

Recommender System



sofo
고대맛집 단골리뷰

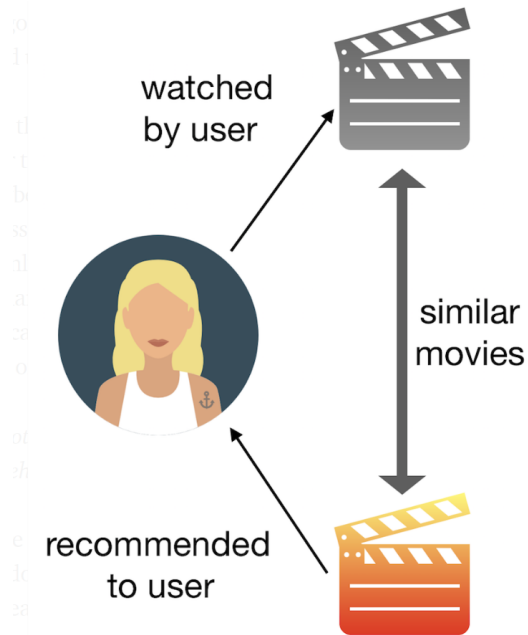
contents

1. Content-based methods
2. Collaborative filtering methods
 - 2.1 Memory based
 - 2.2 Model based

1. Content-based methods

- 사용자가 특정 아이템을 선호하는 경우 그 아이템과 비슷한 콘텐츠를 가진 다른 아이템을 추천해주는 방식
- ex) 사용자가 액션 영화에 높은 평점을 주었다면 장르가 액션인 다른 영화 추천
- Collaborative filtering methods와 다르게 사용자나 아이템의 추가정보를 사용한다.
(사용자의 추가정보 : 나이, 성별, 직업 / 아이템의 추가정보 : 장르, 감독, 주연배우...)
- ex) 10대는 주로 스릴러 영화를 좋아한다. -> 사용자의 나이 정보만 보고 추천 가능

1. Content-based methods



장점

- **“cold start problem”** 의 영향을 덜 받는다.
* cold start problem - 데이터가 부족해 정확한 추천이 이루어질 수 없는 문제
-> 새로운 사용자나 아이템이 들어왔을 때 추가정보 가지고 추천이 가능하기 때문.

단점

- 사용자와 아이템 간의 상호작용을 보지 않고 나이, 성별과 같은 특성 정보만을 활용해 추천한다. (less personalized)
-> **High bias Low variance**
- 유사한 아이템내에서만 추천해주기 때문에 사용자가 다양한 아이템에 접근할 기회가 적다. (information confinement)

2. Collaborative filtering methods

- 사용자가 아이템에 매긴 평점 등 사용자와 아이템 간에 기록된 상호작용을 통해 새로운 추천을 하는 방식. 이 때, 상호작용은 “**user-item interaction matrix**”에 저장된다.
- 상호작용 데이터가 많을 수록 추천이 더 정확하다.
- 그러나 content-based methods와 다르게 오직 user-item interaction matrix만 고려하기 때문에 데이터가 부족할 때, 제대로 추천해주지 못하는 문제가 발생한다.
-> **cold start problem!**
- **Memory based (Nearest neighbor based)**와 **Model based (Latent factor based)**로 크게 나눌 수 있다.

2.1 Memory based collaborative filtering methods

- Memory based 혹은 Nearest neighbor based라고 불리는데, 이 방법은 사용자-아이템 행렬만을 사용하며 최근접 이웃 탐색을 기반으로 한다.
- User-based (사용자 기반)**와 **Item-based(아이템 기반)**으로 나뉜다.

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

<- User-Item Interaction Matrix (사용자-아이템 행렬) 예시

2.1 Memory based collaborative filtering methods

User-based (사용자 기반)

- 사용자-아이템 행렬에서 공통의 아이템에 대해 사용자와 가장 비슷하게 평가한 사용자들을 찾고 그들에게 인기 있는 아이템을 사용자에게 추천해주는 방식

	Item A	Item B	Item C	Item D
User 1	3	4	4	
User2	4	4	4	5

➤ User1과 User2는 Item A~C에 대한 평가가 비슷하기 때문에 서로 유사한 사용자라고 할 수 있다.

➤ User1에게 Item D를 추천

- 사용자 간 유사도를 계산할 때, 공통으로 평가된 아이템의 개수에 주의해야 한다.
- ex) 사용자1과 2는 100% 유사도 / 사용자1과 3은 98% 유사도 ?

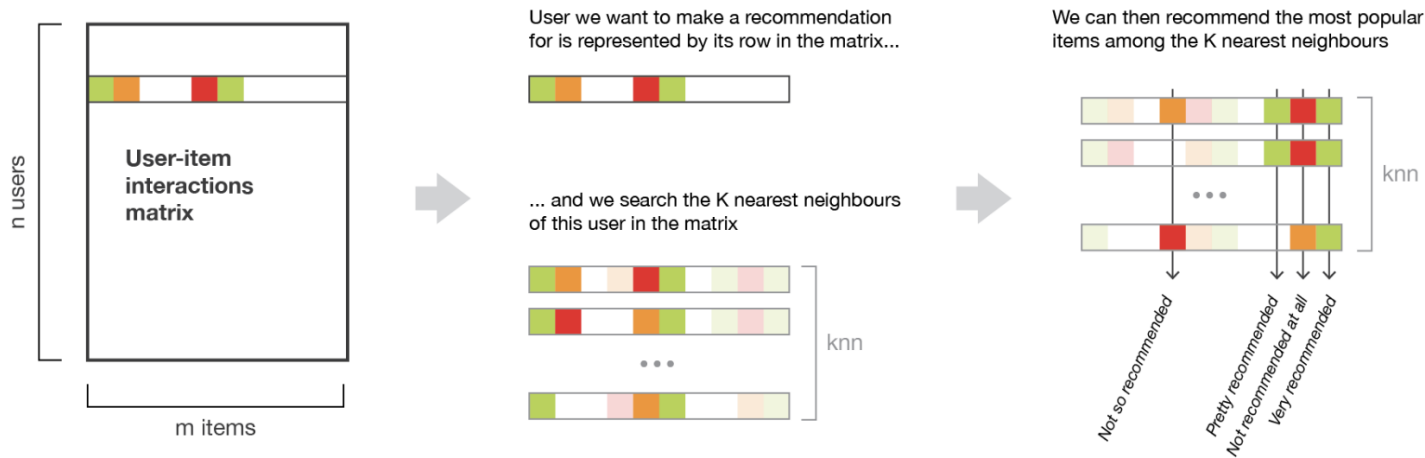
2.1 Memory based collaborative filtering methods

User-based (사용자 기반)

positive interactions

neutral interactions

negative interactions



2.1 Memory based collaborative filtering methods

Item-based (아이템 기반)

- 사용자가 선호하는 아이템에 대한 다른 사용자들의 평점과 유사한 평점을 가진 아이템을 찾아서 추천해주는 방식.

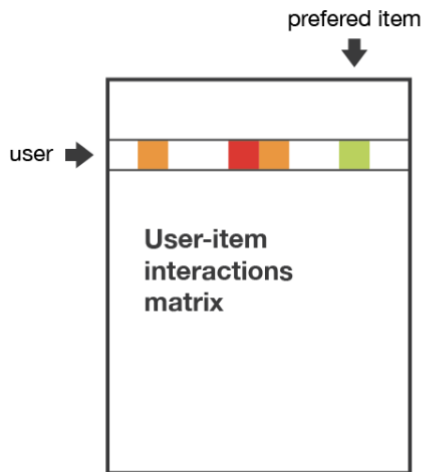
	User 1	User 2	User 3	User 4
Item A	5	4	4	
Item B	4	3	4	5

- Item A와 Item B는 비슷한 평점 분포를 가진다. 따라서 Item A와 Item B는 유사하다고 할 수 있다.
- User 4에게 Item B와 유사한 Item A를 추천

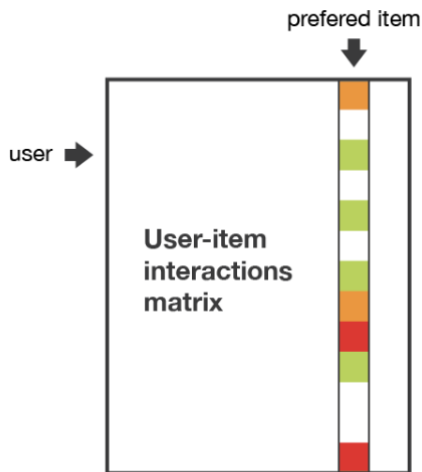
- 사용자가 가장 선호하는 아이템 1개만 보는 것 보다는 선호하는 아이템 여러 개를 고려해야 좀 더 정확한 추천이 가능하다.

2.1 Memory based collaborative filtering methods

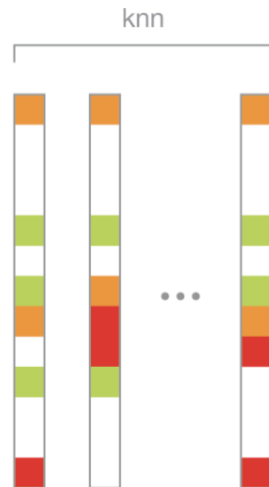
Item-based (아이템 기반)



We identify the preferred item of user we want to make recommendation for.



The preferred item is represented by its column in the matrix.



We can search and recommend the K nearest items to this “preferred item”

2.1 Memory based collaborative filtering methods

- User-based : more personalized / too sensitive (higher variance)
- Item-based : less personalized / lower variance

문제점

- 사용자와 아이템이 무수히 많을 경우에는 **계산량이 많아** 시간이 오래 걸린다.
- **“rich-get-richer”** : 인기 있는 아이템은 대부분 좋은 평점을 받기 때문에 소수의 인기 아이템만을 계속해서 추천한다.
- **“information confinement”** : 사용자가 선호하는 아이템과 유사한 아이템만 추천하기 때문에 사용자가 다른 종류의 아이템을 접할 가능성이 줄어든다.

2.2 Model based collaborative filtering methods

- Model based 혹은 Latent factor based
- 사용자-아이템 행렬을 설명하는 잠재요소가 있다고 가정하고 **차원축소**를 해서 모델로 하여금 잠재요소를 찾아내어 추천해준다.
- **잠재요소는 무엇인지 명확히 알 수 없다.** ex) 장르, 감독 등
- Content-based와 다른 점은 잠재요인이 될 수 있는 아이템의 특징이나 정보 등을 모델에 직접 주는 것이 아니라 모델 스스로 이 요인을 찾게끔 만든다는 것이다.
- **모델이 스스로 학습**해서 잠재요인을 추출하기 때문에 수학적으로 의미 있을 수는 있으나 사람의 직관적 해석이 어려울 수 있다. → “블랙박스 모델”

2.2 Model based collaborative filtering methods

특징

- User-Item Interaction matrix를 User-Factor 행렬과 Item-Factor 행렬로 분해
- 크고 빈 공간이 많은 원래의 matrix를 작은 두 개의 matrix로 분해하기 때문에 공간의 효율적 사용이 가능하다. (저장 공간 절약 가능)

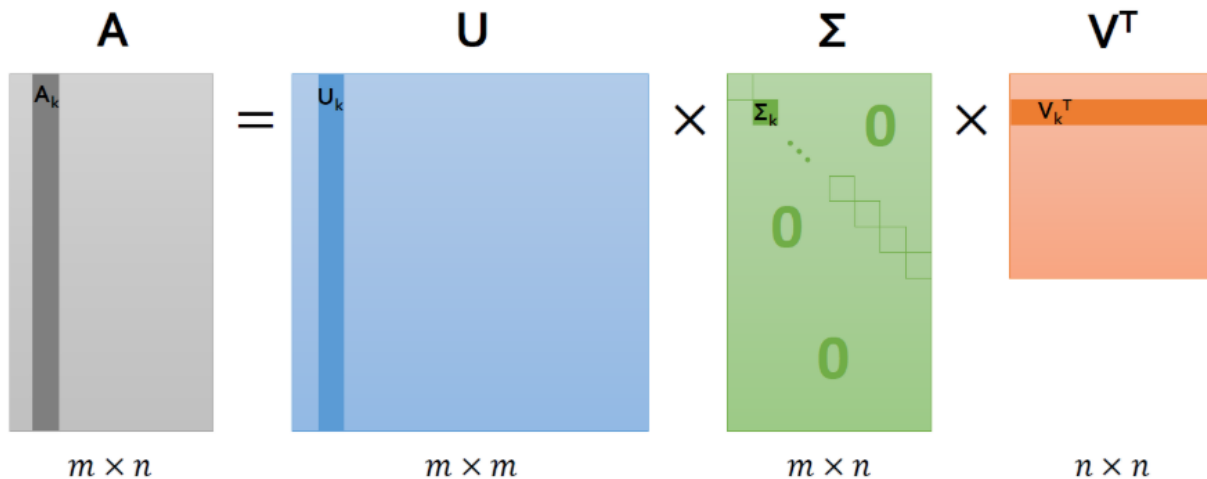
$$\mathbf{R} \approx \mathbf{P} \times \mathbf{Q}^T$$

- $\mathbf{R} = m \times n$ (User-Item matrix)
- $\mathbf{P} = m \times k$ (User-factor matrix)
- $\mathbf{Q} = n \times k$ (Item-factor matrix)

2.2 Model based collaborative filtering methods

SVD(특이값 분해)

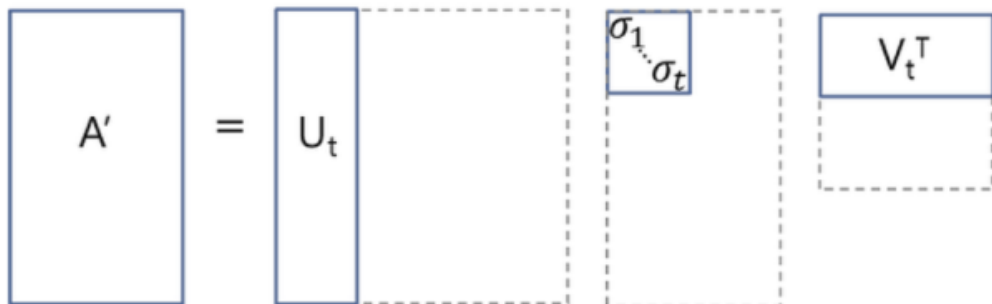
$$A = U\Sigma V^T$$



- Σ 는 AA^T , A^TA 를 고유값 분해해서 나오는 고유값들의 제곱근을 대각원소로 하는 $m \times n$ 대각행렬. 대각원소들은 A 의 특이값

2.2 Model based collaborative filtering methods

Truncated SVD



$$A_k = U_k \Sigma_k V_k^T$$

- 시그마 행렬의 대각원소(특이값) 가운데 상위 t개만 골라낸 형태
- 행렬 A를 똑같이 만들 수는 없지만 데이터 정보를 상당히 압축했는데도 불구하고 행렬 A에 근사하는 행렬 A'를 만들 수 있다.

2.2 Model based collaborative filtering methods

SVD 예시

$$A = U\Sigma V^T$$
$$\begin{bmatrix} 2 & 3 \\ 1 & 4 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.82 & -0.58 & 0 & 0 \\ 0.58 & 0.82 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5.47 & 0 & 0 & 0 \\ 0 & 0.37 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.40 & 0.91 \\ -0.91 & 0.40 \end{bmatrix}$$

truncated SVD 예시

$$A' = U_1 \Sigma_1 V_1^T$$
$$\begin{bmatrix} 1.79 & 4.08 \\ 1.27 & 2.89 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.82 \\ 0.58 \\ 0 \\ 0 \end{bmatrix} [5.47] \begin{bmatrix} 0.40 & 0.91 \end{bmatrix}$$

2.2 Model based collaborative filtering methods

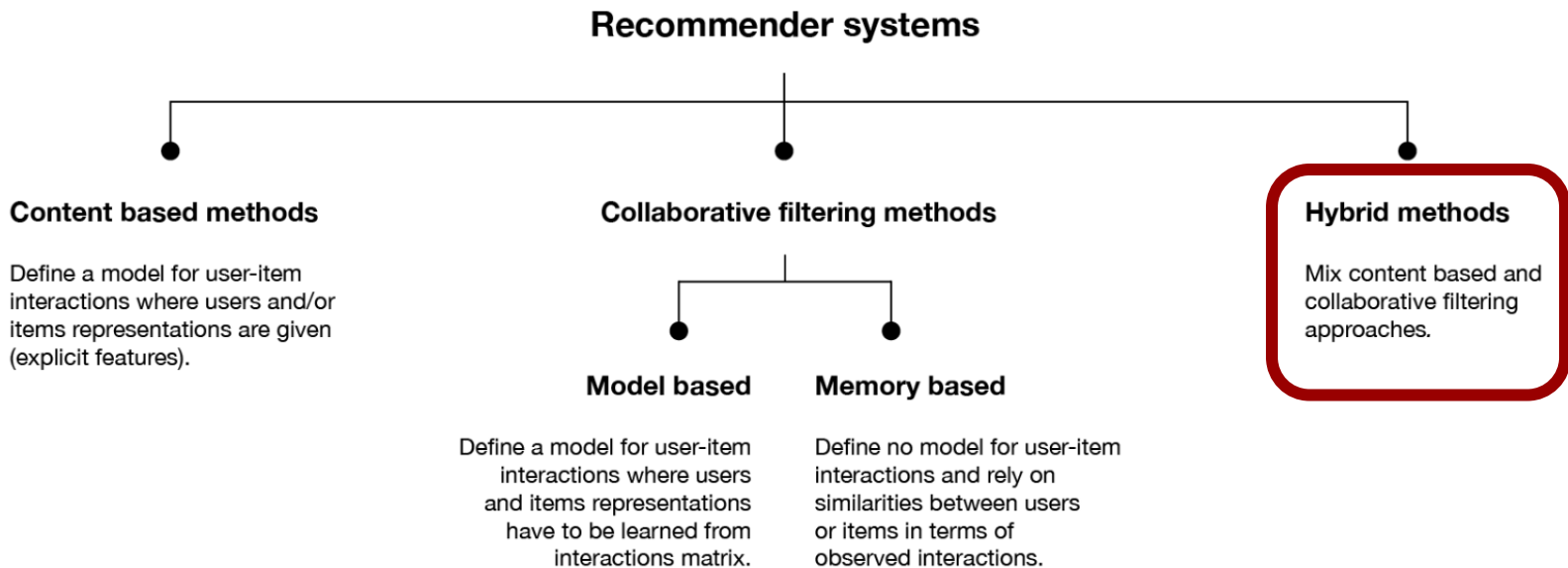
- M을 기존의 사용자-아이템 행렬이라고 하고, X를 사용자-잠재요인 행렬, Y를 아이템-잠재요인 행렬이라고 한다면

$$M \approx X.Y^T$$

$$(X, Y) = \underset{X, Y}{\operatorname{argmin}} \frac{1}{2} \sum_{(i, j) \in E} [(X_i)(Y_j)^T - M_{ij}]^2 + \frac{\lambda}{2} \left(\sum_{i, k} (X_{ik})^2 + \sum_{j, k} (Y_{jk})^2 \right)$$

- 경사하강법을 이용해서 M에 근사하게 하는 X와 Y를 구할 수 있다.

Next week



참고자료

<https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>

<https://lsjsj92.tistory.com/563?category=853217>

<https://ratsgo.github.io/from%20frequency%20to%20semantics/2017/04/06/pcasvdlsa/>

Thank you