Chapter 09. 추천 시스템

01. 추천 시스템의 개요와 배경

추천 시스템을 통해 사용자의 취향을 이해하고 맞춤 상품과 콘텐츠를 제공해 조금이라도 오래 자기 사이트에 고객을 머무르게 함

- 고도화된 추천 엔진을 가지고 있는 사이트에 접속하면 빠져나오기 힘듦
- 사용자 자신도 몰랐던 취향을 시스템이 발견하고 그에 맞는 콘텐츠 추천
- 추천 시스템 구성 데이터: 어떤 상품 구매? 어떤 상품을 장바구니에? 사용자의 평가는? 사용자의 취향은?

추천 시스템의 유형: 콘텐츠 기반 필터링 방식, 협업 필터링 방식

- 콘텐츠 기반 필터링 방식: 초창기에 주로 사용
- 협업 필터링 방식
    - 최근접 이웃 협업 필터링: 초창기에 주로 사용
    - 잠재 요인 협업 필터링: 행렬 분해 기법 이용, 대부분의 온라인 스토어에서 적용

02. 콘텐츠 기반 필터링 추천 시스템

콘텐츠 기반 필터링 방식: 사용자가 특정한 아이템을 매우 선호하는 겨우, 그 아이템과 비슷한 콘텐츠를 가진 다른 아이템을 추천하는 방식

- ex) 사용자가 특정 영화에 높은 평점을 줬다면 그 영화의 장르, 출연 배우, 감독 등의 콘텐츠와 유사한 다른 영화를 추천해주는 방식

03. 최근접 이웃 협업 필터링

협업 필터링 방식: 사용자가 아이템에 매긴 평점 정보나 상품 구매 이력과 같은 사용자 행동 양식만을 기반으로 추천 수행

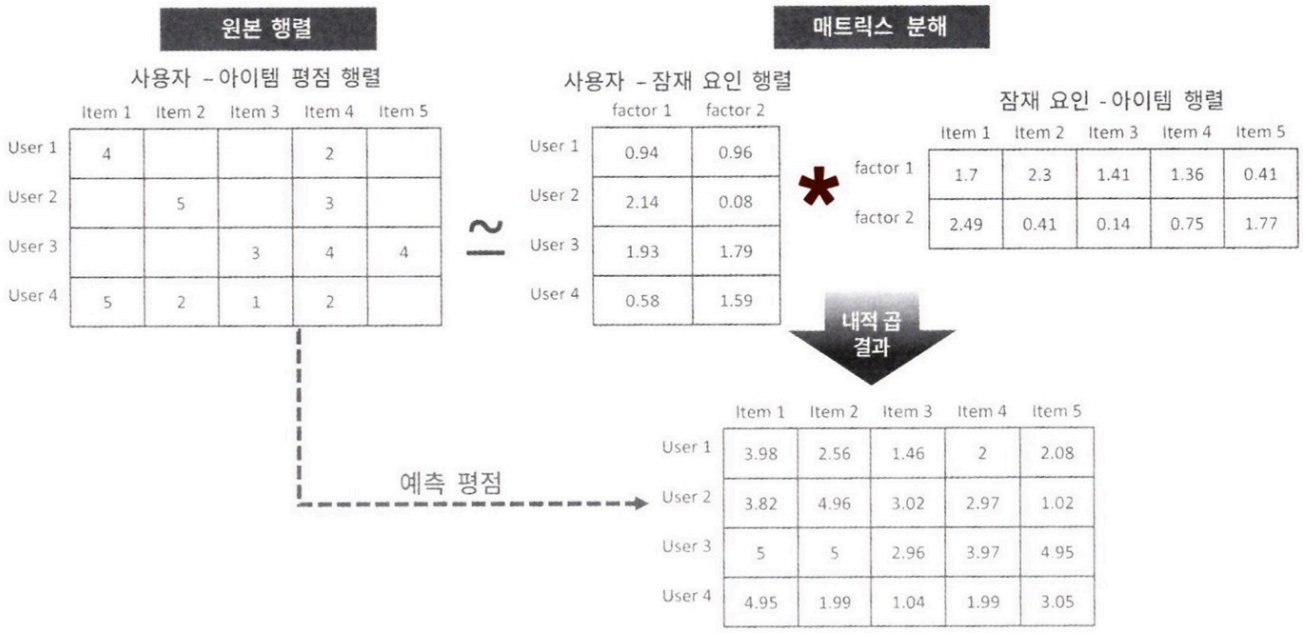
- 목표: 사용자-아이템 평점 매트릭스와 같은 축적된 사용자 행동 데이터를 기반으로 사용자가 아직 평가하지 않은 아이템을 예측 평가
- 최근접 이웃 방식과 잠재 요인 방식으로 나뉨, 둘 다 **사용자-아이템 평점 행렬** 데이터에만 의지

    - 행은 개별 사용자, 열은 개별 아이템으로 구성, 사용자 아이디 행 , 아이템 아이디 열 위치에 해당하는 값이 평점
- **최근접 이웃 방식**: 사용자 기반, 아이템 기반으로 나뉨
    - 사용자 기반: 당신과 비슷한 고객들이 다음 상품도 구매했습니다.
        - 특정 사용자와 유사한 다른 사용자를 TOP-N으로 선정해, 사용자가 좋아하는 아이템을 추천
    - 아이템 기반: 이 상품을 선택한 다른 고객들은 다음 상품도 구매했습니다.
        - 사용자들이 그 아이템을 좋아하는지/싫어하는지 평가 척도가 유사한 아이템을 추천하는 기준이 됨
    - 일반적으로 사용자 기반보다는 아이템 기반 협업 필터링이 정확도가 더 높음

04. 잠재 요인 협업 필터링

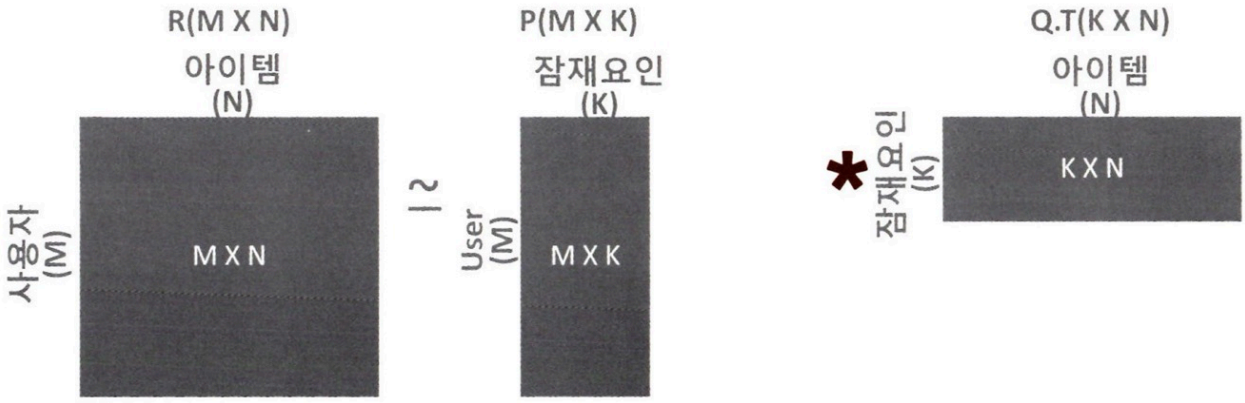
잠재 요인 협업 필터링: 사용자-아이템 평점 매트릭스 속에 숨어 있는 잠재 요인을 추출해 추천 예측을 할 수 있게 하는 기법

- 대규모 다차원 행렬을 SVD와 같은 차원 감소 기법으로 분해하는 과정에서 잠재 요인을 추출 (행렬 분해)
- '잠재 요인'을 기반으로 다차원 희소 행렬인 사용자-아이템 행렬 데이터를 저차원 밀집 행렬의 사용자-잠재요인 행렬과 아이템-잠재요인 행렬의 전치 행렬로 분해, 분해된 두 행렬의 내적을 통해 새로운 예측 사용자-아이템 평점 행렬 데이터를 만듦



〈 행렬 분해를 통한 잠재 요인 협업 필터링 〉

행렬 분해: 다차원의 매트릭스를 저차원 매트릭스로 분해하는 기법, SVD, NMF 등



- R = P * Q.T
  - M: 총 사용자 수, N: 총 아이템 수, K: 잠재 요인의 차원 수, R: 사용자-아이템 평점 행렬, P: 사용자-잠재요인 행렬, Q: 아이템-잠재요인 행렬, Q.T: Q의 행과 열 값을 교환한 전치 행렬
  - 사용자-아이템 평점 행렬의 미정 값을 포함한 모든 평점 값은 행렬 분해를 통해 얻어진 P 행렬과 Q.T 행렬의 내적을 통해 예측 평점으로 다시 계산 가능
  - SVD는 널 값이 없는 행렬에만 적용 가능, 보통 확률적 경사 하강법이나 ALS 방식 이용해서 SVD 수행

- 확률적 경사 하강법: P와 Q 행렬로 계산된 예측 R 행렬 값이 실제 R 행렬 값과 가장 최소의 오류를 가질 수 있도록 반복적인 비용 함수 최적화를 통해 P와 Q를 유추해내는 것
  - 실제 값과 예측값의 오류 최소화와 L2 규제를 고려한 비용 함수식

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

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

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

- SGD 기반 행렬 분해: L2 규제를 반영해 실제 R 행렬 값과 예측 행렬 값의 차이를 최소화하는 방향성을 가지고 P 행렬과 Q 행렬에 업데이트 값을 반복적으로 수행하면서 최적화된 예측 R 행렬을 구하는 방식

08. 파이썬 추천 시스템 패키지 - Surprise

Surprise: 파이썬 기반의 추천 시스템 구축을 위한 전용 패키지 중 하나

- conda나 pip를 통해 설치
  - $pip install scikit-surprise
  - $ conda install -c conda-forge scikit-surprise
- 장점: 다양한 추천 알고리즘, 핵심 API는 사이킷런의 핵심 API와 유사
- 추천을 예측하는 메서드: test(), predict()

- Surprise 주요 모듈
  - Dataset.load_builtin(name='ml-100k')
  - Dataset.load_from_file(file_path, reader)
  - Dataset.load_from_df(df, reader)
- Reader 클래스의 주요 생성 파라미터: line_format(string), sep(char), rating_scale(tuple, optional)
- 추천 예측을 위해 자주 사용되는 추천 알고리즘 클래스: SVD, KNNBasic, Baselineonly
  - SVD 클래스의 입력 파라미터: n_factors, n_epochs, biased(bool)
- 베이스라인 평점: 한 개인의 성향을 반영해 아이템 평가에 편향성 요소를 반영하여 평점을 부과하는 것
  - 보통 전체 평균 평점 + 사용자 편향 점수 + 아이템 편향 점수로 계