Implicit data를 이용한 VOD 컨텐츠 추천 시스템 개발

(VOD contents recommendation system using implicit data)

지도교수 : 박근수

이 논문을 공학학사 학위 논문으로 제출함.

2018년 6월 27일

서울대학교 공과대학

컴 퓨 터 공 학 부

설 재 완

2018년 8월

초 록

인터넷의 발달 덕분에 우리는 선택의 폭이 넓어졌다. 과거에 직접 이동을 하며 상품을 확인했던 것과 달리 오늘날에는 인터넷만 연결이 되어있다면 수많은 상품을 직접 검색하고 선택할 수 있다. 하지만 선택의 폭이 늘어남에 따라 직접 모든 상품을 확인 후 선택을 하는 것이 불가능해졌고 그 결과 추천시스템의 중요성이 점점 커지게 되었다.

현재 explicit data(사용자가 직접 내린 선호도)를 이용한 추천알고리즘은 이미 여러 가지가 알려져 있고 활발하게 연구되고 있지만 implicit data(사용자가 직접 내린 선호도 외의 정보)를 이용한 알고리즘은 아직 많지 않다.

본 연구에서는 implicit data라고 할 수 있는 VOD 시청기록을 일련의 과정을 거쳐 explicit data인 평점으로 환산한 후 기존의 explicit data를 이용한 추천알고리즘을 적용한다. 이로써 implicit data를 이용한 추천 알고리즘의 한 방향을 제시한다.

주요어 : implicit data, recommendation system

목 차

1. 서론

우리는 어떤 상품을 구매할 때 선택을 하게 된다. 인터넷의 발달로 인하여 선택의 폭이 더욱 넓어졌다. 같은 기능을 하는 상품의 종류는 무수히 많으며 우리는 이 수많은 상품 중에서 선택을 해야 한다. 2017년 12월 Netflix에서는 미국 region을 기준으로 약 5600여 개의 콘텐츠를 제공한다.[1] 이 콘텐츠들을 직접 확인한 후 필요한 콘텐츠를 고르는 것은 불가능에 가깝다. 상품의 종류와 개수가 늘어날수록 추천시스템의 중요성은 더욱 커진다.

Cast is는 국내 최대의 VOD 콘텐츠를 제공하는 회사이다. 본 글은 창의적 통합설계과목에서 cast is와 3개월간 수행한 과제의 결과물이다. 주어진 과제는 3개월간의 시청기록을 기반으로 VOD 콘텐츠 추천 시스템을 개발하는 것이다.

오늘날의 VOD 콘텐츠 제공 시스템은 단순히 TV 다시 보기 뿐만 아니라 영화, 애니메이션, 다큐멘터리 등 다양한 영상콘텐츠를 제공한다. 특히 상품확인이 곧 소비가 되는 영상콘텐츠의 특성상 추천의 중요성을 다른 상품들보다 더욱 크다. Netflix와 같은 대규모 영상콘텐츠 회사에서도 추천시스템을 운용하고 있는데 이곳에서는 시청자들이 매긴 별점 또는 평점을 기반으로 추천을 진행하고 있다.[2] 하지만 국내의 VOD 콘텐츠 시장에서는 별점으로 후기를 받지 않는 곳이 많다. 따라서 별점이 없는 상황에서의 콘텐츠 추천시스템을 개발해야 한다.

우리는 시청기록 데이터를 이용하여 사용자의 VOD 시청 시간과 시청 패턴을 끌어내고 이를 바탕으로 가상의 별점 데이터를 만들었다. 가상의 별점 데이터가 생기면 기존의 추천알고리즘에 적용하여 추천을 할 수 있다. (기존의 추천알고리즘이 잘 동작한다고 가정하면 가상의 별점 데이터가 얼마나 실제 시청자의 선호를 잘 표현하는지가 중요하다.)

2. 관련연구

2.1. 유사도

상품 또는 사용자가 얼마나 비슷한지 계산하기 위하여 유사도라는 개념을 도입한다. 사용자가 상품에 내린 평점을 이용하여 사용자와 상품을 1차원 vector로 간주하고 두 1차원 vector 간의 유사도를 이용하여 상품과 상품이 또는 사용자와 사용자가 얼마나 유사한지 확인할 수 있다.

2.1.1. 자카드 유사도(Jaccard similarity)

자카드 유사도는 두 집합 사이의 유사도를 측정하는 방법의 하나다.[3] 자카드 유사도는 0~1의 값을 가지며 값이 1에 가까울수록 유사하다고 할 수 있다. 자카드 유사도는 다음의 수식으로 계산할 수 있다.

$$sim(A,\ B) = \frac{|A \cap B|}{|A \cup B|}$$

우리의 경우에는 A, B를 1차원 벡터로 보고 평점을 내린 경우 1, 평점을 내리지 않은 경우 0으로 간주하여 A와 B 사이의 유사도를 구할 수 있다. 다만 평점의 존재 여부만을 이용하여 유사도를 구하므로 평점의 값을 반영할 수 없는 단점이 있다.

2.1.2. 코사인 유사도(Cosine similarity)

코사인 유사도는 두 벡터 간 각도의 코사인값을 이용하여 측정하는 유사도이다.[4] 주로 0~1의 값을 이용하며 0의 경우 완전독립, 1의 경우 완전동일, -1의 경우 완전 반대로 해석할 수 있다. 코사인 유사도는 다음의 수식으로 계산할 수 있다.

$$sim(A,\ B) = cos(\Theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

우리의 경우 평점을 내리지 않은 경우 0, 평점을 내린 경우 평점 값으로 평점 벡터를 만들어서 코사인 유사도를 구할 수 있으며 이 경우 낮은 평점이 높은 평점에 비교하여 너무 큰 영향력을 주는 단점이 있다.

2.1.3. 피어슨 상관계수(Pearson correlation coefficient)

피어슨 상관계수는 두 변수 간의 관련성을 구하기 위해 보편적으로 사용된다.[5] 코사인 유사도와 마찬가지로 −1에 가까울수록 음의 상관관계(반대), 0에 가까울수록 무시할 수 있는 선형관계(독립), 1에 가까울수록 양의 상관관계(유사)로 해석할 수 있다. 피어슨 상관계수는 다음의 수식으로 구할 수 있다.

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

우리의 경우에는 사용자의 유사도 비교 시에는 두 사용자가 모두 평점을 내린 상품의 평점 값만, 상품과 상품의 유사도 비교 시에는 두 상품 모두에 평점을 내린 사용자의 평점 값을 이용한다. 우리의 주제에서는 이 피어슨 상관계수를 이용한다.

2.2. 추천 알고리즘

2.2.1. baseline 알고리즘

baseline은 사용자와 상품의 평균적인 경향을 고려한 방식이다. 전체 별점의 평균, 특정 사용자가 내린 전체 평점의 평균, 특정 상품의 전체 평점의 평균을 이용하여 사용자 A의 상품 X에 대한 평점을 예측한다. 이 경우 각각의 사용자와 상품 사이의 관계를 고려하지 못하는 단점이 있다.

2.2.2. 협력 필터링(Collaborative filtering)

협력 필터링은 많은 사용자로부터 얻은 상품에 대한 선호 정도에 따라 사용자의 관심상품을 자동으로 예측하게 해주는 방법이다.[6] 협력 필터링에서는 사용자의 과거 행동의 경향성이 미래에도 유지될 것이라는 가정에서 출발한다. 예를 들어 A라는 사람이 X라는 상품에 대하여 좋은 평가를 한다면 미래에 상품 X와 유사한 X'에 대해서도 좋은 평가를 할 것이라는

것이다. 협력 필터링에는 크게 상품과 상품의 유사도를 기반으로 하는 추천 방법과 사용자와 사용자의 유사도를 기반으로 하는 추천방법이 있다. 우리의 경우에 사용자는 시간이 지날수록 계속 증가하는데 이에 따른 scalability를 감당할 수 없으므로 상품과 상품의 유사도를 이용한다.

2.2.3. matrix factorization

matrix factorization에서는 사용자와 상품을 몇 차원의 수치화된 값으로 나타낼 수 있다고 가정한다. 전체 사용자 M명과 상품 N개의 M x N의 matrix를 F개의 차원을 이용하여 M x F matrix와 F x N matrix의 곱으로 분리한다. 각각의 상품과 사용자는 F차원의 벡터로 볼 수 있으며 사용자 A의 상품 X에 대한 예상 평점은 A벡터와 X벡터의 내적으로 계산할 수 있다.

3. 구조 및 설계

우리는 3개월(7, 8, 9월)의 VOD 콘텐츠 시청기록을 이용하여 사용자에게 VOD 콘텐츠를 추천해야 한다. 이를 위해 먼저 3개월의 시청기록을 앞의 2개월과 뒤의 1개월로 나눈다. 그 후 앞의 2개월 시청기록을 이용하여 협력 필터링 모델과 matrix factorization 모델을 트레이닝시킨다. 트레이닝 시킨 모델과 뒤의 1개월 시청기록을 이용하여 모델의 성능을 평가한다.

3.1. 별점데이터 생성

협력 필터링 모델과 matrix factorization 모델을 트레이닝 시키는 데에는 별점 데이터가 필요하다. 시청기록 그 자체는 별점이 아니므로 직접 모델 트레이닝에 사용할 수 없다. 그래서 일련의 과정을 거쳐 시청기록을 별점 데이터로 환산하였다.

평점 환산은 2가지 방법을 이용하였다. 첫 번째는 시청시간의 순서대로 1~5 사이의 별점을 linear 하게 환산하는 방법이고 두 번째는 Log(시청시간 / 시청시간의 median) 값을 취한 후 이 값의 범위가 1~5 사이가 되도록 적절한 수를 더해주는 방법이다.

3.2. 추천알고리즘의 선택

여러 가지 추천알고리즘의 성능을 평가하기 위해 실제 Netflix prize에서 사용한 별점 데이터를 이용하였다. 각각의 모델을 training set을 이용하여 훈련한 후 test set의 별점의 값을 예측한 후 RMSE(평균 제곱근 편차)를 비교한다. 그 결과 RMSE가 가장 낮은 모델을 이용한다.

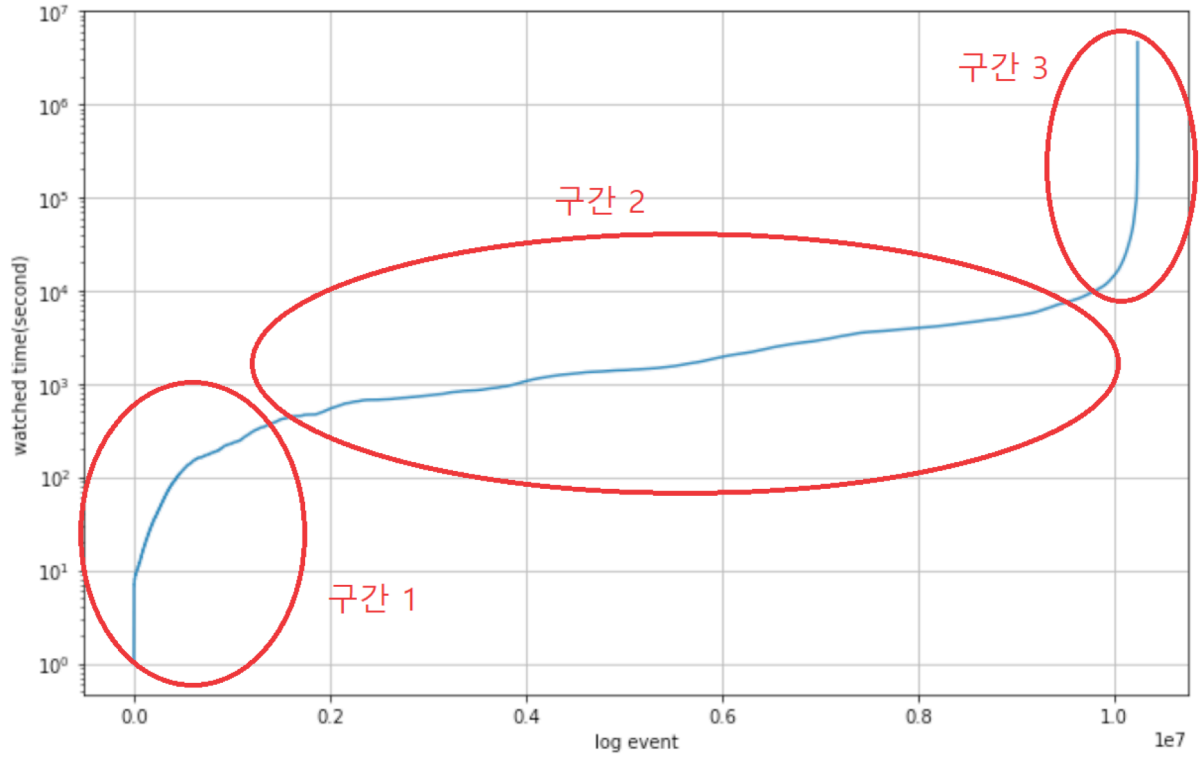
4. 평가 및 결과

4.1. 별점데이터 생성

우리의 시청기록은 총 약 550000명의 사용자와 약 50000개의 VOD 콘텐츠에 대하여 약 2300만 건의 기록으로 이루어져 있다. 이 시청기록에 대하여 두 가지 별점 환산방법을 적용하는 과정에서 시청기록 데이터의 특성에 따른 문제들이 있었다.

첫 번째로 유효하지 않은 데이터의 비율이 높았다. 똑같은 시청기록이 중복으로 기록된 경우, 시청 시작시각이 시청 마감시각보다 더 늦은 경우 등 9.46%의 시청기록이 유효하지 않아서 시청기록에서 제거하였다. (23,451,732 -> 21,234,059) 이 데이터를 (사용자, VOD)로 group을 지어서 시청시간의 합을 구하였다. (21,234,059 -> 10,252,942) 그 후 VOD 시청시간의 단위(시 또는 분)의 통일되지 않아서 유효하지 않은 데이터 2.3%를 제거하였다. (10,252,942 -> 10,236,116) 여기서 시청시간의 합이 0인 잘못된 시청기록 또한 제거하였다. (10,236,116 -> 10,000,000) 마지막으로 한 시청자가 동일 VOD를 40회 이상(최대 380여 회)보는 기록도 유효하지 않은 것으로 판단하였고 이는 약 2.5%에 해당하는 수치이다. (10,000,000 -> 9,746,638)

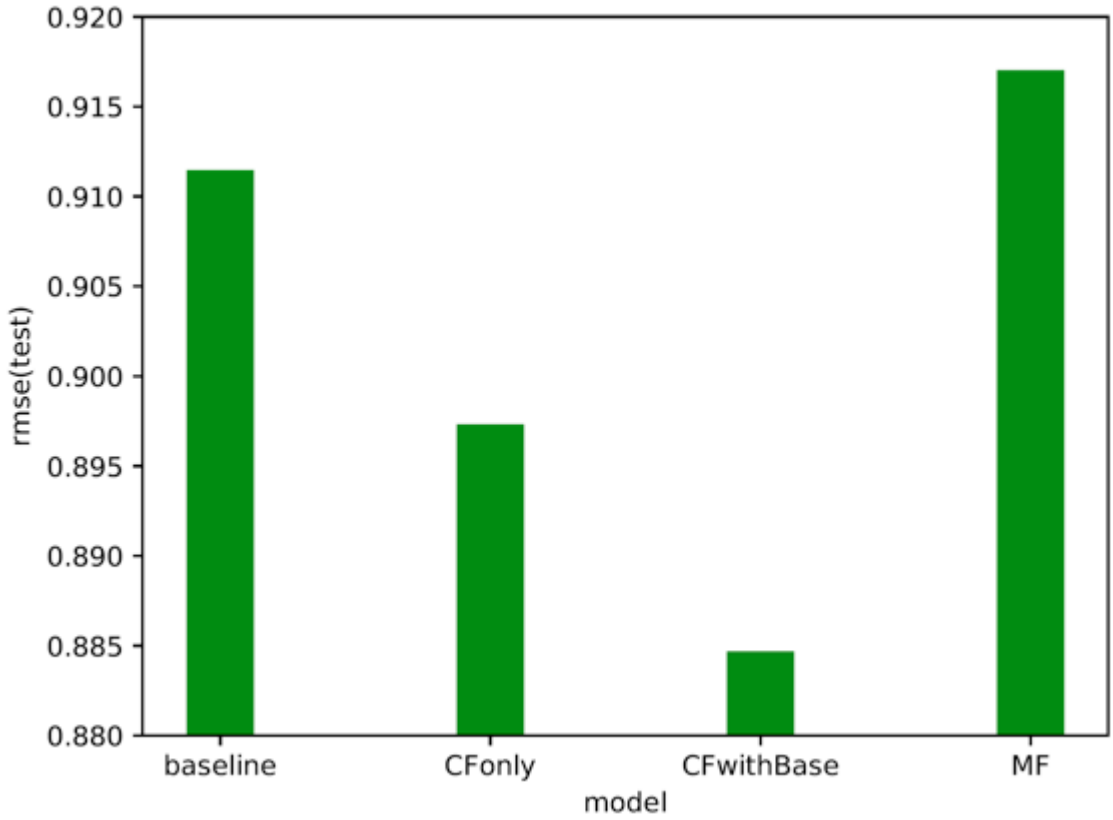
우리의 시청기록데이터에서는 log를 이용한 평점 환산방법의 경우가 적절했다.



그림에서 확인할 수 있듯이 시청시간의 분포가 선형적이지 않아서 첫 번째 환산방법인 단순 시청기록의 순서로 별점을 환산할 경우 현실 왜곡이 컸다.

4.2. 추천알고리즘의 선택

Netflix prize에 사용된 별점 데이터를 이용하여 다음의 4가지 추천 알고리즘의 성능을 비교해보았다.



각각의 RMSE(평균 제곱근 편차)는 그래프에서 확인할 수 있으며 CF(협력 필터링)와 baseline 모델을 결합한 모델이 가장 좋은 성능을 보였다.

본 프로젝트의 구현은 크게 2부분으로 이루어져 있다. Raw data의 처리와 model training이 그것이다. 협력 필터링(collaborative filtering)의 구현은 본 저자가 구현하였으며 raw data의 처리와 matrix factorization의 구현은 타 팀원에 의해 구현되었다.

5. 결론 및 마무리

log를 이용한 별점 환산방법과 CFwithBase(협력 필터링과 baseline의 결합)알고리즘을 이용하여 실제 7, 8월 시청기록 데이터로 트레이닝을 시킨 후, 전체 VOD 중 예상 평점이 높은 상위 20%의 VOD를 9월에 시청하리라 예측한다. 그 후 실제 9월에 시청한 VOD 콘텐츠 중 예상한 VOD가 어느 정도의 비율로 포함되어있는지를 확인해 보았다. 그 결과 평균적으로 47.4%에 해당하는 콘텐츠가 실제 9월에 본 VOD 콘텐츠에 포함되어 있었다.

이 과제에서 결과에 영향을 가장 크게 주는 부분은 시청기록 데이터를 평점으로 환산하는 부분이다. 실제로 우리는 2가지 방법을 이용하여 평점 데이터를 만들었지만, 시청시간 및 시청 패턴을 다각적으로 분석한다면 더욱 좋은 환산방법이 있을 것이다. 또한 실제 VOD 콘텐츠의 내용을 확인해본바 대부분이 가정에서 시청하는 유아용 VOD 콘텐츠였다. 우리의 환산방법을 더 큰 규모의 다양한 시청기록을 가진 데이터에 적용한다면 결과가 달라질 수도 있다.

이 과제에서 우리는 implicit data(사용자가 직접 평가한 점수 외의 다른 정보)를 이용하여 추천알고리즘을 개발하는 한 가지 가능성을 제시하였다. 특히 우리는 시청시간만을 이용하여 추천알고리즘을 개발하였지만, 시청 시간대, 시청 패턴 등의 다른 정보를 결합한다면 더욱 성능이 좋은 추천알고리즘을 개발할 수 있을 것이다.

6. 참고문헌

[1] Netflix International: What movies and TV show can I watch, and where can I watch then?, https://www.finder.com/global-netflix-library-totals, Accessed: 2018-06-18.

[2] CARLOS A. GOMEZ-URIBE and NEIL HUNT, Netflix, Inc. The Netflix Recommender System: Algorithms, Business Value, and Innovation

[3] Jaccard Index, https://en.wikipedia.org/wiki/Jaccard_index, Accessed: 2018-06-18.

[4] Cosine similarity, https://en.wikipedia.org/wiki/Cosine_similarity, Accessed: 2018-06-20.

[5] Pearson correlation coefficient, https://en.wikipedia.org/wiki/Pearson_correlation_coefficient, Accessed: 2018-06-20.

[6] 협력필터링, https://ko.wikipedia.org/wiki/%ED%98%91%EC%97%85_%ED%95%84%ED%84%B0%EB%A7%81, Accessed: 2018-06-21.