9️⃣

09 _ 추천 시스템

추천시스템의 개요와 배경

<추천 시스템의 유형>

- 콘텐츠 기반 필터링 (Content based filtering)
- 협업 필터링 (Collaborative Filtering)
  - 최근접 이웃 협업 필터링 (Nearst Neighbor)
  - 잠재요인 협업 필터링 (Latent Factor)

콘텐츠 기반 필터링 추천 시스템

- 사용자가 특정한 아이템을 매우 선호하는 경우, 그 아이템과 비슷한 콘텐츠를 가진 다른 아이템을 추천하는 방식
  - ex) 특정 영화에 높은 평점을 주었다면 그 영화의 장르, 출연 배우, 감독, 영화 키워드 등의 콘텐츠와 유사한 다른 영화를 추천해 줌

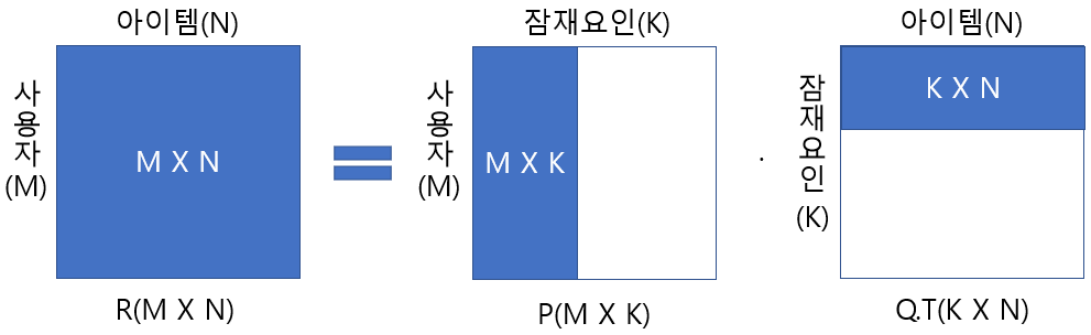
최근접 이웃 협업 필터링

- 사용자가 남긴 평점, 평가, 구매이력 등의 행동 양식(User Behavior)을 기반으로 추천하는 방식
- 즉, 사용자가 평가하지 않은 아이템을 평가한 아이템에 기반하여 예측 및 평가
- 사용자-아이템 행렬에서 행(Row)은 개별 사용자, 열(Column)은 개별 아이템
- 일반적으로 사용자-아이템 행렬은 다차원 행렬이며, 희소 행렬(Sparse)의 특징이 있음

- 최근접 이웃 필터링 분류
  - 사용자 기반(User-User): 당신과 비슷한 고객들이 다음 상품도 구매했습니다.
  - 아이템 기반(Item-Item): 이 상품을 구매한 다른 고객들은 다음 상품도 구매했습니다.
- 사용자 기반(User-User)
  - 특정 사용자와 타 사용자 간의 유사도(Similarity)를 측정하여 유사도가 높은 타 사용자들이 선호하는 아이템을 추천
- 아이템 기반(Item-Item)
  - 사용자들의 아이템 선호도를 바탕으로 추천
- 일반적으로는 아이템 기반 필터링이 정확도가 좋으며, 유사도 측정으로는 코사인 유사도를 적용

잠재요인 협업 필터링

- 사용자-아이템 평점 매트릭스 안에 있는 잠재 요인을 추출해 추천 예측하는 방법
- 대규모 다차원 행렬을 SVD 등의 차원 감소 기법으로 분해하면서 잠재 요인을 추출 → 행렬 분해(Matrix Factorization)
- 머신러닝의 블랙박스(Black Box)처럼, 잠재 요인 추출 역시 정확하게 무엇을 추출하는지는 아직 알 수 없음...
- 사용자-아이템 평점 행렬 → ①사용자-잠재요인 행렬, ②잠재요인-아이템 행렬, ③ ①·② 내적 및 예측
- 행렬 분해의 이해
  - 다차원 매트릭스를 저차원으로 분해
  - 대표적으로 SVD(Singular Vector Decomposition), NMF(Non-Negative Matrix Factorization)



P는 사용자-잠재요인(M X K) 행렬

Q는 아이템-잠재요인(N X K) 행렬

Q.T는 Q의 행과 열을 교환한 전치 행렬

R행렬은 평점이 없는 NULL값이 많으므로 SVD방법으로 P, Q행렬 분해 불가

경사 하강법(Stochastic Gradient Descent) 혹은 ALS(Alternating Least Squares) 방식 활용

<확률적 경사하강법(SGD)를 이용한 행렬 분해>

R 행렬을 P와 Q 행렬로 분해하기 위해서는 주로 SVD 방식을 이용하나, 이는 널(NaN) 값이 없는 행렬에만 적용할 수 있다.

R 행렬은 대부분의 경우 널(NaN) 값이 많이 존재하는 희소행렬이기 때문에 일반적인 SVD 방식으로 분해할 수 없고, 확률적 경사 하강법(Stochastic Gradient Descent, SGD) 이나 ALS(Alternating Least Squares) 방식을 이용해 SVD 를 수행한다.

특히, 확률적 경사 하강법을 이용한 행렬 분해를 살펴보자면 P와 Q 행렬로 계산된 예측 R 행렬 값이 실제 R 행렬 값과 최소한의 오류를 가질 수 있도록 반복적으로 비용함수를 최적화함으로써 적합한 P와 Q 행렬을 유추하는 것이 알고리즘의 골자이다.

확률적 경사 하강법을 이용한 행렬 분해의 전반적인 절차

1. P와 Q 행렬을 임의의 값을 가진 행렬로 초기화 한다.
2. P와 Q 전치행렬을 곱해 예측 R 행렬을 계산하고, 실제 R 행렬과의 차이를 계산한다.
3. 차이를 최소화할 수 있도록 P와 Q 행렬의 값을 적절한 값으로 각각 업데이트한다.
4. 특정임계치 아래로 수렴할 때까지 2, 3번 작업을 반복하면서 P와 Q 행렬을 업데이트 해 근사화한다.

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

실제 값과 예측값의 오류 최소화와 L2 규제를 고려한 비용함수 식

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

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

위의 비용함수 식을 최소화 하기 위해 업데이트 되는 p, q

- pu: P 행렬의 사용자 u행 벡터, qti: Q 행렬의 아이템 i행의 전치 벡터, r(u, i): 실제 R 행렬의 u행 i열에 위치한 값
- r'(u, i): 예측 R' 행렬의 u행 i열에 위치한 값, e(u, i): u행 i열에 위치한 실제 행렬 값과 예측 행렬 값의 차이 오류
- δ: SGD 학습률
- γ: L2 규제(Regularization) 계수