Euron_ML_Week16

9. 추천시스템

01. 추천 시스템의 개요와 배경

- 추천 시스템의 개요

: 하나의 콘텐츠를 선택했을 때 선택된 콘텐츠와 연관된 추천 콘텐츠가 얼마나 사용자의 관심을 끌고 개인에게 맞춘 콘텐츠를 '추천'했는지는 사이트의 평판을 좌우하는 중요요소

: 사용자 자신도 좋아하는지 몰랐던 취향을 시스템이 발견하고 그에 맞는 콘텐츠를 추천해주는 것이 추천 시스템의 묘미

: 사이트를 더 강하게 신뢰하게 되는 사용자

- 온라인 스토어의 필수 요소, 추천 시스템

: 특히 온라인에서 진가를 발휘하는 추천 시스템

: 사용자가 무엇을 원하는지 빠르게 찾아내 사용자의 온라인 쇼핑의 즐거움을 증가시킴

- 추천 시스템의 유형

- 콘텐츠 기반 필터링(Content based filtering)
- 협업 필터링(Collaborative Filtering)
    - 최근접 이웃 협업 필터링
    - 잠재 요인 협업 필터링

---

02. 콘텐츠 기반 필터링 추천 시스템

: 사용자가 특정한 아이템을 매우 선호하는 경우, 그 아이템과 비슷한 콘텐츠를 가진 다른 아이템을 추천하는 방식

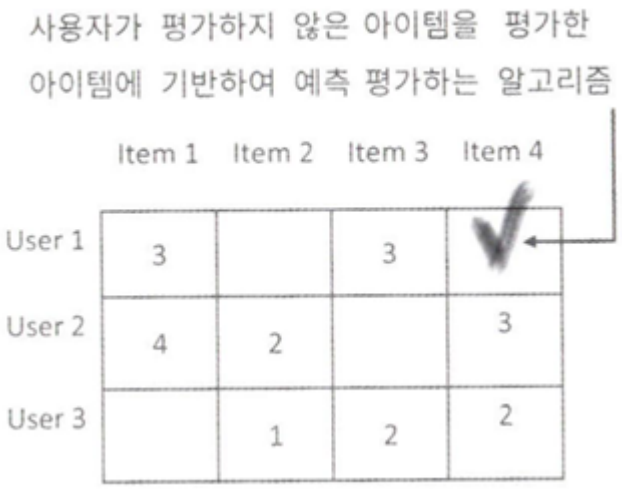
---

03. 최근접 이웃 협업 필터링

: 친구에게 물어보는 것과 유사한 방식으로, 사용자가 아이템에 매긴 평점 정보나 상품 구매 이력과 같은 사용자 행동 양식만을 기반으로 추천을 수행하는 것

- 주요 목표

  : 사용자-아이템 평점 매트릭스와 같이 축적된 사용자 행동 데이터를 기반으로 **사용자가 아직 평가하지 않은 아이템을 예측 평가**하는 것



user 1은 item 4에 대한 평점이 없음. 협업 필터링은 사용자가 평가한 다른 아이템을 기반으로 사용자가 평가하지 않은 아이템의 예측 평가를 도출하는 방식

- 사용자-아이템 평점 행렬 데이터에만 의지해 추천을 수행
  - 행: 개별 사용자
  - 열: 개별 아이템

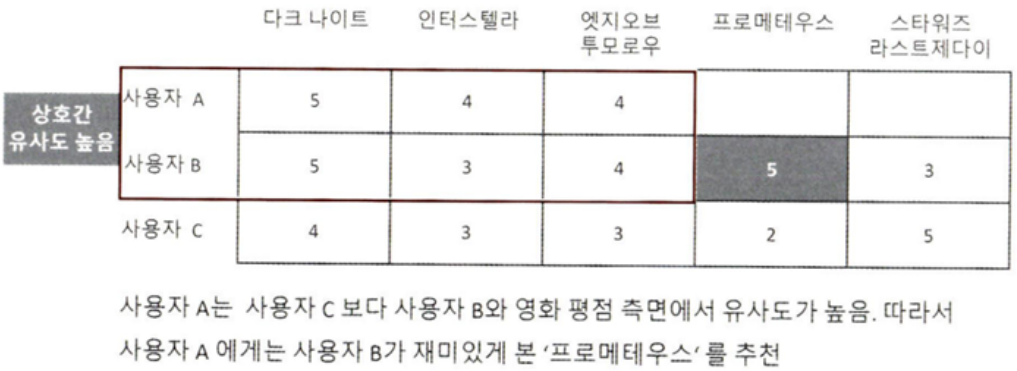
  ⇒ 많은 아이템을 열로 가지는 다차원 행렬이며, 사용자가 아이템에 대한 평점을 매기는 경우가 많지 않아 희소 행렬 특성을 가지고 있음

- 메모리 협업 필터링 이라고도 함
  - 사용자 기반

    : 당신과 비슷한 고객들이 다음 상품도 구매했습니다

    : 특정 사용자와 유사한 다른 사용자를 TOP-N으로 선정해 이 사용자들이 좋아하는 아이템을 추천하는 방식

: 특정 사용자와 타 사용자 간의 유사도를 측정한 뒤 가장 유사도가 높은 TOP-N사용자를 추출해 그들이 선호하는 아이템 추천



		다크 나이트	인터스텔라	엣지오브 투모로우	프로메테우스	스타워즈 라스트제다이
상호간 유사도 높음	사용자 A	5	4	4		
	사용자 B	5	3	4	5	3
	사용자 C	4	3	3	2	5

사용자 A는 사용자 C 보다 사용자 B와 영화 평점 측면에서 유사도가 높음. 따라서 사용자 A 에게는 사용자 B가 재미있게 본 '프로메테우스' 를 추천

⇒ 사용자 A와 유사도가 높은 사용자 B가 재밌게 관람한 '프로메테우스'를 추천하는 것이 사용자 기반 최근접 이웃 협업 필터링

- 아이템 기반

  : 이 상품을 선택한 다른 고객들은 다른 상품도 구매했습니다

  : 아이템이 가지는 속성과는 상관 없이 사용자들이 그 아이템을 좋아하는지/ 싫어하는지의 평가 척도가 유사한 아이템을 추천하는 기준이 됨

		사용자 A	사용자 B	사용자 C	사용자 D	사용자 E
상호간 유사도 높음	다크 나이트	5	4	5	5	5
	프로메테우스	5	4	4		5
	스타워즈 라스트제다이	4	3	3		4

여러 사용자들의 평점을 기준으로 볼 때 '다크 나이트' 와 가장 유사한 영화는 '프로메테우스'

⇒ '다크나이트'를 매우 좋아하는 사용자 D에게 아이템 기반 협업 필터링은 D가 아직 관람하지 못한 '프로메테우스'와 '스타워즈-라스트 제다이' 중 '프로메테우스'를 추천

➕

- 일반적으로 사용자 기반보다는 아이템 기반보다는 아이템 기반 협업 필터링이 정확도가 더 높다

  → 비슷한 상품을 구입한다고 해서 사람들의 취향이 비슷하다고 판단하기는 어려운 경우가 많기 때문

- 추천 시스템의 유사도 측정에 '코사인 유사도'가 가장 많이 적용됨

  → 추천 시스템에 사용되는 데이터는 피처 벡터화된 텍스트 데이터와 동일하게 **다차원 희소 행렬**이라는 특징이 있으므로 유사도 측정을 위해 주로 이용

---

04. 잠재 요인 협업 필터링

- 잠재 요인 협업 필터링의 이해

: 사용자-아이템 평점 매트릭스 속에 숨어 있는 잠재 요인을 추출해 추천 예측을 할 수 있게 하는 기법

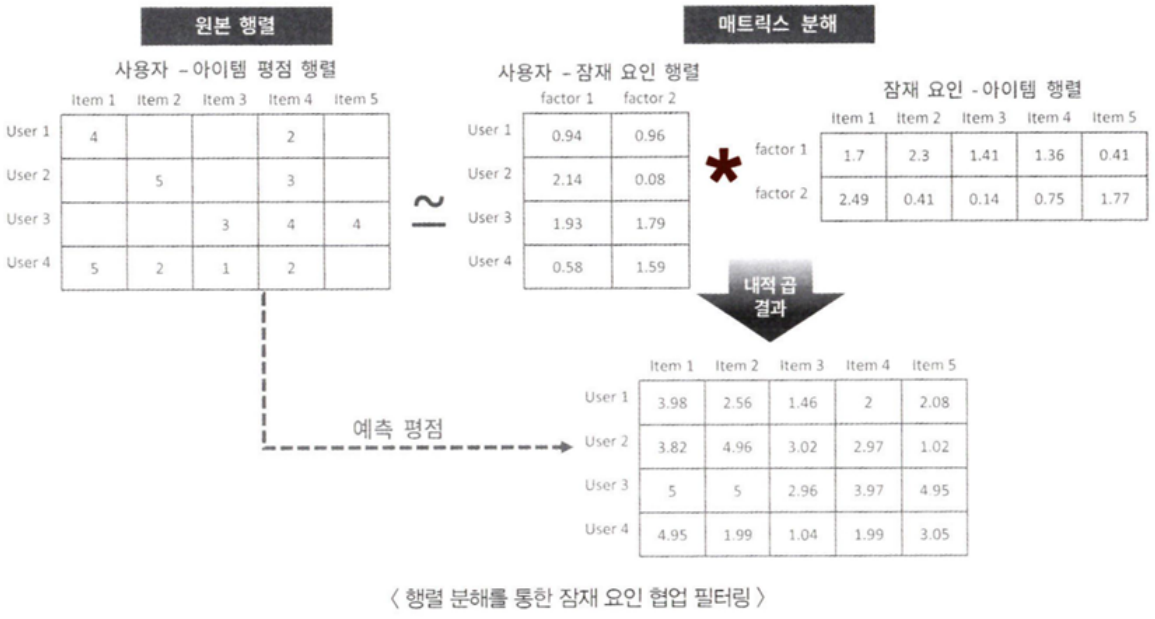
: 행렬 분해 기반의 잠재 요인 협업 필터링

📎 행렬 분해

: 대규모 다차원 행렬을 SVD와 같은 차원 감소 기법으로 분해하는 과정에서 잠재 요인 추출

: '잠재 요인'을 기반으로 다차원 희소 행렬인 사용자-아이템 행렬 데이터를 저차원 밀집 행렬의 사용자-잠재 요인 행렬과 아이템-잠재 요인 행렬의 전치 행렬로 분해 가능.

이렇게 분해된 두 행렬의 내적을 통해 새로운 예측 사용자-아이템 평점 행렬 데이터를 만들어 사용자가 아직 평점을 부여하지 않는 아이템에 대한 예측 평점을 생성.



〈 행렬 분해를 통한 잠재 요인 협업 필터링 〉

아이템-잠재 요인 행렬은 영화 별로 여러 장르 요소로 구성된 영화의 장르별 요소 행렬 Q로 가정하고 factor 1 은 영화의 action 요소 값, factor 2는 romance요소 값.

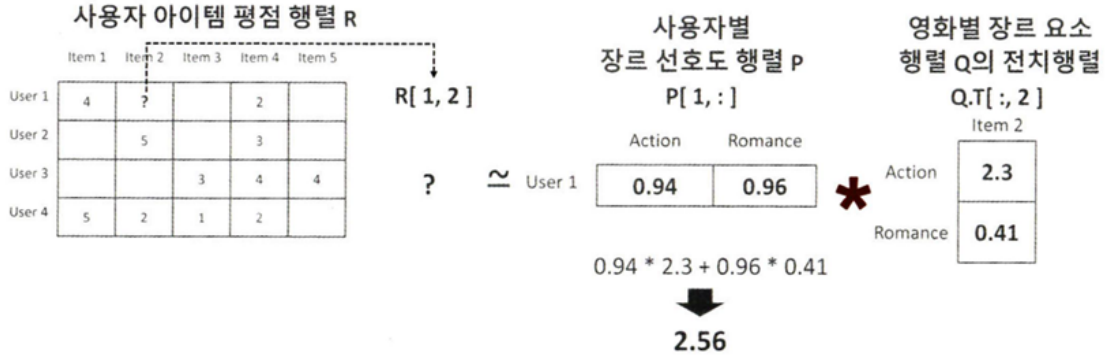
Q는 P와의 내적 계산을 통해 예측 평점을 계산하기 위해 Q의 행과 열 위치를 서로 교환한 Q,T로 변환.

- 평점

  : 사용자의 특정 영화 장르에 대한 선호도와 개별 영화의 그 장르적 특성값을 반영해 결정됨.

  user 1이 평점을 매기지 못한 item 2에 대해 예측 평점 수행 가능

  ⇒ R(1,2)는 행렬 분해된 P매트릭스의 user 1 벡터와 Q.T 매트릭스의 item 2 벡터의 내적 결괏값인 2.56으로 예측 가능

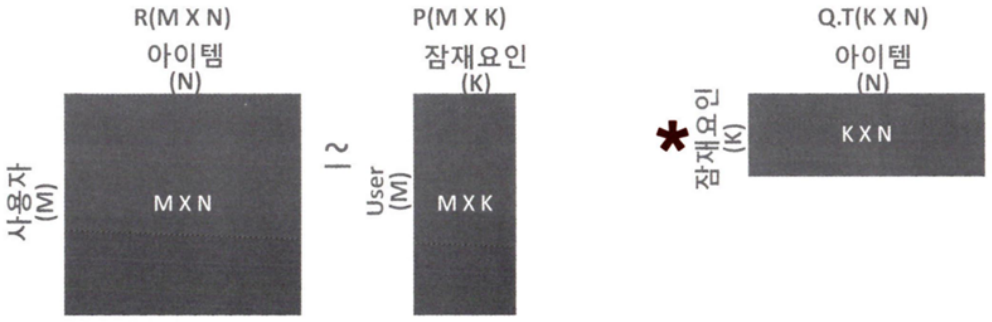


- 행렬 분해의 이해

- 행렬 분해

  : 다차원의 매트릭스를 저차원 매트릭스로 분해하는 기법

  : SVD, NMF 등이 있음



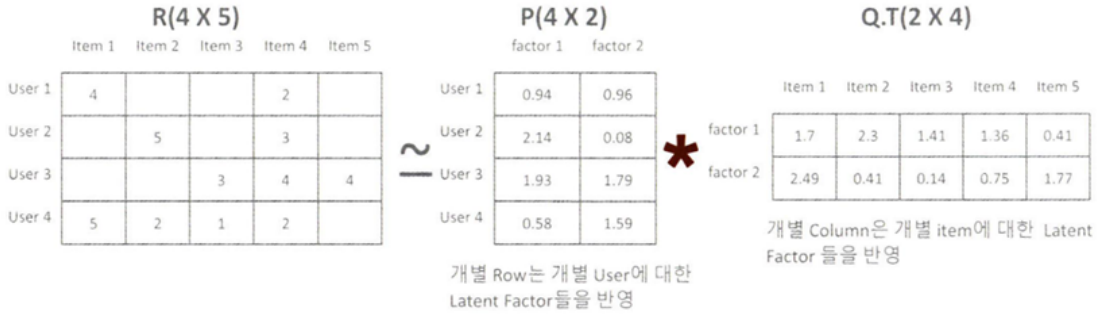
즉, R = P*Q.T이며 각 기호에 대한 설명은 다음과 같습니다.

- M은 총 사용자 수
- N은 총 아이템 수
- K는 잠재 요인의 차원 수
- R은 M X N 차원의 사용자–아이템 평점 행렬

09 _ 추천 시스템 573

- P는 사용자와 잠재 요인과의 관계 값을 가지는 M X K 차원의 사용자–잠재 요인 행렬
- Q는 아이템과 잠재 요인과의 관계 값을 가지는 N X K 차원의 아이템–잠재 요인 행렬
- Q.T는 Q 매트릭스의 행과 열 값을 교환한 전치 행렬

행렬 내에 널 값을 많이 가지는 고차원의 희소 행렬인 R행렬은 아래 그림과 같이 저차원의 밀집 행렬인 P행렬과 Q 행렬로 분해될 수 있음

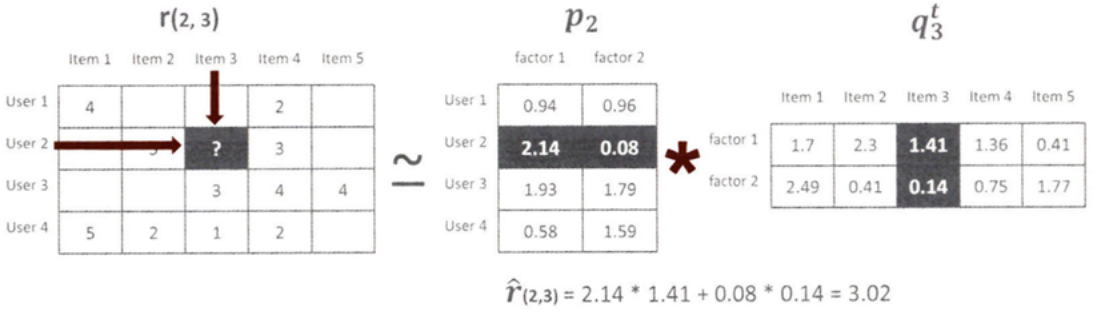


R 행렬의 u행 사용자와 i열 아이템 위치에 있는 평점 데이터를 $r_{(u,i)}$라고 하면

$$r_{(u,i)} = p_u * q_i^t$$

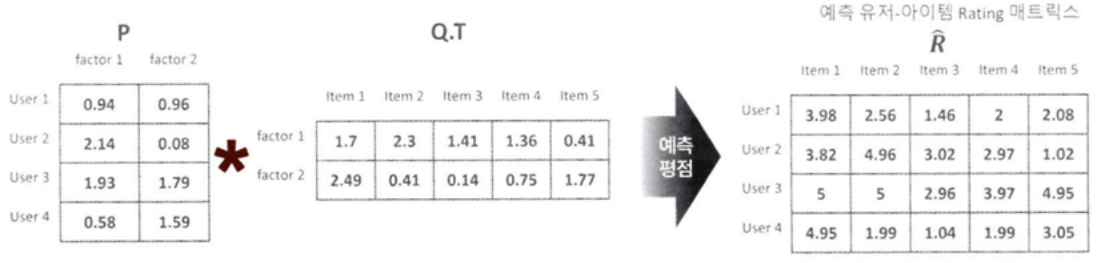
위 평점 데이터 식에서 pu는 u행 사용자의 벡터, q는 Q행렬의 i 행 아이템 벡터의 전치 벡터

사용자가 평가하지 않은 아이템에 대한 평점도 잠재 요인으로 분해된 P행렬과 Q행렬을 이용해 예측 가능



사용자-아이템 평점 행렬의 미정 값을 포함한 모든 평점 값은 행렬 분해를 통해 얻어진 P 행렬과 Q.T 행렬의 내적을 통해 예측 평점으로 다시 계산할 수 있습니다.

$$R \cong \hat{R} = P * Q.T$$



그렇다면 R행렬을 어떻게 P와 Q행렬로 분해하는지?

⇒ 행렬 분해는 주로 SVD 방식을 이용

하지만 SVD는 널값이 없는 행렬에만 적용할 수 있다.

평점이 입력 되지 않은 R행렬에는 많은 널값이 있어

이러한 경우, **확률적 경사 하강법 방식**을 이용해 SVD 수행

- 확률적 경사 하강법을 이용한 확률 분해

: P와 Q 행렬로 계산된 예측 R 행렬 값이 실제 R 행렬 값과 가장 최소의 오류를 가질 수 있도록 반복적인 비용함수 최적화를 통해 P와 Q를 유추해내는 것

1. P와 Q를 임의의 값을 가진 행렬로 설정
2. P와 Q.T를 곱해 예측 R행렬을 계산하고 예측 R행렬과 실제 R행렬에 해당하는 오류 값 계산
3. 이 오류 값을 최소화할 수 있도록 P와 Q 행렬을 적절한 값으로 각각 업데이트
4. 만족할 만한 오류 값을 가질 때까지 2,3 번 작업을 반복하면서 P와 Q값을 업데이트 해 근사화함

실제 값과 예측값의 오류 최소화와 L2 규제(Regularization)를 고려한 비용 함수식은 다음과 같습니다.

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

일반적으로 사용자-아이템 평점 행렬의 경우 행렬 분해를 위해서 단순히 예측 오류값의 최소화와 학습 시 과적합을 피하기 위해서 규제를 반영한 비용 함수를 적용합니다. 그리고 위의 비용 함수를 최소화하기 위해서 새롭게 업데이트되는 $\acute{p}_u$와 $\acute{q}_i$는 다음과 같이 계산할 수 있습니다(식의 유도는 이 책의 범위를 벗어나 생략하겠습니다).

$$\acute{p}_u = p_u + \eta (e_{(u,i)} * q_i - \lambda * p_u)$$
$$\acute{q}_i = q_i + \eta (e_{(u,i)} * p_u - \lambda * q_i)$$

비용 함수식과 업데이트 식의 기호가 의미하는 바는 다음과 같습니다.

- p_u: P 행렬의 사용자 u행 벡터
- q_i^t: Q 행렬의 아이템 i행의 전치 벡터(transpose vector)
- $r_{(u,i)}$: 실제 R 행렬의 u행, i열에 위치한 값.

- $\hat{r}_{(u,i)}$: 예측 $\hat{R}$ 행렬의 u행, i열에 위치한 값. $p_u * q_i^t$로 계산
- $e_{(u,i)}$: u행, i열에 위치한 실제 행렬 값과 예측 행렬 값의 차이 오류. $r_{(u,i)} - \hat{r}_{(u,i)}$로 계산
- η: SGD 학습률
- λ: L2 규제(Regularization) 계수

SGD를 이용해 행렬 분해를 수행하는 예제 → Colab에서 실행

08. 파이썬 추천 시스템 패키지-Surprise

- Surprise 패키지 소개

: 파이썬 기반의 추천 시스템 구축을 위한 전용 패키지 중 하나인 Surprise 는 파이썬 기반에서 사이킷런과 유사한 API 와 프레임워크를 제공하여, 추천 시스템의 전반적인 알고리즘을 이해하고 사이킷런 사용경험이 있으면 쉽게 사용할 수 있다.

: Surprise 패키지는 "pip install scikit-surprise" 혹은 "conda install -c conda-forge scikit-surprise" 를 입력하여 설치할 수 있다.

: API 를 이용해 쉽게 추천 시스템을 구축할 수 있도록 만들어짐

- Surprise를 이용한 추천 시스템 구축

colab 에서 실행

- Surprise 주요 모듈 소개

[Dataset]

- Dataset.load_builtin(name= 'ml-100k')

  : 무비렌즈 아카이브 FTP서버에서 무비렌즈 데이터를 내려받음

- Dataset.load_from_file(file_path, reader)

  : OS 파일에서 데이터를 로딩할 때 사용

  : 콤마, 탭 등으로 칼럼이 분리된 포맷의 OS파일에서 데이터를 로딩

  : 입력 파라미터로 OS 파일명, reader로 파일의 포맷 지정

- Dataset.load_from_df(df, reader)

  : 판다스의 dataframe에서 데이터를 로딩

[OS 파일 데이터를 Surprise 데이터 세트로 로딩]

colab에서 실행

- Surprise 추천 알고리즘 클래스

- SVD: 행렬 분해를 통한 잠재 요인 협업 필터링을 위한 SVD 알고리즘
- KNNBasic: 최근접 이웃 협업 필터링을 위한 KNN 알고리즘
- BaselineOnly: 사용자 Bias와 아이템 Bias를 감안한 SGD 베이스라인 알고리즘

- SVD 클래스의 입력 파라미터
  - n_factors

    : 잠재요인 K의 개수

    : 디폴트는 100

    : 커질수록 정확도가 높아질 수 있으나 과적합 문제가 발생할 수 있다
  - n_epochs

    : SGD 수행 시 반복횟수

    : 디폴트는 20
  - biased(bool)

    : 베이스라인 사용자 편향 적용 여부

    : 디폴트는 True

- 베이스라인 평점

: 한 개인의 성향을 반영해 아이템 평가의 편향성 요소를 반영하여 평점을 부과하는 것

: 보통 '전체 평균 평점+사용자 편향 점수+아이템 편향 점수' 공식으로 계산됨

- 전체 평균 평점 = 모든 사용자의 아이템에 대한 평점을 평균한 값
- 사용자 편향 점수 = 사용자별 아이템 평점 평균 값 – 전체 평균 평점
- 아이템 편향 점수 = 아이템별 평점 평균 값 – 전체 평균 평점

- 교차 검증과 하이퍼 파라미터 튜닝

colab에서 실행