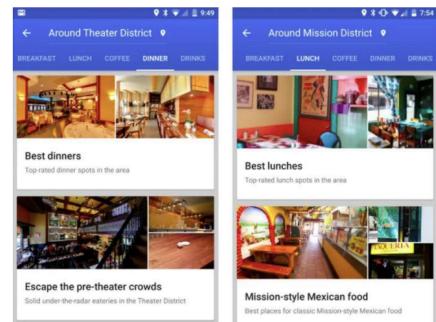
Social recommendation

People You May Know

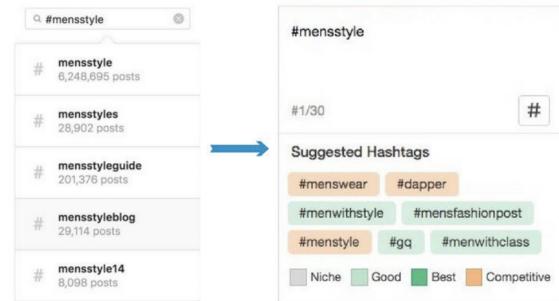
Add people you know as friends and connect with public profiles you like.



Restaurant recommendation



App recommendation



Tag recommendation

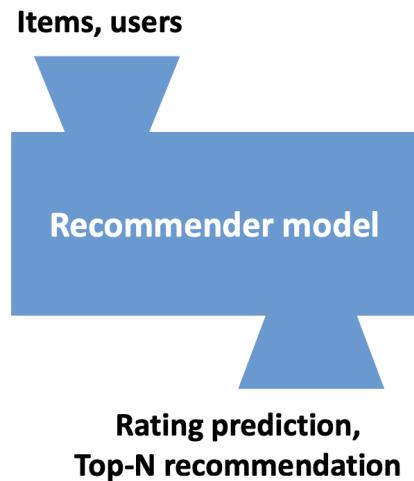
12

Recommendation Problems

- Estimate a **utility function** that **automatically predicts** how much a user would prefer an item.
 - 여기서 utility function은 user, item이 입력이고 출력으로 user's preference인 함 수이다.
- Based on
 - Past user behavior
 - Relationship to other uses
 - Item similarity
 - Context
 - ...

What are Recommender Models?

- Given
 - User models
 - Explicit/implicit feedback
 - hidden user preferences
 - Situational context
 - Item models
 - Descriptions of items
 - Characteristics of items



- Find
 - Rating prediction : predict the ratings of unrated items.
 - Top-N recommendation : rank top- N items among unrated items.

여기서 Explicit/implicit feedback이 중요하다. 이것들은 뭘까?

Explicit Feedback

단어 의미 그대로 명백하고 숨김없이 피드백을 주는 것이다.

- Directly ask the users what they like.



- Example : star ratings
 - Commonly, five stars with (or without) half-stars
 - Vote up/down

명백하지만 수집하기 힘들며(사람들이 많이 제공하지 않음) noise가 많이 있음 → 점수(평점)을 의미없이 입력 할 경우

Implicit Feedback

은연중에 확신없이 피드백을 주는 것이다. 예를 들어, 우리가 어떤 영화 A를 여러번 보는 행위를 생각할 수 있다. 어떤 영화 A를 본다는 것 자체에서 은연 중에 우리는 이 영화 A를 본다라는 피드백을 준다. 또한 여러번 본다는 것은 적어도 우리는 이 영화를 더 볼수도 있다는 피드백을 은연중에 내포하고 있다.

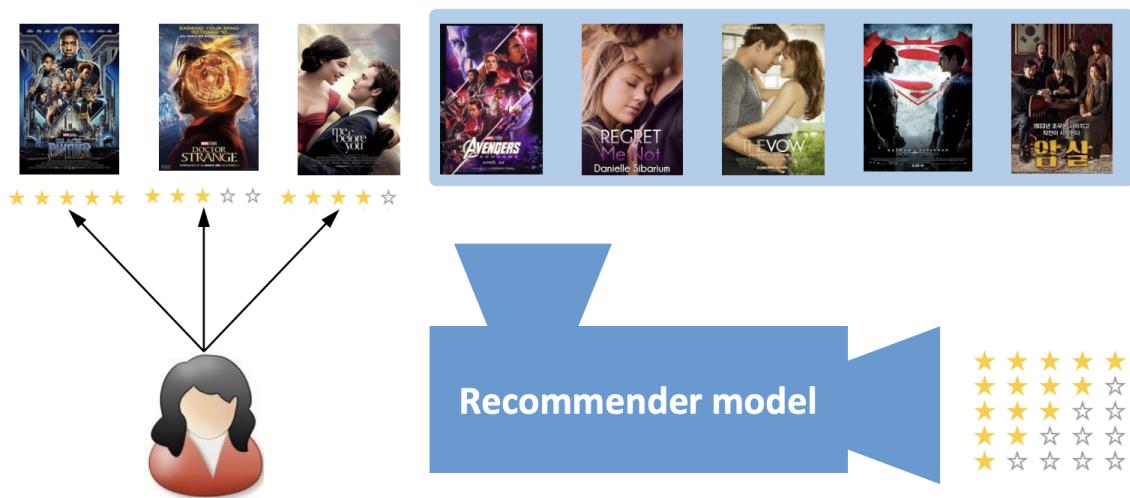
- Data collected from user behavior
 - Buy, click, view and watch
 - **Key difference** : user actions are for some other purpose **not expressing a direct preference.**
- The actions say a lot!
 - Clicked : Positive or noisy(그냥 막 누름)
 - Non-clicked : Negative or positive-unlabeled(이미 많이 본 것 or 관심있지만 클릭만 안했을 경우 등등)

추천시스템의 목적인 Rating Prediction, Top-N recommendation은 뭐지?

Rating Prediction

Predict users' ratings for unrated items(**regression problems**).

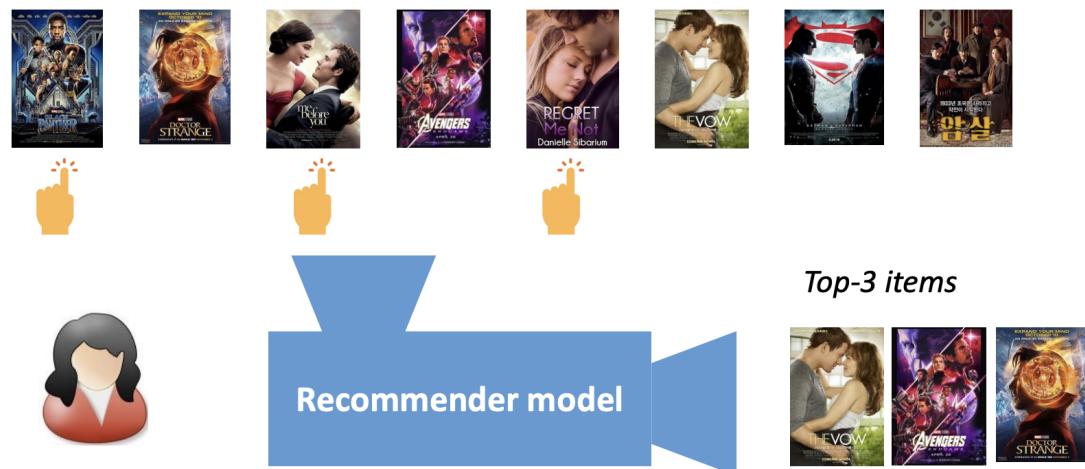
- Usually, predictions are computed from the **user's explicit feedback**.



Top -N Recommendation

For each user, recommended a list of top-N unrated items.

- usually, it is computed from the user's implicit feedback.



2. How to Design Recommender Models?

Approaches to Recommendations

- **Content-based recommendation** : recommend based on item features and descriptions.
- **Collaborative filtering** : recommend items based only on the **user's past behavior**.
 - Memory-based Approach는 아래 두 가지로 나뉘어짐
 - **User-based** : finding similar users and recommending what they like
 - **Item-based** : finding similar items to those that I have previously liked

1. Memory-Based Approach

- 유사한 사용자(Users)나 아이템(Item)을 사용
 - 특징 : 최적화 방법이나, 매개변수를 학습하지 않음. 단순한 산술 연산만 사용
 - 방법 : Cosine Similarity나 Pearson Correlation을 사용함, (* KNN 방법도 포함됨)
 - 장점 : 1. 쉽게 만들 수 있음
2. 결과의 설명력이 좋음
3. 도메인에 의존적이지 않음
 - 단점 : 1. 데이터가 축적 X or Sparse한 경우 성능이 낮음
2. 확장가능성이 낮음 (∵ 데이터가 너무 많아지면, 속도가 저하됨)

1. Model-Based Approach

- 기계학습을 통해 추천
 - 특징 : 최적화 방법이나, 매개변수를 학습
 - 방법 : 행렬분해(Matrix Factorization), SVD, 신경망
 - 장점 : 1. Sparse한 데이터도 처리 가능
 - 단점 : 1. 결과의 설명력이 낮음

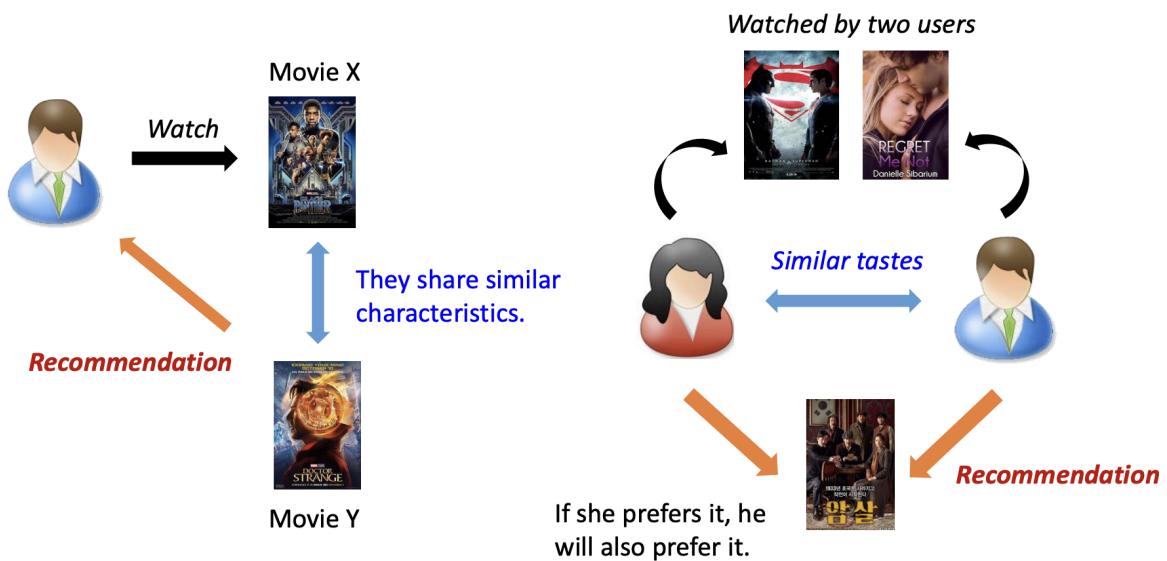
- **Social recommendation** : recommend based on trust graphs
- **Knowledge-based recommendation** : recommend based on the knowledge graph

- **Hybrid** : combine any of the above

이 수업에서는 Collaborative filtering 위주의 내용을 다룬다.

Categories of Recommender Models

Content-based vs. Collaborative filtering



Content-based Recommendation



➤ Show me more of the similar what I have liked.

- ◆ Alice likes three movies A, B, C .
- ◆ Because D has the same genre for B and C , Alice is likely to prefer D .



Limitation: Content Recommendation



➤ Limited content analysis

- ◆ Content may **not** be automatically extractable (e.g., multimedia).
- ◆ Missing domain knowledge
- ◆ Keywords may not be sufficient

➤ Overspecialization: more of the same, too similar items

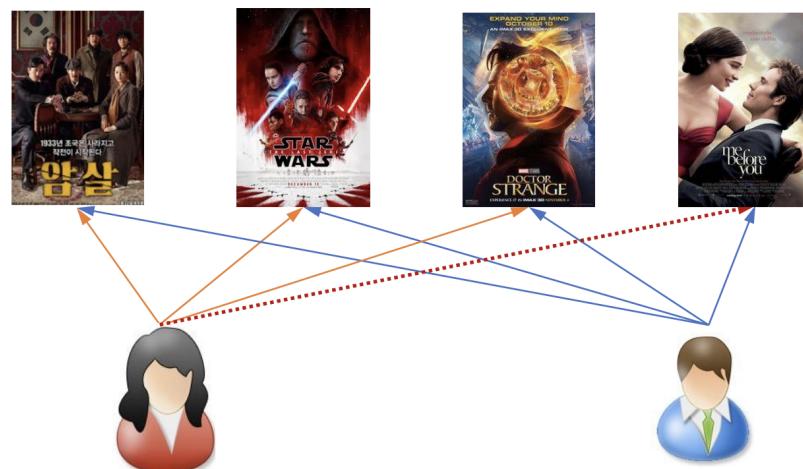


Collaborative Filtering



➤ Tell me what my peers like.

- ◆ Alice likes three movies *A*, *B*, and *C*.
- ◆ Bob also likes three movies *A*, *B*, and *C*. Besides, Bob likes *D*.
- ◆ Then, Alice is likely to prefer *D*.



24

Limitation: Collaborative Filtering



➤ Works for any kind of item: no feature selection

➤ Cold-start problem

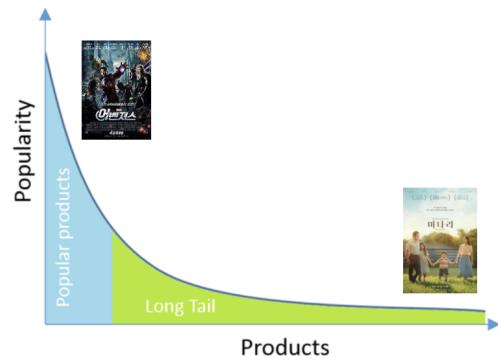
- ◆ Need enough users in the system to find a match

➤ Popularity bias

- ◆ Cannot recommend items to someone with a unique taste
- ◆ Tends to recommend popular items

➤ Impact of noise

- ◆ One account used by different people



이외에도 Social Recommendation, Knowledge-based Recommendation, Conversational Recommendation 등이 있다.