

추천시스템의 이해와 구현

1

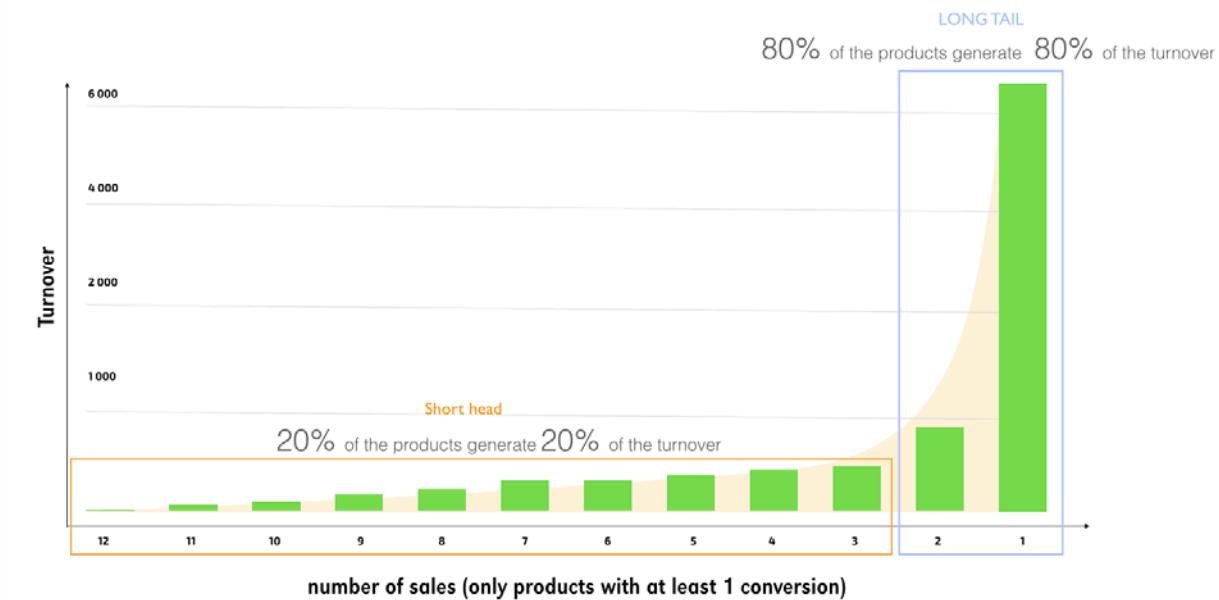
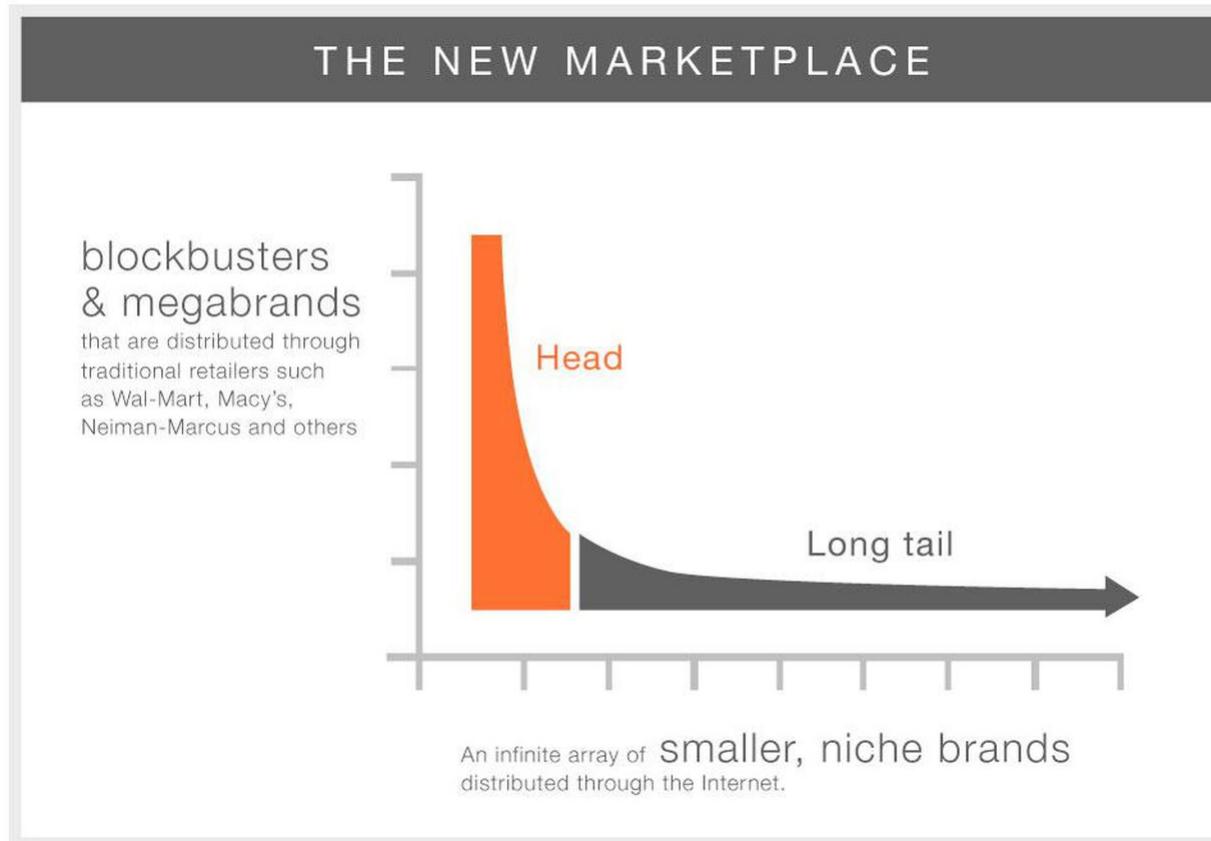
추천 시스템 개요

Recommender System Introduction

등장 배경

■ 시장 환경의 변화

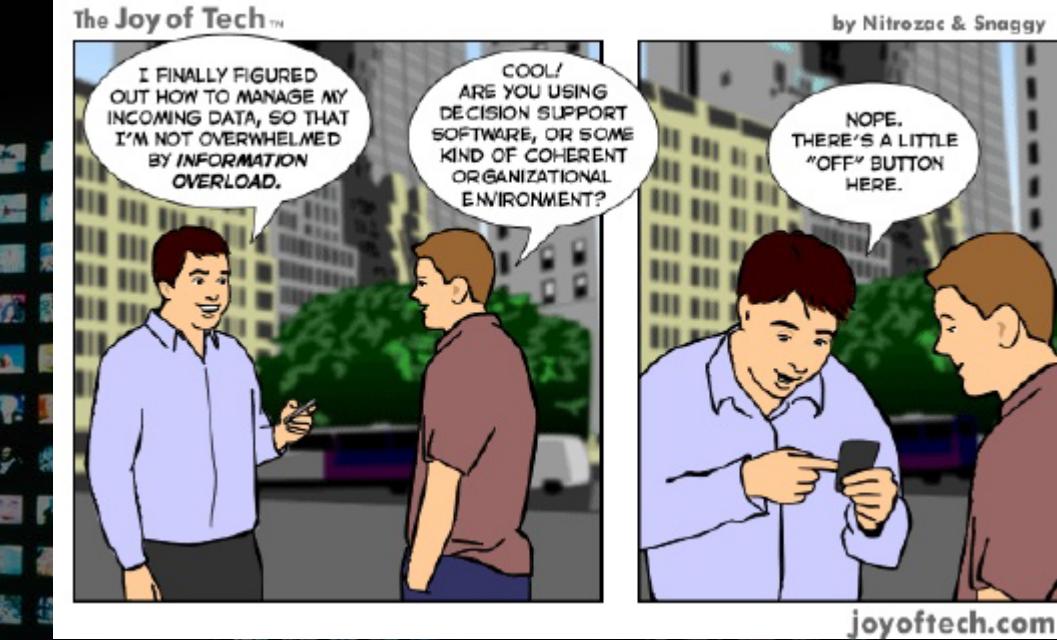
- ▶ 온라인 쇼핑 등의 e-commerce 활성화로 long-tail(긴꼬리) 시장 주목



등장 배경

■ 정보 과부하 (information overload) 문제의 해결을 위한 개인화 서비스의 대두

- ▶ 인간이 처리할 수 있는 정보량 이상의 정보는 소비자의 학습과 의사결정을 오히려 방해



등장 배경

■ 개인화 서비스 중 하나인 추천 시스템은 각 개인의 선호를 분석하여 적합한 컨텐츠를 제공

- ▶ 다양한 기업에서 추천 시스템을 활용하여 기업의 경쟁력을 확보



Google



LinkedIn



Quora

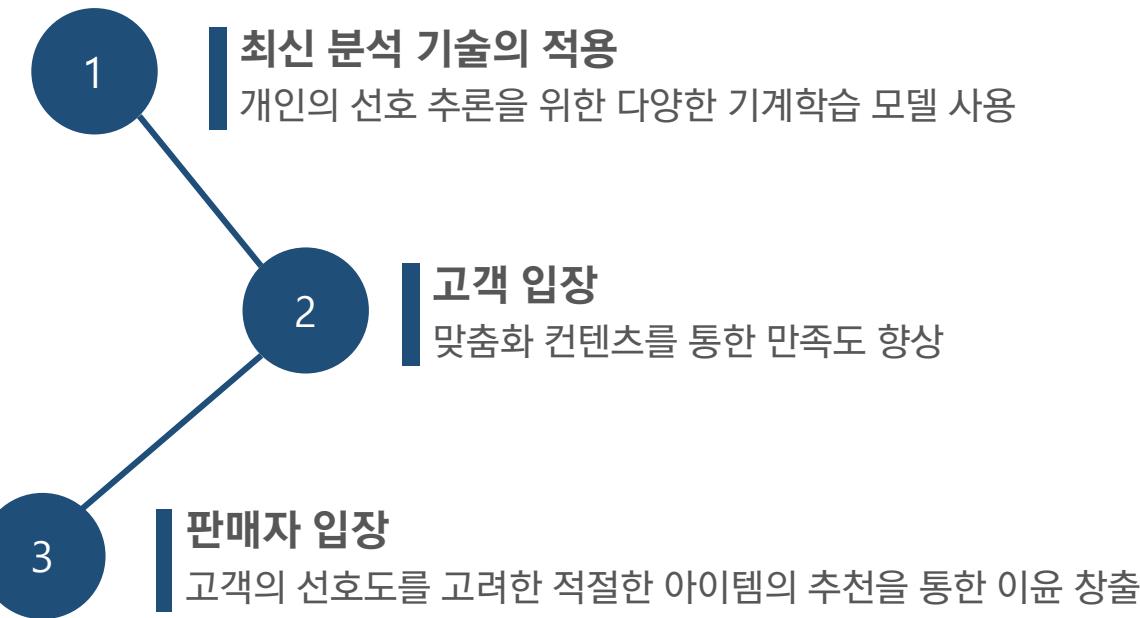


추천 시스템의 정의 및 효과

■ 추천 시스템의 정의

▶ 개별 고객 선호도를 파악하여 개인별로 차별화된 서비스 또는 컨텐츠를 제공하는 시스템

- 개별 고객의 선호도를 파악하기 위하여 고객 개개인의 동적인 행위를 분석하여 선호 체계를 모델링
- 기계 학습을 통해 각 개인별 맞춤화 컨텐츠 또는 서비스를 제공



기업 활용 사례



- 대표적인 추천 시스템 기법인 협업필터링 기법을 바탕으로 개발된 아이템 기반 협업 필터링
- 전체 판매의 약 35%가 추천 시스템을 통해 이루어짐



- 100만 달러(약 10억원)의 Netflix Prize 대회를 통해 축적된 100여개 이상의 추천 알고리즘을 양상화
- 자사의 성공 요인을 신뢰성 있는 추천시스템으로 언급



- 협업 필터링과 자연어 처리 모델, 오디오 분석 모델을 결합한 하이브리드 음악 추천 시스템
- 소비자 경험을 높이기 위한 추천시스템의 도입으로 현재 세계 음악 스트리밍 시장 점유율이 40%

2

대표적인 추천 기법

Recommender System Techniques

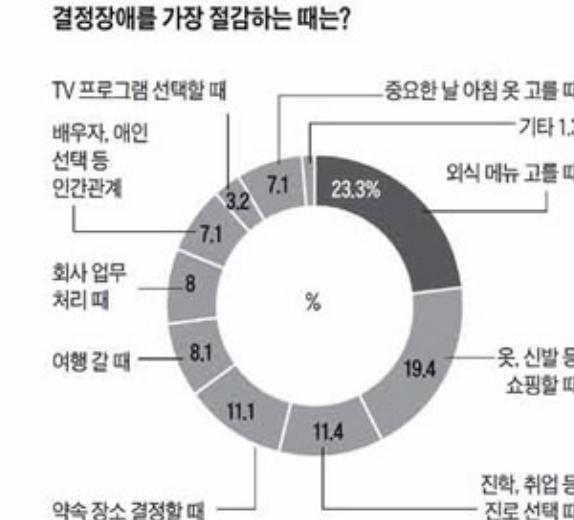
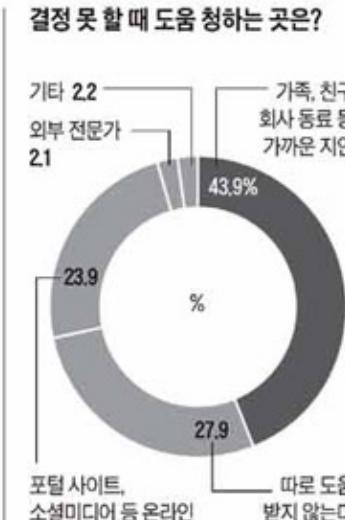
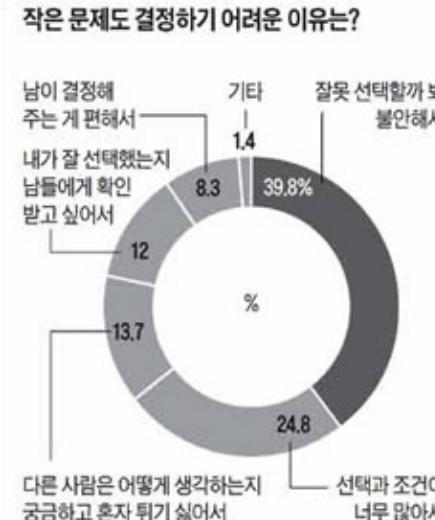
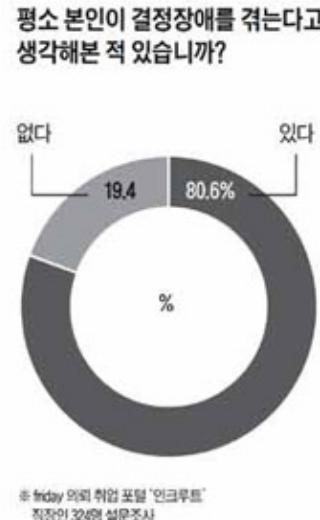
대표적인 추천 시스템 기법 개요

내용 기반 필터링(Content-based Filtering; CB) 기법

- ▶ 아이템의 컨텐츠를 직접 분석하여 아이템과 아이템 또는 아이템과 사용자 선호도간 유사성을 토대로 추천

협업 필터링(Collaborative Filtering; CF) 기법

- ▶ 대규모의 기존 사용자 구매정보를 분석하여 해당 사용자와 비슷한 성향의 사용자들이 선호한 항목 또는 유사한 항목을 추천



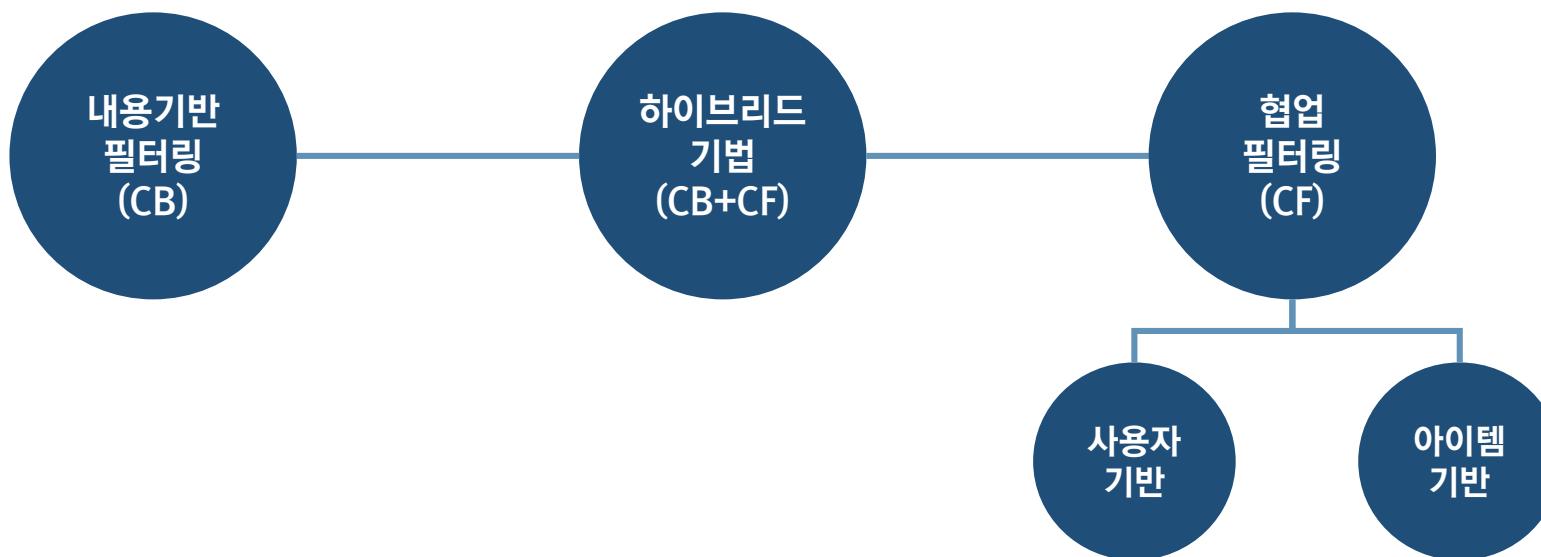
대표적인 추천 시스템 기법 개요

■ 잠재 요인 모델(Latent Factor Model)

- ▶ 사용자 구매정보에서 아이템과 사용자 간의 잠재 요인을 발견하는 모형을 사용하여 알려지지 않은 선호도를 추정

■ 하이브리드(Hybrid) 기법

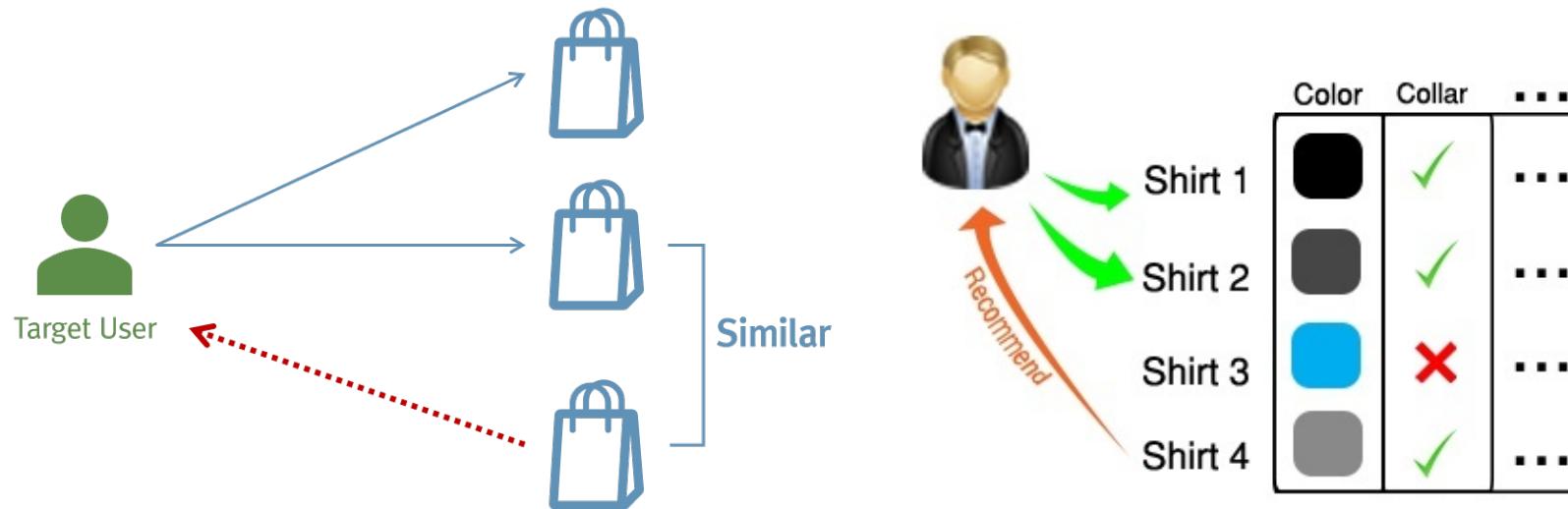
- ▶ 내용 기반 기법과 협업 필터링의 장점을 결합하여 추천



Content-based Filtering

내용 기반 필터링(Content-based Filtering; CB) 기법

- ▶ 아이템의 컨텐츠를 직접 분석하여 추천하는 기법
 - 목표 고객이 구매한 아이템과 아이템 또는 아이템과 사용자 선호도 간의 유사성을 토대로 추천



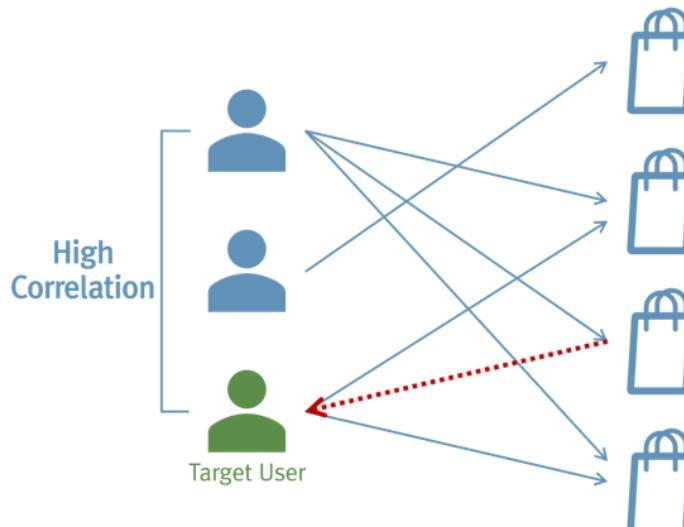
Collaborative Filtering

I 협업 필터링 (Collaborative Filtering; CF)

- ▶ 학계 및 산업계에서 가장 성공적으로 알려진 기법
 - 목표 고객의 구매 기록을 바탕으로 유사한 사용자 또는 아이템을 탐색하여 추천하는 기법
 - 유사도 측정 대상에 따라 사용자 기반(User-based CF)과 아이템 기반(Item-based CF)로 구분

User-based CF

- 목표 고객과 유사한 구매 이력을 보이는 이웃 고객들의 상품에 대한 선호를 바탕으로 추천



	Item1	Item2	Item3	Item4	Item5	Correlation
Target	X	Rec1	X	Rec2		
User1	X	X	X			0.667
User2		X		X	X	-1
User3	X			X		0.167
User4			X	X	X	-0.167
User5		X		X		-0.667

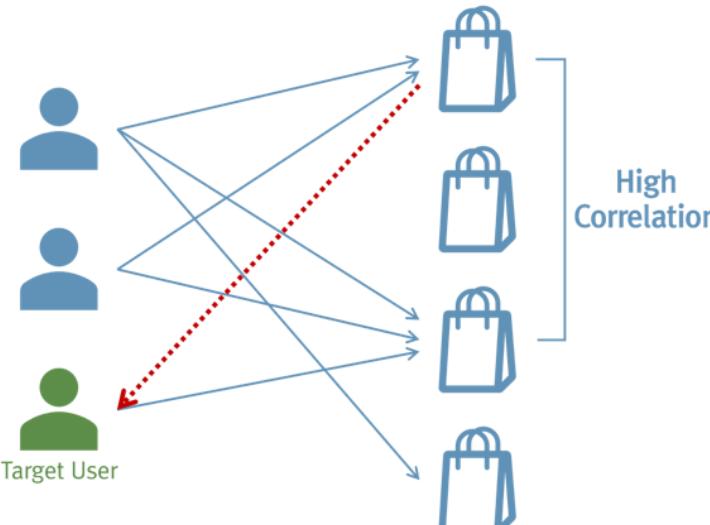
Collaborative Filtering

I 협업 필터링 (Collaborative Filtering; CF)

- ▶ 학계 및 산업계에서 가장 성공적으로 알려진 기법
 - 목표 고객의 구매 기록을 바탕으로 유사한 사용자 또는 아이템을 탐색하여 추천하는 기법
 - 유사도 측정 대상에 따라 사용자 기반(User-based CF)과 아이템 기반(Item-based CF)로 구분

Item-based CF

- 상품들 간의 유사성을 측정하여 목표 고객이 어떤 상품을 선호하는지 예측하여 추천

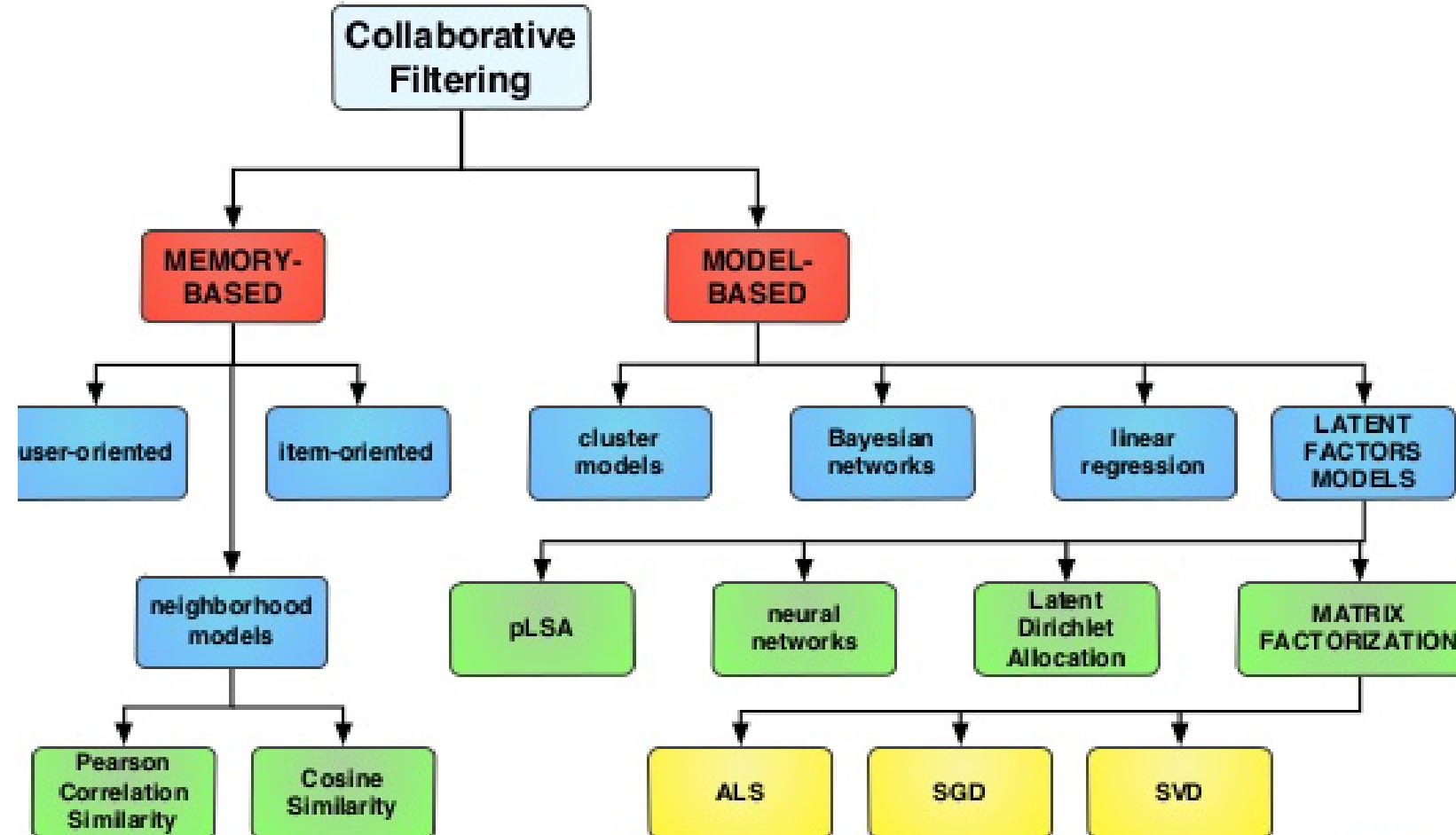


	Item1	Item2	Item3	Item4	Item5
Target	X		Rec2		Rec1
User1	X	X	X		X
User2			X		X
User3	X			X	
User4				X	X
User5			X		X
Correlation		-0.33	0.33	-0.71	0.71

Collaborative Filtering

I 협업 필터링의 구분

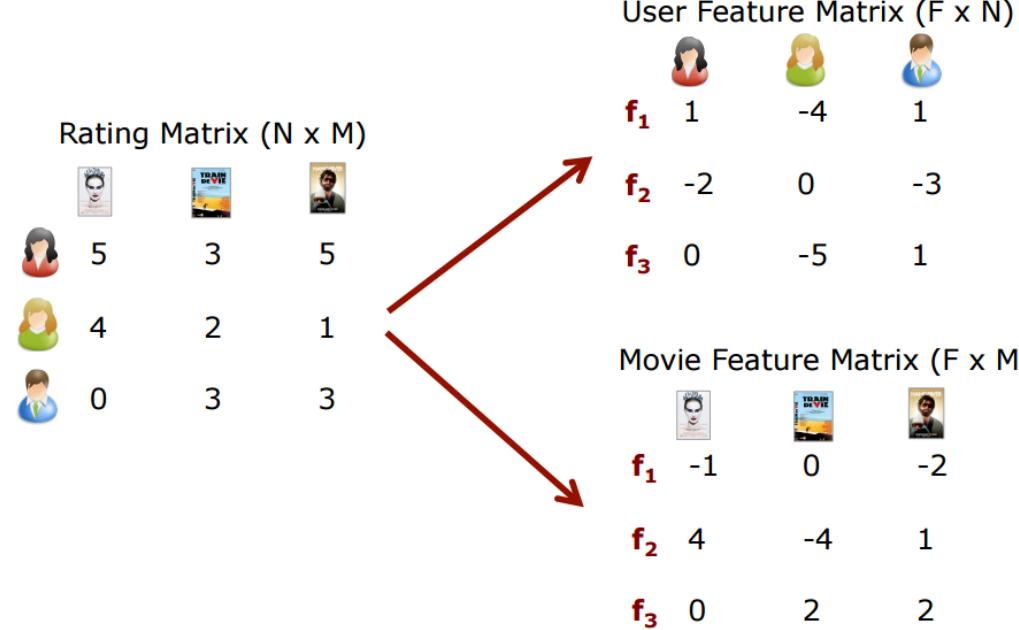
- ▶ Lazy Learner Vs. Eager Learner



Latent Factor Models

희박한 구매 기록을 기반으로 사용자와 아이템의 잠재 요인 탐색

- ▶ 차원 축소 기법으로 Matrix Factorization 또는 SVD를 사용하여 사용자와 아이템을 나타내는 잠재 요인 모델 생성
- ▶ Netflix Prize에서 가장 높은 성과를 보인 모델



The diagram shows two sparse matrices side-by-side, separated by a large 'X' symbol.

User Feature Matrix ($F \times N$)

	Feature 1	Feature 2
User 1	?	?
User 2	?	?
User 3	?	?
User 4	?	?
User 5	?	?

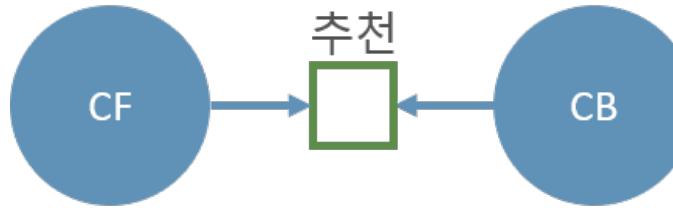
Movie Feature Matrix ($F \times M$)

	Item 1	Item 2	Item 3	Item 4	Item 5
Item 1	?	3	?	3	?
Item 2	4	?	?	2	?
Item 3	?	?	3	?	?
Item 4	3	?	4	?	3
Item 5	4	3	?	4	?

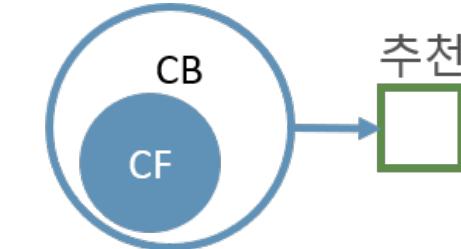
Hybrid Models

CB와 CF의 장점을 결합하는 기법

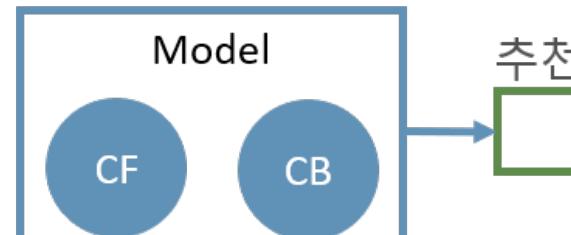
- ▶ CB와 CF를 결합하는 방법에 따라 4가지 기법으로 분류



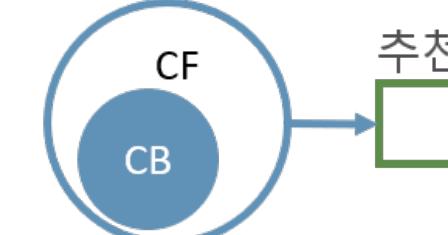
(a) 가중평균합



(b) CF의 특성을 CB에 포함



(c) 통합 모델



(d) CB의 특성을 CF에 포함

추천 시스템 주요 이슈

1

신규 고객 및 상품 (Cold Start)



새로운 사용자나 상품의 경우
구매 정보가 부족하여
추천이 어려움

2

데이터의 희박성 (Sparsity)

bookNum 회원번호	1	2	3	4	5	6
89002	0	0	0	0	0	0
89013	0	0	0	0	0	0
89019	0	0	0	0	0	0
89021	0	0	0	0	0	0
89026	0	0	0	0	0	0
89028	0	0	0	0	0	0
89033	0	0	0	1	0	0
89034	0	0	0	0	0	0
89041	0	0	0	0	0	0
89042	0	0	0	0	0	0
89051	0	0	0	0	0	0
89059	0	0	0	0	0	0
89070	0	0	0	0	0	0
89085	0	0	0	0	0	0
89087	0	0	0	0	0	0
89089	0	0	0	0	0	0
89092	0	0	0	0	0	0

3

확장성 (Scalability)



새로운 사용자나 상품의 경우
구매 정보가 부족하여
추천이 어려움

구매 정보 등 선호 추론에
데이터가 충분하지 않을 경우
추천 정확도가 현저히 저하

사용자나 아이템의 수가
늘어날 수록 계산량이
지수적으로 증가

3

추천 시스템 평가

Evaluation metrics for Recommender System

사용자 선호 데이터의 구분

명시적 선호(Explicit Rating) vs 암시적 선호(Implicit Rating)

- ▶ 사용자의 선호를 직접적으로 파악 가능한 경우와 아닌 경우가 존재
- ▶ 명시적 선호(Explicit rating) : 평점 등 사용자의 선호가 수치로 존재
- ▶ 암시적 선호(Implicit rating) : 구매 기록 등 사용자의 선호를 간접적으로 측정할 수 있는 데이터

명시적 선호 데이터의 특징

- ▶ 사용자의 선호를 추론하는 과정이 불필요
- ▶ 명시적 선호 데이터를 남기는 고객이 극 소수
- ▶ 명시적 선호 데이터의 왜곡 가능성

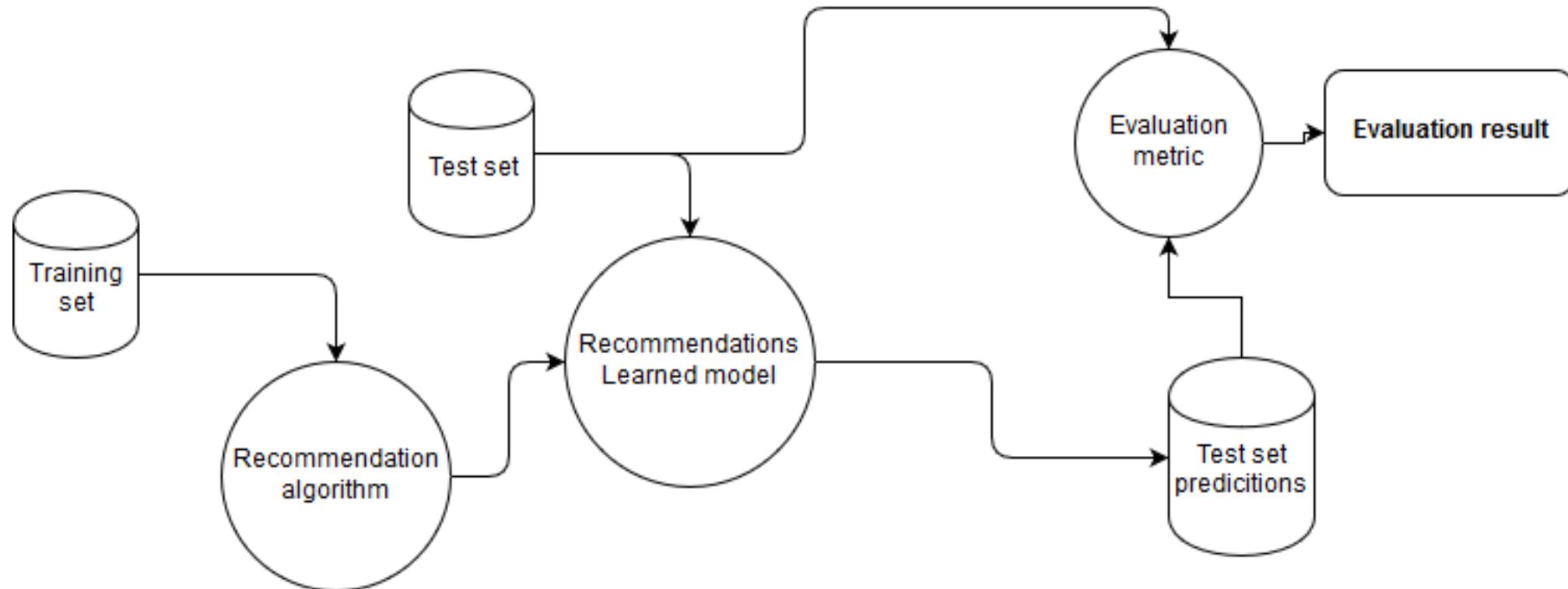
암시적 선호 데이터의 특징

- ▶ 명시적 선호 데이터 획득의 어려움으로 암시적 선호 데이터를 활용하려는 노력이 지속적으로 발전
- ▶ 구매 기록 등의 로그 데이터로부터 선호를 추론하는 모델이 필요
- ▶ Information Utilization Problem : 과연 데이터가 선호를 나타낼 수 있는가?

추천 시스템의 성과 측정

직접 실험의 어려움

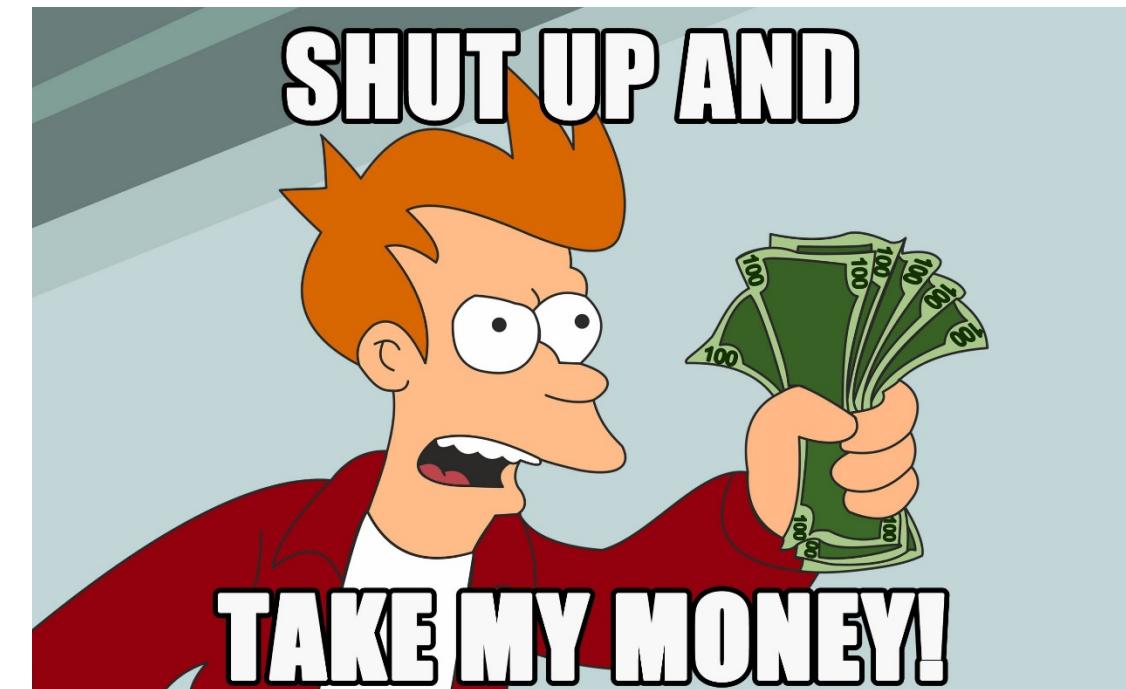
- ▶ 머신 러닝 기법의 평가와 유사하게 Hold out Method 사용



추천 시스템의 성과 측정

■ 추천 시스템의 평가 기준

- ▶ 정확도(Accuracy) : 다양한 평가기준을 통해 해당 시스템이 정확한 추천을 제공하는지 측정
- ▶ 다양성(Diversity) : 추천 시스템이 다양한 아이템을 추천할 수 있는지 평가
- ▶ Serendipity : 상품 추천의 우연성
- ▶ 속도(Speed) : 추천 시스템이 추천 정보를 제공하는 데 소요된 시간



추천 시스템의 성과 측정

■ 명시적 선호 데이터의 정확도 측정 방법

- ▶ 예측값과 실제값의 오차를 측정
- ▶ RMSE (Root Mean Squared Error)

$$\text{RMSE} = \sqrt{\frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2}$$

- ▶ MAE (Mean Absolute Error)
- ▶ $\text{MAE} = \frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}|$
- ▶ FCP(Fraction of Concordant Pairs) : 평점에 의한 순위를 기준으로 오차 측정

추천 시스템의 성과 측정

■ 구매 기록 데이터의 정확도 측정 방법

▶ 의사결정 지원 Metrics

		Predicted class		
		P	N	
Actual Class	P	True Positives (TP)	False Negatives (FN)	$precision = \frac{TP}{TP + FP}$
	N	False Positives (FP)	True Negatives (TN)	$recall = \frac{TP}{TP + FN}$
				$F1 = \frac{2 \times precision \times recall}{precision + recall}$
				$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$
				$specificity = \frac{TN}{TN + FP}$

4

Surprise 패키지

Surprise Package

Surprise 패키지 개요

■ 파이썬에서 CF를 비롯한 기본 추천 시스템을 구현하는 패키지

- ▶ 세부적인 파라미터 설정 및 수식은 다음을 참고
 - <https://surprise.readthedocs.io/en/stable/>
 - <http://courses.ischool.berkeley.edu/i290-dm/s11/SECURE/a1-koren.pdf>
 - <http://www.ijcai.org/Proceedings/13/Papers/449.pdf>

■ 지원 추천 시스템

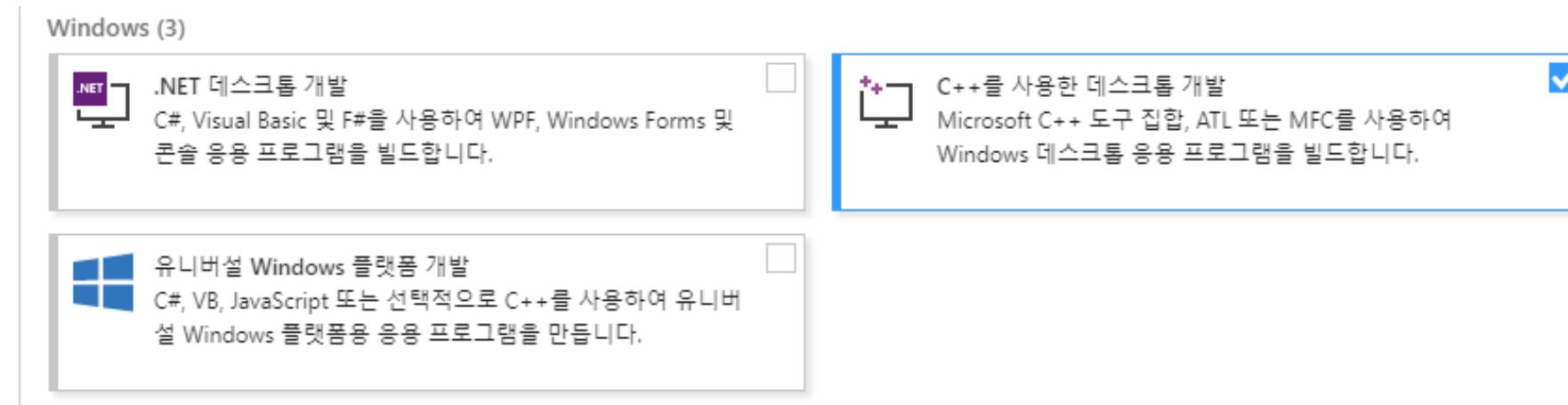
- ▶ 베이스라인 모형
- ▶ CF
 - User-based CF
 - Item-based CF
 - Matrix Factorization
 - SVD
- ▶ Content-based Recommendation

Surprise 패키지 개요

■ 패키지 설치

- ▶ Surprise는 Microsoft Visual C++를 사용함에 따라 먼저 설치 필요

- <https://visualstudio.microsoft.com/downloads/>
- 설치시 C++를 사용한 데스크톱 개발만 체크



■ 패키지 설치

- ▶ pip 이용 설치 : !pip install surprise
- ▶ Github에서 repository를 clone하여 설치
 - <https://github.com/NicolasHug/Surprise>
 - 압축 해제 후 해당 폴더에서 python setup.py install 실행

The screenshot shows the GitHub repository page for the Surprise package. Key details include:

- 574 commits
- 7 branches
- 9 releases
- 21 contributors
- BSD-3-Clause license
- Branch: master
- New pull request
- Clone or download
- Clone with HTTPS
- Use Git or checkout with SVN using the web URL: <https://github.com/NicolasHug/Surprise.git>
- Open in Desktop
- Download ZIP

The repository page lists several recent commits:

- ZachGlassman and NicolasHug allow to skip confirmation for load_builtin (#222)
- .github: Update issue_template.md
- doc: Updated documentation to reflect missing implementation of
- examples: rating_scale now set at dataset creation
- surprise: allow to skip confirmation for load_builtin (#222)

Surprise 기초 사용 실습

■ 데이터 로딩 및 확인

▶ GroupLens에서 제공하는 MovieLens 샘플 평점 데이터 로딩

- 10만개 샘플 데이터셋('ml-100k')로딩 후 데이터 프레임 생성

```
1 import surprise  
2 import pandas as pd  
3 import numpy as np
```

```
1 data = surprise.Dataset.load_builtin('ml-100k')
```

```
1 df = pd.DataFrame(data.raw_ratings, columns=['user','item','rate','id'])
```

```
1 del df['id']
```

```
1 df.head()
```

	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

Surprise 기초 사용 실습

■ 데이터 로딩 및 확인

▶ 사용자 데이터 사용시

- userID / itemID / rating 순으로 정렬된 데이터프레임 사용
- reader에서 rating_scale 설정

```
1 from surprise import Dataset
2 from surprise import Reader
3
4 reader = Reader(rating_scale=(최소값, 최대값))
5 data = Dataset.load_from_df(데이터프레임명, reader)
```

Surprise 기초 사용 실습

데이터 로딩 및 확인

▶ 데이터 희박정도(Sparsity) 확인

- #### ■ 테이블로 확인

```
1 dt = df.set_index(['user','item']).unstack()
2 dt.iloc[0:10, 0:10].fillna('')
```

Surprise 기초 사용 실습

■ 데이터 로딩 및 확인

▶ 데이터 희박정도(Sparsity) 확인

- 그래프로 확인

```
1 import matplotlib.pyplot as plt  
2 %matplotlib inline
```

```
1 plt.figure(figsize=(15,15))  
2 plt.imshow(dt)  
3 plt.grid(False)  
4 plt.xlabel('item')  
5 plt.ylabel('user')  
6 plt.title('Rate Matrix')
```

```
Text(0.5,1,'Rate Matrix')
```



Surprise 기초 사용 실습

■ 베이스 라인 모형

- ▶ 사용자 아이디와 상품 아이디로 예측하는 단순한 모형

- 사용자와 상품의 평균 평점의 합으로 계산

$$\hat{r}_{ui} = \mu + b_u + b_i$$

- 다음 오차 함수의 최소화가 목표

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - (\mu + b_u + b_i))^2$$

- 과적합을 줄이기 위한 정규항 추가

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - (\mu + b_u + b_i))^2 + \lambda (b_u^2 + b_i^2)$$

Surprise 기초 사용 실습

■ 베이스 라인 모형

▶ 모형 학습 알고리즘

- SGD (Stochastic Gradient Descent)
 - reg : 정규화 가중치 (기본값=0.02)
 - learning_rate : 학습률 (기본값=0.005)
 - n_epochs : 반복횟수 (기본값=20)
- ALS (Alternating Least Squares)
 - reg_i : 상품에 대한 정규화 가중치 (기본값=10)
 - reg_u : 사용자에 대한 저육화 가중치 (기본값=15)
 - n_epochs : 반복횟수 (기본값=10)

Surprise 기초 사용 실습

■ 베이스 라인 모형

▶ 모형 학습 방법

- 모형 객체 생성
- 모형 셋 생성
- 모수 추정 및 예측
- 성능 평가 함수로 평가

▶ 성능평가 기준

- RMSE
- MAE
- FCP

```

1 from surprise.model_selection import KFold
2
3 bsl_param = {'method': 'als', 'n_epochs':5, 'reg_u':12, 'reg_i':5}
4 model = surprise.BaselineOnly(bsl_options=bsl_param)
5
6 acc = np.zeros(3)
7 cv = KFold(3)
8 for i, (trainset, testset) in enumerate(cv.split(data)):
9     model.fit(trainset)
10    pred = model.test(testset)
11    acc[i] = surprise.accuracy.rmse(pred, verbose=True)
12
13 acc.mean()

```

Estimating biases using als...

RMSE: 0.9455

Estimating biases using als...

RMSE: 0.9432

Estimating biases using als...

RMSE: 0.9431

0.9439531573610811

Surprise 기초 사용 실습

■ 베이스 라인 모형

- ▶ 자동으로 여러 번 테스트를 반복할 수 있는 cross_validate 제공

```
1 from surprise.model_selection import cross_validate  
2 cross_validate(model, data)
```

Estimating biases using als...
Estimating biases using als...
Estimating biases using als...
Estimating biases using als...
Estimating biases using als...

```
{'test_rmse': array([0.93827201, 0.93593255, 0.94762093, 0.94463634, 0.94185712]),  
'test_mae': array([0.74240804, 0.74201526, 0.74980574, 0.74993755, 0.74350559]),  
'fit_time': (0.09604120254516602,  
 0.14729094505310059,  
 0.15619254112243652,  
 0.14063286781311035,  
 0.10198974609375),  
'test_time': (0.07810449600219727,  
 0.1250009536743164,  
 0.10938739776611328,  
 0.12347054481506348,  
 0.09372496604919434)}
```

협업 필터링 (CF) 구현 실습

■ 이웃기반 모형의 유사도 기준

▶ 평균 제곱 차이 유사도 (Mean Squared Difference)

- 유클리드 공간에서의 거리 제곱에 비례하는 값
- 유사도는 msd값의 역수로 계산
- 0이 되는 경우를 대비하여 1을 더하여 역수

▶ 코사인 유사도

▶ 피어슨 유사도

- 두 벡터의 상관계수
- 1~-1 : 0일 경우에는 상관관계가 없음

▶ 피어슨-베이스라인 유사도

- 각 벡터의 기대값을 단순 평균이 아니라 베이스라인 모형에서 예측한 값 사용

▶ 유사도 설정 옵션

- name : 사용할 유사도의 종류 (기본값='MSD')
- user_based : True인 경우 user-based CF, False인 경우 item-based CF
- min_support : 두 사용자 또는 상품에서 공통적으로 있는 평점 수의 최소값
- shirinkage : Shirinkage 가중치 (기본값=100)

협업 필터링 (CF) 구현 실습

■ 이웃기반 모형의 이웃 평점 사용 방식

- ▶ KNNBasic : 평점들의 가중평균

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)}$$

- ▶ KNNWithMeans : 평점들의 평균값 기준으로 가중 평균

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - \mu_v)}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)}$$

- ▶ KNNBaseline : 평점들을 베이스라인 모형의 값 기준으로 가중 평균

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - b_{vi})}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)}$$

협업 필터링 (CF) 구현 실습

I Latent Factor 모형

- ▶ Matrix Factorization
 - 모든 사용자와 상품에 대해 오차 함수를 최소화하는 요인 벡터 탐색
 - SVD는 MF 문제를 푸는 방법 중 하나로 surprise 패키지에서는 matrix_factorization 서브패키지에서 SVD와 SVDpp 클래스 제공

II Slope One

- ▶ 사용자간 공유되는 아이템의 수를 반영하여 추천

III CoClustering

- ▶ 군집 분석을 수행한 뒤 아이템을 추천