2장 Neighborhood-based Collaborative Filtering

이마태

목차

1. 개요

✓Neighborhood-based Collaborative Filtering의 종류

- User-based

  자신과 **유사한 사용자**들이 매긴 점수 사용

- Item-based

  자신이 매긴 아이템 중 예측하려는 아이템과 **유사한 아이템**의 점수 사용

✓Neighborhood-based Collaborative Filtering 사용 목적

- Rating value 예측

  Ex) 왓챠 영화 평점 예측

- Top-k 아이템 / Top-k 유저

  Ex) 넷플릭스 컨텐츠 목록, 쿠팡 상품 목록, 페이스북 추천 친구

2. Rating 예측 방법

- User-based: user 간의 유사도
- Item-based: item 간의 유사도 계산
- Neighborhood 집단 결정. 집단 안에서의 가중 평균을 최종 예측에 사용

2. Rating 예측 방법 – User-Based

- 유저들이 점수를 매기는 scale이 다름
  - Ex) 전체적으로 좋아하는 경향 or 전체적으로 싫어하는 경향
- 유저 row별로 mean-centered된 점수 사용

Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6	Mean Rating
1	7	6	7	4	5	4	5.5
2	6	7	?	4	3	4	4.8
3	?	3	3	1	1	?	2
4	1	2	2	3	3	4	2.5
5	1	?	1	2	3	3	2

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

2. Rating 예측 방법 – User–Based

1. 타겟 유저(3번)와 다른 유저들 간의 유사도 계산
   - Top–2(k) closest users: 1, 2

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

2. Rating 예측 방법 – User–Based

2. k–neighbor(1,2번 user)의 가중 평균 계산하여 평점 예측

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

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

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

Item-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

2. Rating 예측 방법 – User–Based

3. raw rating 사용한 점수 / mean–centered rating 사용한 점수 비교

Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6
1	7	6	7	4	5	4
2	6	7	?	4	3	4
3	?	3	3	1	1	?
4	1	2	2	3	3	4
5	1	?	1	2	3	3

Raw

Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6
1	7	6	7	4	5	4
2	6	7	?	4	3	4
3	6.49	3	3	1	1	4
4	1	2	2	3	3	4
5	1	?	1	2	3	3

Mean–centered

Item-Id ⇒ / User-Id ⇓	1	2	3	4	5	6
1	7	6	7	4	5	4
2	6	7	?	4	3	4
3	3.35	3	3	1	1	0.86
4	1	2	2	3	3	4
5	1	?	1	2	3	3

2. Rating 예측 방법 – User-Based

- Raw rating 사용한 방법
  - 각 유저의 점수 scale을 반영하지 않음
- Mean-centered rating 사용한 방법
  - 해당 유저의 scale을 반영
  - Allowed range 밖의 값(0.86)이 나올 수 있음

3. User-Based, Item-Based 비교

- Item-based
  - Robust
    - User 수 ⟨⟨ Item 수 → User-based는 유저의 rating 변화에 민감
  - 유저에게 설명하기 용이: "태극기 휘날리며를 시청하여, 실미도 추천"
- User-based
  - 다양한 추천 결과 제공
  - 유저에게 설명하기 애매함. 자기와 관련된 유저의 profile을 제공할 수는 없기 때문

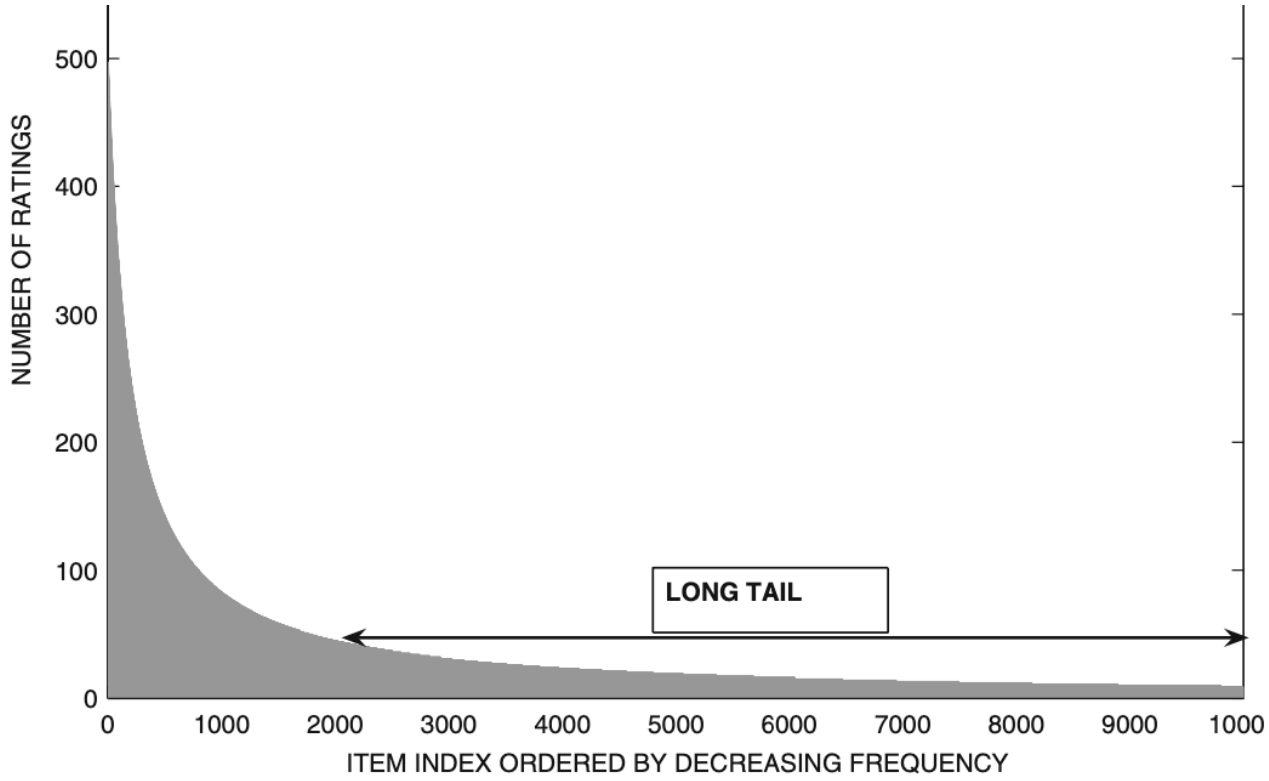
4. Neighborhood-based CF의 강/약점

- 강점
  - 직관적. 설명하기 쉬움 (model-based는 그렇지 않음)
  - 새로운 유저나 아이템이 추가되어도 stable
- 약점
  - offline phase에서 시간 복잡도가 커짐 (online에서는 항상 효율적)
    - offline phase: User간 / Item 간 유사도 계산
    - online phase: 가중 평균을 이용한 평점 예측
  - sparsity 문제
    - user-based에서 나의 이웃들 중 타겟 영화를 **평가한 사람이 한 명도 없다**면, rating을 예측할 수가 없음
    - 한편으로는 좋지 않은 추천이라는 의미도 됨

5. 질문 & 디스커션

High-frequency items

- 인기 item vs 비인기 item
- 낮은 수익성 vs 높은 수익성
- Long tail에 있는 비인기 아이템은 마케팅 비용, 분배 비용 낮음[1]
- 검색 엔진, 가격 비교 엔진 등 트래픽 비용이 낮음[2]
- 많은 추천 알고리즘은 인기 item을 추천해주는 경향 → 다양성 없음



참고 문서

[1] https://www.investopedia.com/terms/l/long-tail.asp

[2] https://www.practicalecommerce.com/Amazon-Does-Not-Do-Long-tail-Why-Should-You

감사합니다