Report

2016025196

김동규

1. 개발 환경

Windows 10, python 3.9.2, numpy 1.19.5

2 알고리즘

주요 알고리즘으로는 cosine similarity를 사용하였다. 초기에는 사용자간의 유사도를 사용하였으나, 여러 실험 결과 컨텐츠 간의 유사도를 통해 추천 점수를 찾는 것이 정확도가 더 높은 것을 알 수 있었다.

User-Content 점수 DB 생성 알고리즘

User id와 content id를 축으로 점수를 저장하는 행렬을 생성한다. 이때, 메모리를 소모하더라도 속도를 위해 최대 id숫자만큼 index를 생성한다.

Similarity 측정

Content간의 유사도 측정을 위해 DB의 content index 크기로 2차원 행렬을 생성한다. 이후, 서로 다른 index 사이의 cosine similarity만 계산한다.

Prediction

User line중 목표 item과 가장 유사도가 높은 컨텐츠에 준 점수를 참고하여 점수를 준다.

3. 코드

텍스트이(가) 표시된 사진

자동 생성된 설명

open\_file - input파일을 읽어서 [user\_id, item\_id, rating]을 갖는 리스트를 생성한다.

Preprocessing – open\_file에서 생성된 리스트를 (max user index, max item index) 크기의 numpy 행렬로 전처리한다.

Class USER\_DB – DB와 similarity matrix를 갖는 클래스이다.

similarity – DB에서 각 content사이의 cosine similarity를 계산하는 함수이다.

predict – 주어진 user id의 item id에 대한 rating을 예측하는 함수이다. similarity matrix를 참고하여 예측한다. 기존에 없던 item id가 들어오면 3으로 예측하고, user id가 들어오면 해당 item에 대한 최빈값으로 예측한다.

4. 실행 방법

cmd 명령어로 recommender.py u#.base u#.test 를 사용한다.



5. 실행 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

1차(v0) - user간의 단순한 cosine similarity를 이용하였다.

2차(v1) – 겹치는 attribute로만 유사도를 구하였다.

3차(v1-2) – max similarity들의 평균으로 평가를 내리도록 수정했다.

4차(v0-1) – new user에 대해 처리하는 코드를 추가했다.

5차(v0-2) – max similarity 평가의 평균으로 예측, max similarity=0일 때(겹치는 attribute가 없을 때)처리하는 코드를 추가했다.

여기까지 실험하면서 예외처리를 제외하고 cosine similarity에 기능을 추가할수록 성능이 떨어진 다는 것을 알 수 있었다. 그렇기에 v0에서 item간의 유사도를 이용하도록 수정해보았다.

6차(v2) - item간의 cosine similarity를 이용하도록 수정했다.

7차(v2-1) – 겹치는 attribute로만 유사도를 구하도록 수정했다.

이때에도 v2보다 v2-1이 성능이 떨어지는 것을 확인하고 v2를 최종 프로그램으로 선정하였다.