Recommender System

Content-based Recommendation

- 각 사용자가 구매 / 만족했던 상품과 유사한 것을 추천하는 방법
  - 동일한 장르의 영화를 추천
  - 동일한 감독의 영화 혹은 동일 배우가 출현한 영화를 추천
  - 동일한 카테고리의 상품을 추천하는 것
  - 동갑의 같은 학교를 졸업한 사람을 친구로 추천하는 것

과정

1. 사용자가 선호했던 상품들의 상품 프로필 (Item Profile)을 수집하는 단계
   - 해당 상품의 특성을 나열한 벡터
   - 무수하게 많은 Feature들에 대해 One-hot Encoding 으로 표현
2. 사용자 프로필 (User Profile)을 구성하는 단계
   - 사용자 프로필은 선호한 상품의 상품 프로필을 선호도를 사용하여 가중 평균하여 계산
   - 어떠한 사용자가 영화를 본다고 했을 때, 각 영화들은 특성 벡터 가지나 그 각 영화들의 가중치는 서로 다르기 때문에 그 가중치를 반영하여 Aggregation을 진행
3. 사용자 프로필과 다른 상품들의 상품 프로필을 매칭하는 단계
   - 사용자 프로필 벡터와 상품 프로필 벡터의 Similarity를 계산

장점

- 다른 사용자의 구매 기록이 필요하지 않음
- 독특한 취향의 사용자에게도 추천이 가능
- 새 상품에 대해서도 추천이 가능
- 추천의 이유를 주장할 수 있음 (History 바탕)

단점

- 상품에 대한 부가 정보가 없는 경우에는 사용할 수 없음

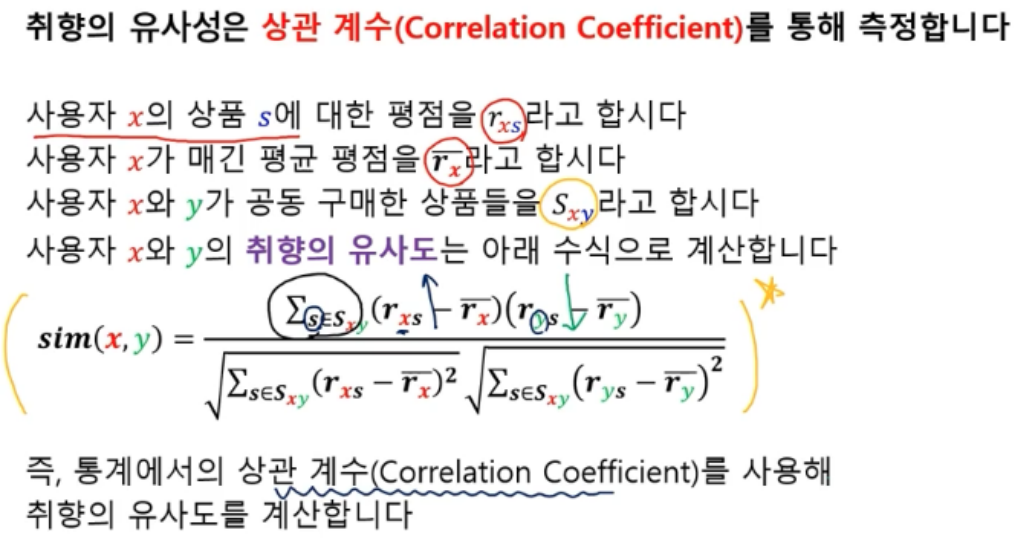
- 구매 기록이 없는 사용자에게는 사용할 수 없음
- Overfitting으로 지나치게 협소한 추천을 할 위험이 존재

ch9.pdf

Collaborative Filtering

과정

1. 사용자와 유사한 취향을 갖는 기존 사용자들을 찾음.
    - 어떻게 유사한 취향의 사용자를 찾을 수 있을까 ?
        - 상관계수 (Correlation Coefficient) 를 통해 측정



2. 유사한 취향의 사용자들이 선호한 상품을 찾음
    - 이러한 상관계수를 이용하여 특정 상품 S를 구매한 사용자 중에 대상 사용자 X와 취향이 가장 유사한 K명의 사용자를 도출
    - 취향의 유사도를 가중치로 사용한 평점의 가중 평균을 통해 평점을 추정
3. 이 상품들을 사용자에게 추천
    - 대상 사용자 X가 구매하지 않은 모든 제품에 대해 평점을 추정한 뒤 평점이 가장 높은 상품을 추천

장점

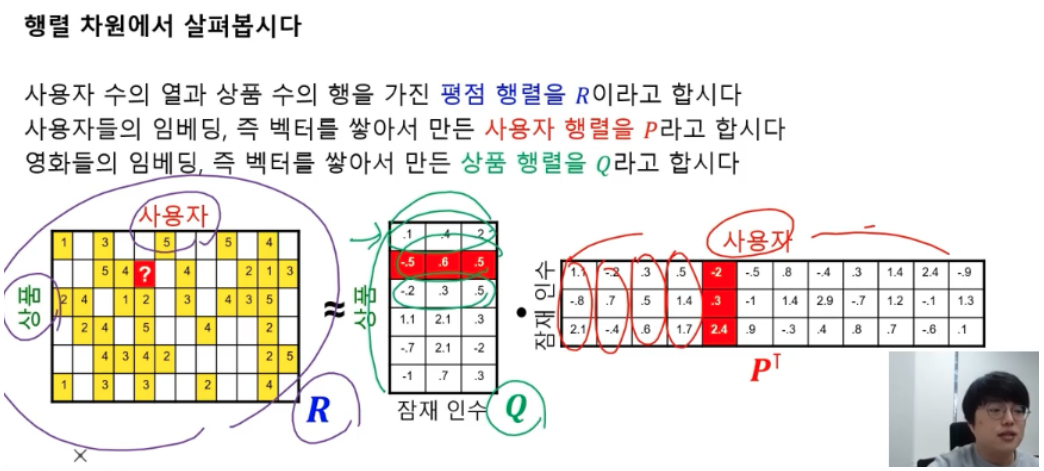
- 상품에 대한 부가 정보가 없는 경우에도 사용할 수 있다.

단점

- 충분한 수의 평점 데이터가 누적되어야 효과적
- 새 상품, 새로운 사용자에 대한 추천이 불가능
- 독특한 취향의 사용자에게 추천이 어려움

Latent Factor Model

- Latent Factor Model의 핵심은 사용자와 상품을 벡터로 표현하는 것
- 어떻게 Embedding 할 것인가 ?
    - 각 사용자와 상품의 임베딩 벡터 내적이 평점과 최대한 유사하도록 학습
        - 사용자 x의 임베딩을 p_x, 상품 i의 임베딩을 q_i
        - 사용자 x의 상품 i에 대한 평점을 r_xi
        - 임베딩의 목표는 p_x^t * q_i 가 r_xj 와 유사하도록 학습

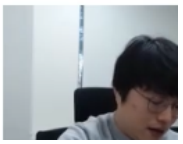


잠재 인수 모형은 다음 손실 함수를 최소화하는 P와 Q를 찾는 것을 목표로 합니다

$$\sum_{(i,x)\in R} (r_{xi} - p_x^\intercal q_i)^2$$

훈련 데이터에 있는
평점에 대해서만 계산합니다

하지만, 위 손실 함수를 사용할 경우 과적합(Overfitting)이 발생할 수 있습니다
과적합이란 기계학습 모형이 훈련 데이터의 잡음(Noise)까지 학습하여,
평가 성능은 오히려 감소하는 현상을 의미합니다



×

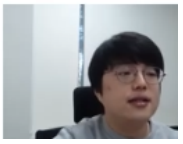
과적합을 방지하기 위하여 정규화 항을 손실 함수에 더해줍니다

정규화의 세기
(하이퍼파라미터)

$$\sum_{(i,x)\in R} (r_{xi} - p_x^\intercal q_i)^2 + [\lambda_1 \sum_x ||p_x||^2 + \lambda_2 \sum_i ||q_i||^2]$$

오차 (훈련 데이터에 있는 평점에 대해서만 계산합니다)

모형 복잡도



×

Latent Factor Model with bias from user and item

- 사용자의 편향: 해당 사용자의 평점 평균과 전체 평점 평균의 차이
- 아이템의 편향: 해당 상품에 대한 평점 평균과 전체 평점 평균의 차이

개선된 잠재 인수 모형에서는 평점을 전체 평균, 사용자 편향, 상품 편향, 상호작용으로 분리합니다

$$r_{xi} = \mu + b_x + b_i + p_x^T q_i$$

평점 전체 평균 사용자 편향 상품 편향 상호작용

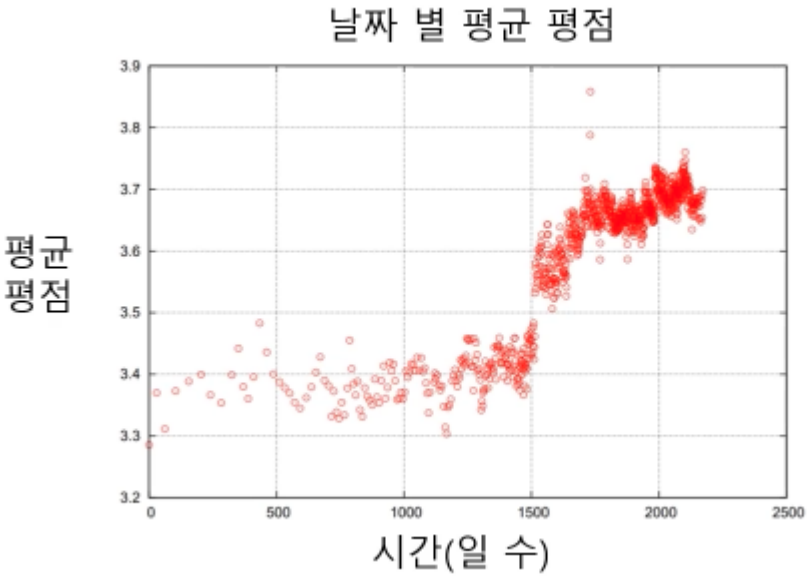
- 사용자 편향과 상품 편향을 함께 학습하여 평점을 예측 (전체 평균은 단순히 사용)

개선된 잠재 인수 모형의 손실 함수는 아래와 같습니다

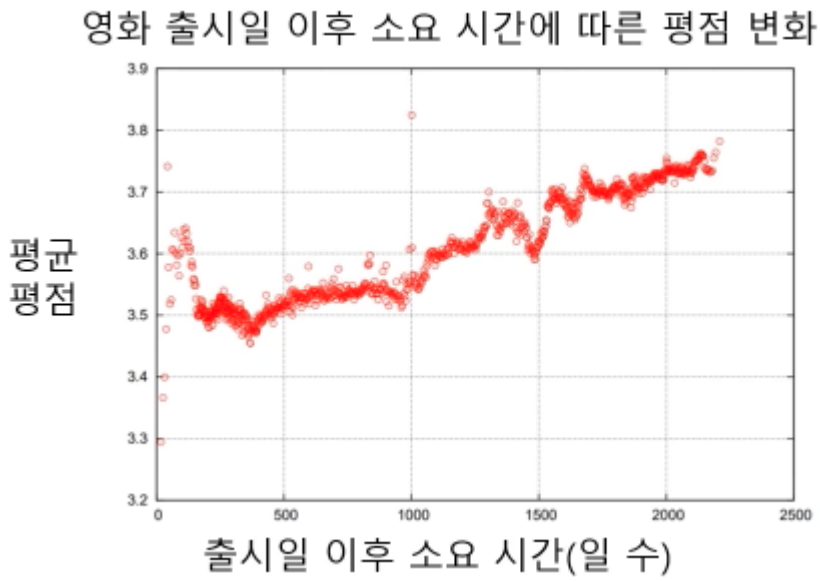
$$\sum_{(i,x)\in R} (r_{xi} - (\mu + b_x + b_i + p_x^T q_i))^2 + [\lambda_1 \sum_x ||p_x||^2 + \lambda_2 \sum_i ||q_i||^2 + \lambda_3 \sum_x b_x^2 + \lambda_4 \sum_i b_i^2]$$

Latent Factor Model with Time-series

- 영화의 평점은 시간의 변화에 따라 영향을 받는 경우가 존재
  - 넷플릭스 UI/UX 시스템의 변화로 평균 평점이 크게 상승하는 사건 존재



  - 영화의 평점은 출시일 이후 시간이 지남에 따라 상승하는 경향을 가짐



- 사용자 편향과 상품 편향이 시간에 따라 변할 수 있게 반영

개선된 잠재 인수 모형에서는 이러한 시간적 편향을 고려합니다

구체적으로 사용자 편향과 상품 편향을 시간에 따른 함수로 가정합니다

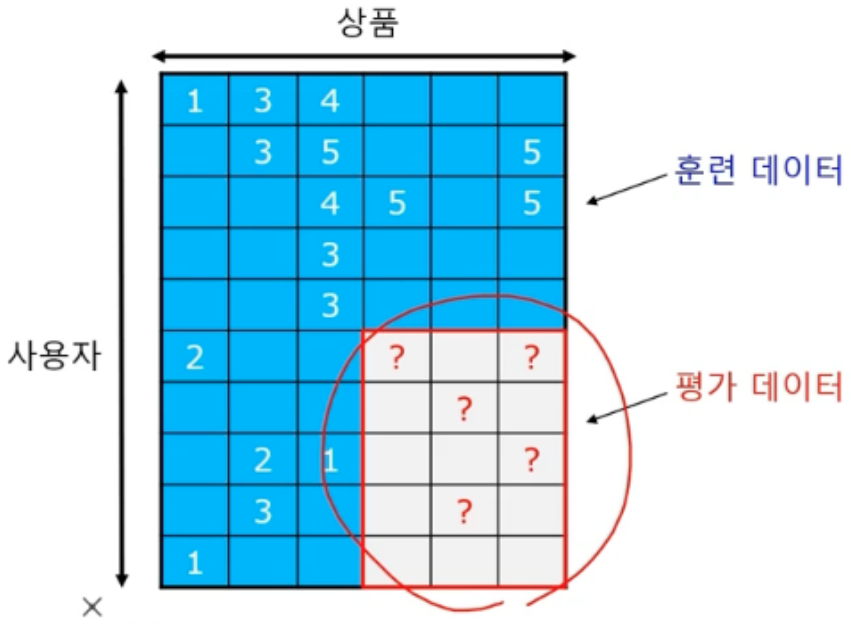
$$r_{xi} = \mu + b_x(t) + b_i(t) + p_x^T q_i$$

평점 전체 평균 사용자 편향 상품 편향 상호작용

추천 시스템 평가

- User-Item Rating Matrix를 통해 조사

평가 데이터는 주어지지 않았다고 가정합시다



- Matrix의 일부를 훈련 - 테스트용 데이터로 분리
- 추천 시스템의 평점과 실제 평점과 비교
  - MSE
  - RMSE