협업필터링 이해

Writer	최윤서(학부학생/공과대학 도시공학)
키워드	
Reference	
주차	2주차

협업필터링

💡 **다른 사용자들로부터 얻은 정보**를 바탕으로 컨텐츠 추천

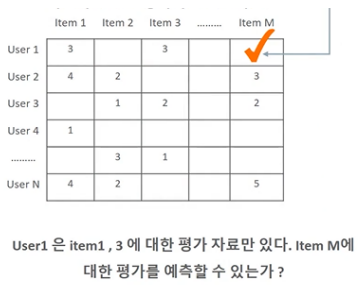
- 협업 필터링에서 데이터

  Explicit feedback 데이터: 유저가 자신의 선호도를 직접 표현한 데이터 (평점에 대한 정보가 포함됨)

- 작동 방식

  1. 사용자가 평가하지 않은 Item에 대한 평점을 예측.

     (다음과 같은 표에서 비어있는 값을 예측해 채워넣는다고 생각하면 됨)



	Item 1	Item 2	Item 3	.....	Item M
User 1	3		3		✓
User 2	4	2			3
User 3		1	2		2
User 4	1				
.....		3	1		
User N	4	2			5

User1 은 item1 , 3 에 대한 평가 자료만 있다. Item M에 대한 평가를 예측할 수 있는가 ?

  2. 높은 평점을 가질 아이템으로 예측되면 해당 상품을 추천.

- 최근접 이웃 vs 잠재요인 기반

최근접이웃기반

- **사용자 기반**

  나와 **취향이 가장 비슷한 사용자들**이 선호하는 상품을 추천

  = 당신과 비슷한 고객들이 다음 상품도 구매했습니다

	다크 나이트	인터스텔라	엣지오브투모로우	프로메테우스	스타워즈라스트제다이
사용자 A	5	4	4	추천	
사용자 B	5	3	4	5	3
사용자 C	4	3	3	2	5

- **아이템 기반**

  특정 **상품과 가장 유사한 평가를 받은 상품** 추천

  = 이 상품을 선택한 다른 고객들은 다음 상품도 구매했습니다

	사용자 A	사용자 B	사용자 C	사용자 D	사용자 E
세탁기 A	5	4	4	추천	
세제 A	5	3	4	5	3
세제 B	4	3	3	2	5

비슷한 평가를 받은 상품 → 비슷한 상품으로 간주

ex) 사용자기반으로 평점을 예측해보자.

1. user3과 **나머지 user와의** 유사도 구하기

	아이템1	아이템2	아이템3	아이템4	아이템5	아이템6	평균	Cosine(i, 3)	Pearson(i, 3)
사용자1	7	6	7	4	5	4	5.5	0.956	0.894
사용자2	6	7	?	4	3	4	4.8	0.981	0.939
사용자3	?	3	3	1	1	?	2	1.0	1.0
사용자4	1	2	2	3	3	4	2.5	0.789	-1.0
사용자5	1	?	1	2	3	3	2	0.645	-0.817

2. 나와 유사한 사용자들을 고려해 user3에서 비어있는 평점 예측하기

	아이템1	아이템6	평균	Pearson(i, 3)
사용자1	7	4	5.5	0.894
사용자2	6	4	4.8	0.939
사용자3	3.35	0.86	2	1.0
사용자4	1	4	2.5	-1.0
사용자5	1	3	2	-0.817

사용자1의 아이템1의 평점 - 사용자1의 평균 평점

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

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

사용자3의 평균 평점

원래 평점을 후하게 주는 user인지에 대한 정보를 제거하기 위하여 bias term 제거해준 이후에 평점 예측에 이용

ex) 아이템기반으로 평점을 예측해보자.

1. 평점을 예측하고자 하는 아이템(1, 6번)과의 유사도를 구하기

	아이템1	아이템2	아이템3	아이템4	아이템5	아이템6
사용자1	7	6	7	4	5	4
사용자2	6	7	?	4	3	4
사용자3	3	3	3	1	1	1
사용자4	1	2	2	3	3	4
사용자5	1	?	1	2	3	3
Cosine(1, j)	1	0.735	0.912	-0.848	-0.813	-0.990
Cosine(6, j)	-0.990	-0.622	-0.912	0.829	0.730	1

2. 유사한 아이템들을 고려해 user3에서 비어있는 평점 예측하기

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

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

잠재요인기반

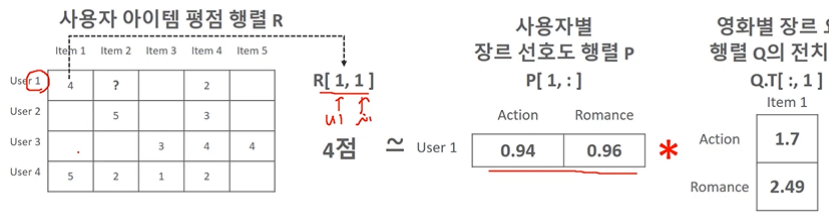
- 사용자-아이템 평점 행렬에 **잠재 요인**이 있다고 가정.
- 원본행렬을 **P와 Q로 분해**한뒤, 예측된 P와 Q를 기반으로 **재결합**하여 몰랐던 평점을 예측할 수 있음.

R (사용자-아이템 평점 행렬) ≃ Decomposed by P (사용자-잠재 요인 행렬) * Q^T (잠재 요인-아이템 행렬)

(m*n) = (m*k) * (k*n)

m = user 수 / n=아이템수 / k=잠재요인의 차원수(임의 설정)

- *ex) 사용자들의 영화 평점 행렬에 숨겨져 있는 잠재요인이 '장르'라고 한다면 사용자별 장르 선호도 행렬과 영화(아이템)별 장르 선호도 행렬을 기반으로 몰랐던 평점을 예측할 수 있는 것임*



- 원본 사용자-아이템 평점 행렬에는 비어져 있는 값이 많음. 그렇다면 어떻게 P행렬과 Q 행렬을 구하지?
- **확률적 경사 하강법(SGD)**
  - 목적함수

$$min \sum (r_{(u,i)} - p_u q_i)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

  - 원본 행렬 R과 추정한 P,Q를 결합한 R-hat과의 차이가 최소화되도록
  - 과적합 제어하기 위하여 L2 규제

- SGD를 통해 구하는 방법
  1. P와 Q 행렬을 임의의 값을 가진 행렬로 초기화
  2. P, Q를 통해 도출한 R-hat과 R과의 차이를 최소화할 수 있도록(+과적합제어) P와 Q를 업데이트

$$p_{u_{new}} = p_u + \eta(e_{(u,i)} * q_i - \lambda * p_u)$$
$$q_{i_{new}} = q_i + \eta(e_{(u,i)} * p_u - \lambda * q_i)$$

  3. 특정임계치 아래로 수렴할 수 있도록 P,Q 행렬을 계속해서 업데이트(SGD)
- SGD는 두 개의 행렬을 동시에 최적화

- **ALS(Alternating Least Squares)**
  - 둘 중 하나를 고정시키고 다른 행렬을 최적화 하는 방법.

    하나를 고정했을때는 convex 형태로 바뀌기 때문에 수렴된 행렬을 찾을 수 있다는 장점
  - how
    1. 모르는 정보는 모두 0으로 채워놓고 진행.
    2. 아이템 행렬 고정해두고 사용자 행렬 최적화
    3. 사용자 행렬 고정해두고 아이템 행렬 최적화
    4. 위의 두 과정을 계속 방법