추천 시스템

추천 시스템(recommender system)

- 사용자가 선호하는 상품을 예측하는 시스템

- 상품에 대한 평가(평점)을 기반으로 사용자에게 어떤 상품을 추천할지 예측
  - » Amazon, Netflix 등의 컨텐츠 제공 비즈니스 영역에서 광범위하게 사용

- 사용자 자신도 인지하지 못했던 취향을 시스템이 발견하고 그에 맞는 컨텐츠 추천 → 추천 시스템 신뢰도 향상 → 사용량 증가에 따른 데이터 증가 → 정교한 추천 시스템 구축의 선순환 구조 형성

추천 시스템 알고리즘

- 두 개의 범주 값 입력으로 하나의 연속형 값을 예측하는 회귀 모형

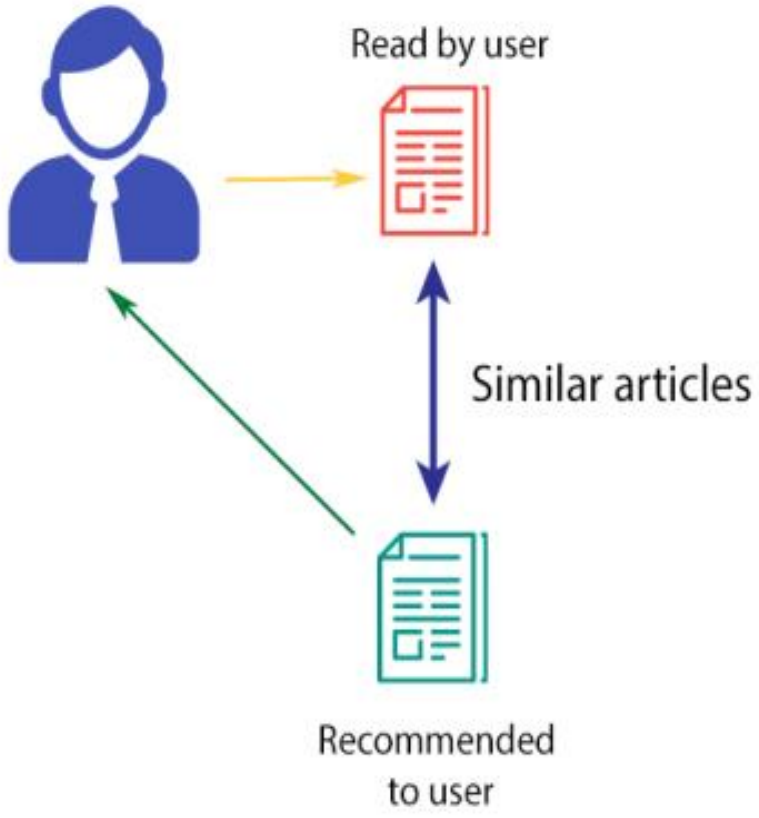
- 종류
  - » 베이스라인 모형
  - » 컨텐츠 기반 필터링 (Content-based Filtering)
  - » 협력 필터링 (Collaborative Filtering)
    - Neighborhood Model
      - . 사용자 기반 협력 필터링 (User-based CF)
      - . 아이템 기반 협력 필터링 (Item-based CF)
    - Latent Factor Model
      - . Matrix Factorization
      - . SVD
    - 하이브리드 모델
      - . Neighborhood Model + Latent Factor Model

컨텐츠 기반 필터링

- 사용자가 특정한 아이템을 매우 선호하는 경우 그 아이템과 비슷한 컨텐츠를 가진 다른 아이템을 추천하는 방식

사용자가 특정 영화에 높은 평점

장르, 출연 배우, 감독, 영화 키워드 등 유사한 컨텐츠를 가진 다른 영화 추천



최근접 이웃 협업 필터링

- 사용자가 아이템에 부여한 평점 정보, 상품 구매 이력 등 사용자 행동 양식을 기반으로 추천 수행
  - » 평점 행렬이 가진 특정한 패턴을 찾아서 평점 예측에 사용하는 방법

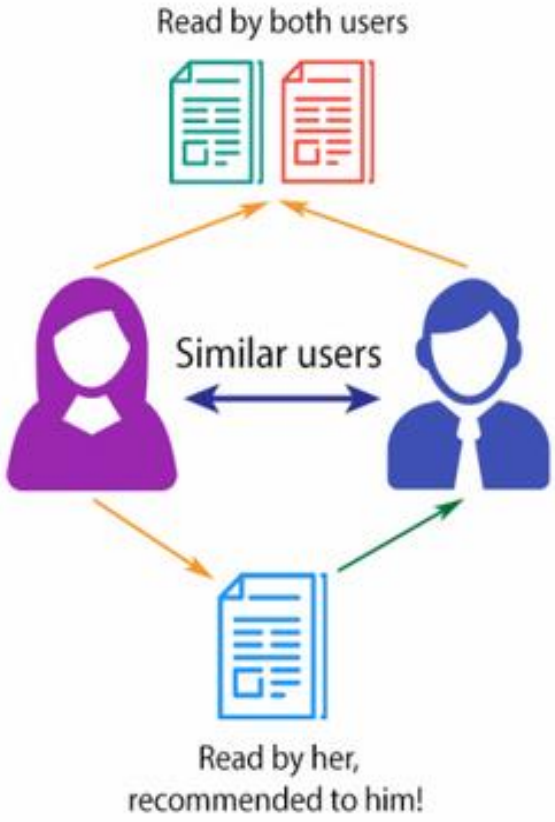
- 사용자 / 아이템 평점 매트릭스와 같은 축적된 사용자 행동 데이터를 기반으로 사용자가 아직 평가하지 않은 아이템을 예측

	Item 1	Item 2	Item 3	Item4
User 1	3		3	
User 2	4	2		3
User 3		1	2	3

최근접 이웃 협업 필터링

종류	설명
사용자 기반	유사한 사용자를 찾기 위해 평점 행렬에서 유사한 사용자 행 벡터를 찾아서 이를 기반으로 비어 있는 데이터(평점) 계산
아이템 기반	평점 행렬에서 상품 열 벡터의 유사성을 찾아서 특정 상품과 유사한 평점 정보를 가지는 상품으로 데이터(평점) 예측

- 일반적으로 아이템 기반 협업 필터링의 정확도가 높으며 사용 비율도 높음

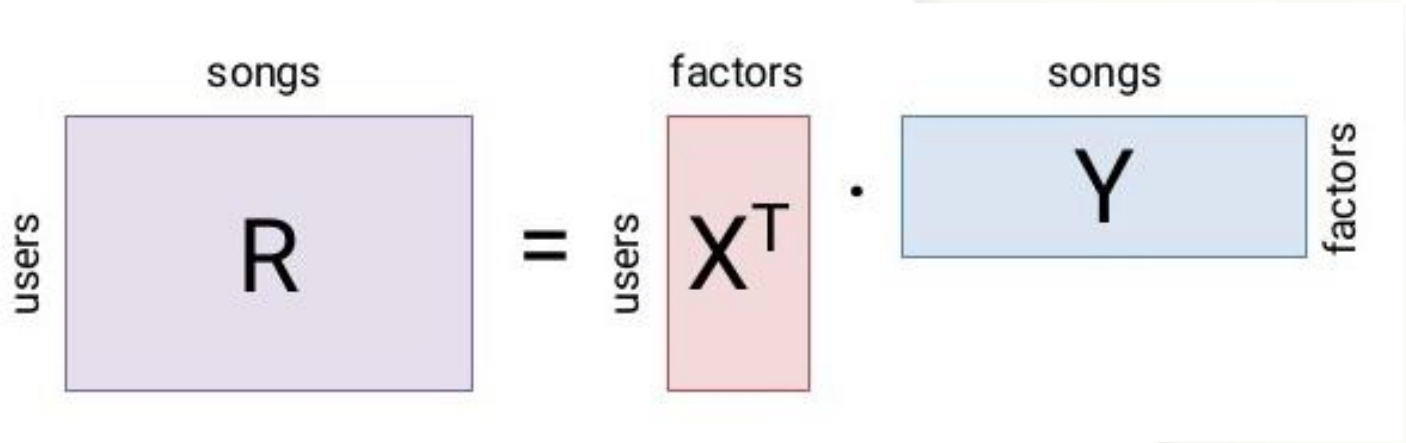


최근접 이웃 협업 필터링

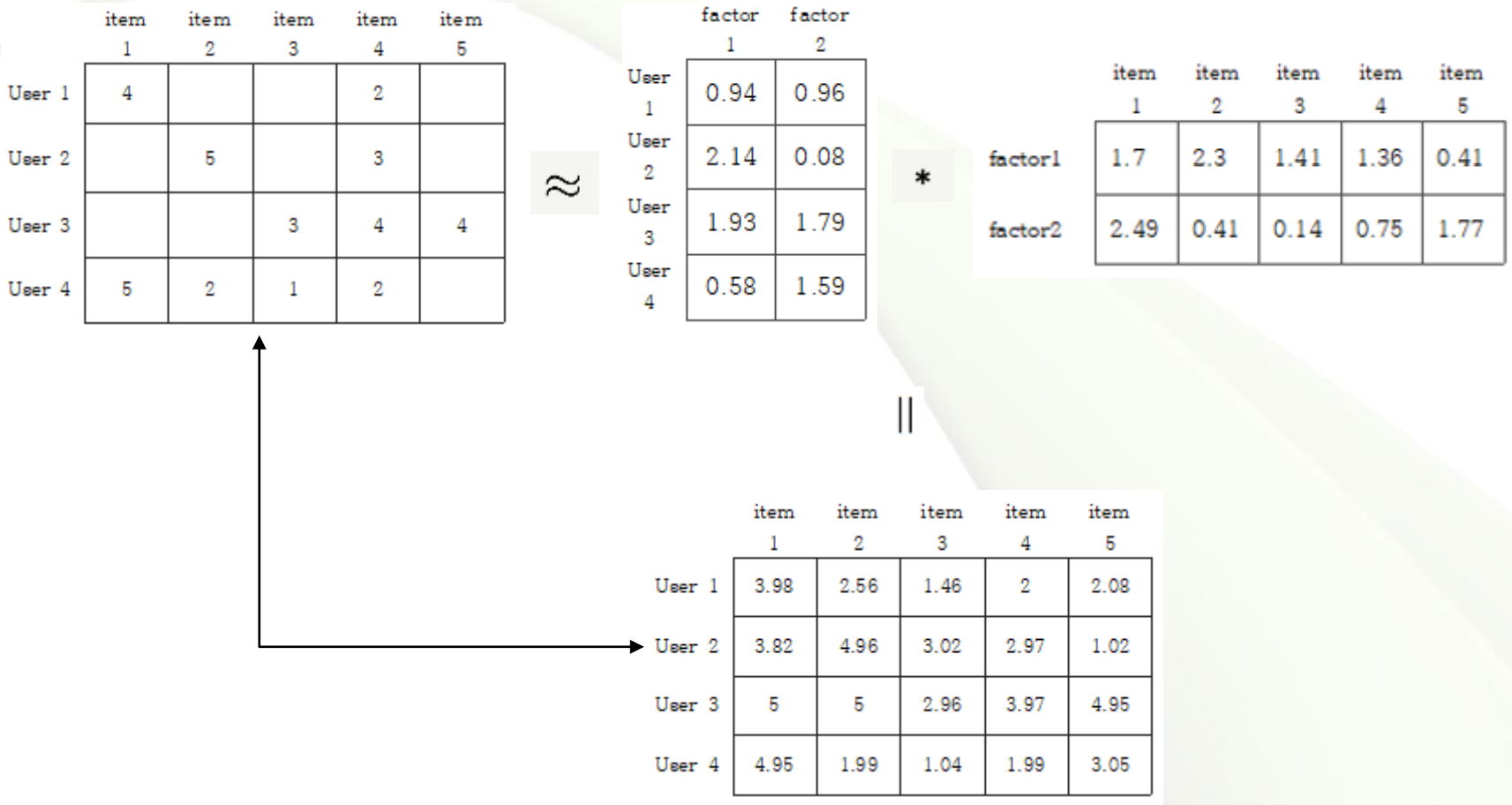
- 유사도 계산
  - » 평균제곱차이 유사도 (Mean Squared Difference Similarity)
  - » 코사인 유사도 (Cosine Similarity)
  - » 피어슨 유사도 (Pearson Similarity)
  - » 피어슨-베이스라인 유사도 (Pearson-Baseline Similarity)

잠재 요인 협업 필터링

- 사용자 특성 벡터 또는 상품의 특성 벡터는 수십 억에 달하는 크기가 될 수 있음
- 사용자 데이터 또는 상품 데이터의 특성이 매우 많은 경우 몇 개의 요인 벡터로 간략화 할 수 있다는 가정에서 출발
- PCA를 사용해서 특성 벡터의 차원을 줄일 수 있듯이 사용자 및 상품의 특성도 차원 축소 가능



잠재 요인 협업 필터링


item 1
item 2
item 3
item 4
item 5
User 1
4
2
User 2
5
3
User 3
3
4
4
User 4
5
2
1
2
≈
factor 1
factor 2
User 1
0.94
0.96
User 2
2.14
0.08
User 3
1.93
1.79
User 4
0.58
1.59
*
item 1
item 2
item 3
item 4
item 5
factor1
1.7
2.3
1.41
1.36
0.41
factor2
2.49
0.41
0.14
0.75
1.77
=
item 1
item 2
item 3
item 4
item 5
User 1
3.98
2.56
1.46
2
2.08
User 2
3.82
4.96
3.02
2.97
1.02
User 3
5
5
2.96
3.97
4.95
User 4
4.95
1.99
1.04
1.99
3.05

Surprise Package

- Python SciKits(SciPy Toolkits) 기반 추천 시스템 패키지

- 설치

```
conda install –c conda-forge scikit-surprise
```

or

```
pip install scikit-surprise
```

- 특징
  - » 다양한 추천 알고리즘을 쉽게 적용해서 추천 시스템 구축 가능
  - » scikit-learn과 유사한 API 제공

모델 사용 방법

- split, folds 등의 함수를 사용해서 K-Fold 훈련 세트와 테스트 세트 만들기
- 모형 알고리즘 객체 만들기
- 모형 알고리즘 객체의 train 함수와 훈련 데이터 세트를 사용해서 모수 추정
- 모형 알고리즘 객체의 test 메서드로 테스트 데이터 세트에 대한 예측 수행
- accuracy 서브 패키지의 성능 평가 함수를 사용해서 예측 성능 계산

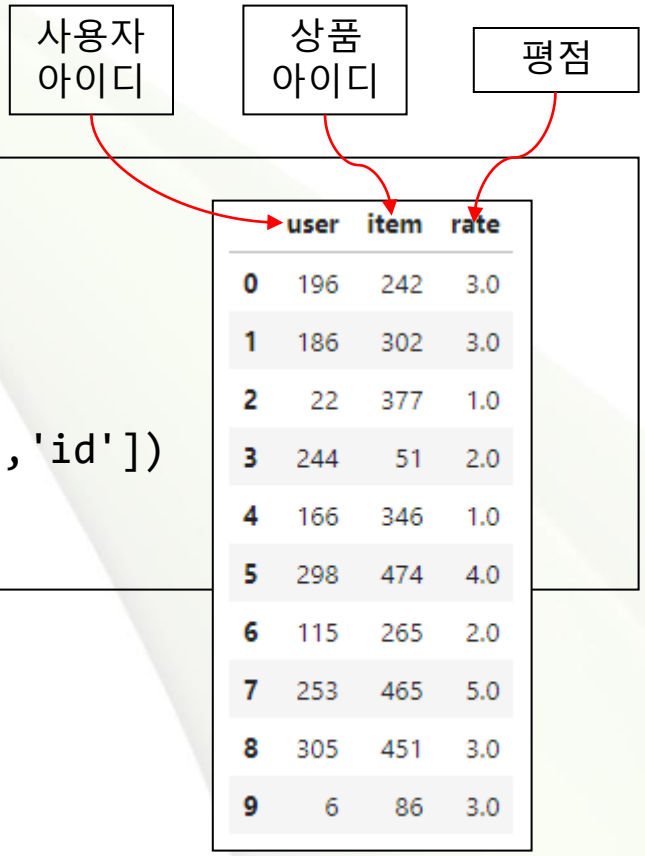
예제 평점 데이터

- 영화 추천 웹사이트 (MovieLense) 평점 데이터
  - » 다운로드 → https://grouplens.org/datasets/movielens/
  - » 100k, 1m, 10m, 20m 등 다양한 규모의 데이터 제공
  - » surprise 패키지에 다운로드 기능 내장

- 구현

```
import surprise
import pandas as pd

data = surprise.Dataset.load_builtin('ml-100k’)
df = pd.DataFrame(data.raw_ratings,
                  columns=['user','item','rate','id'])
del df['id']
df.head(10)
```



	user	item	rate
0	196	242	3.0
1	186	302	3.0
2	22	377	1.0
3	244	51	2.0
4	166	346	1.0
5	298	474	4.0
6	115	265	2.0
7	253	465	5.0
8	305	451	3.0
9	6	86	3.0

추천 시스템 구축을 위한 데이터 구조

- 사용자 아이디 및 상품 아이디 범주 입력과 평점 출력으로 이루어진 데이터 사용

- 데이터를 상품 축과 사용자 축의 평점 행렬로 변경 → sparse matrix

```
df_table = df.set_index(['user', 'item']).unstack()

df_table.iloc[212:222, 808:817]
```

	rate								
item	211	212	213	214	215	216	217	218	219
user									
290	3.0	NaN	NaN	NaN	NaN	4.0	NaN	2.0	NaN
291	NaN	4.0	NaN	4.0	4.0	NaN	NaN	4.0	4.0
292	NaN	NaN	NaN	3.0	NaN	NaN	NaN	NaN	NaN
293	4.0	NaN	3.0	NaN	4.0	4.0	3.0	2.0	NaN
294	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
295	NaN	NaN	5.0	NaN	5.0	5.0	4.0	5.0	NaN
296	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
297	4.0	NaN	3.0	NaN	2.0	4.0	NaN	3.0	NaN
298	5.0	NaN	3.0	NaN	5.0	NaN	NaN	NaN	NaN
299	4.0	4.0	5.0	NaN	NaN	5.0	NaN	NaN	NaN