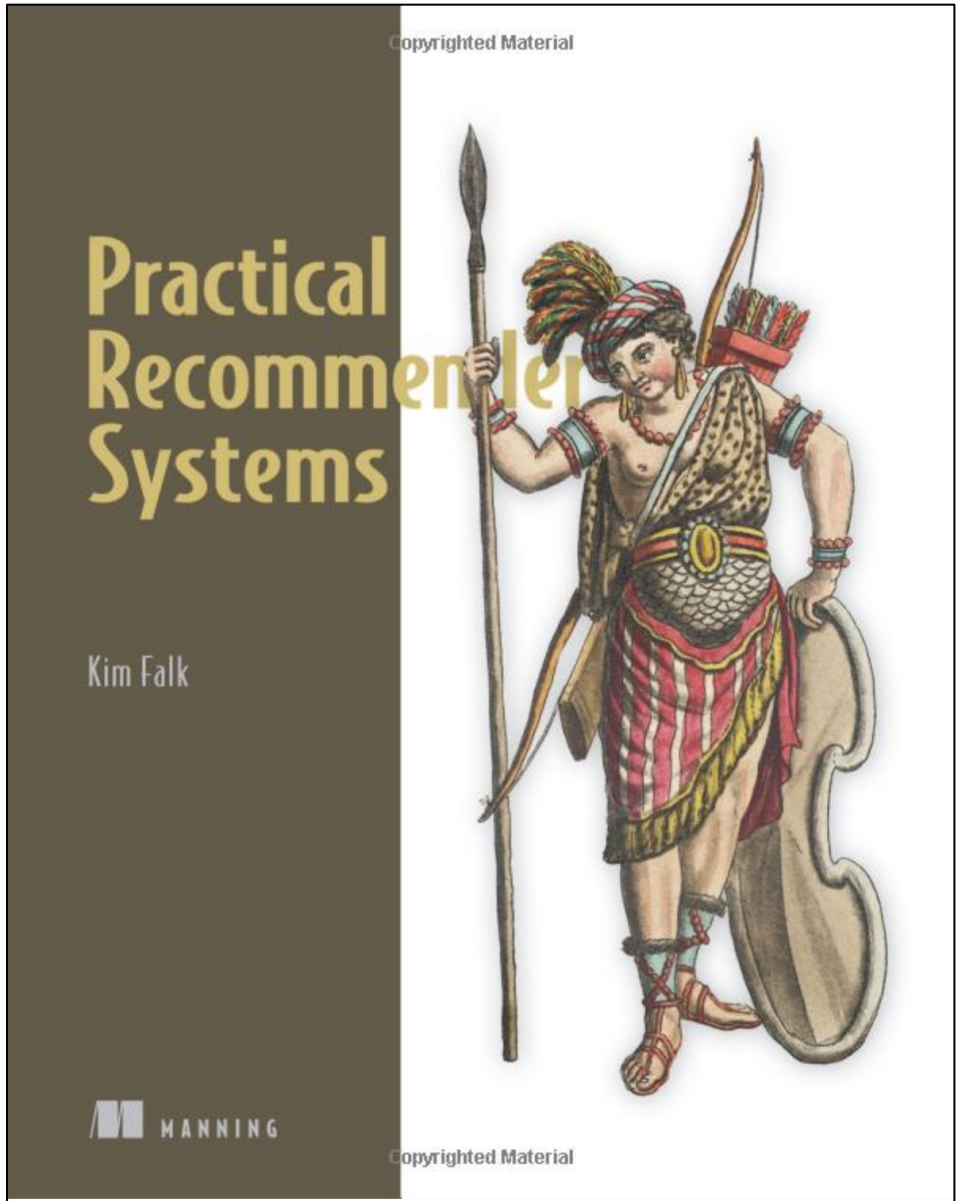
SUMMARY

PAI
PUBLIC AI

이 수업을 설계하면서 다루었던 참조자료들



이 책에서 다룬 내용들

- 연관분석
- Collaborative Filtering
- Bayesian Personalized Ranking

PAI
PUBLIC AI

이 수업을 설계하면서 다루었던 참조자료들

Deep Learning based Recommender System: A Survey and New Perspectives

SHUAI ZHANG, University of New South Wales
LINA YAO, University of New South Wales
AIXIN SUN, Nanyang Technological University
YI TAY, Nanyang Technological University

With the ever-growing volume of online information, recommender systems have been an effective strategy to overcome such information overload. The utility of recommender systems cannot be overstated, given its widespread adoption in many web applications, along with its potential impact to ameliorate many problems related to over-choice. In recent years, deep learning has garnered considerable interest in many research fields such as computer vision and natural language processing, owing not only to stellar performance but also the attractive property of learning feature representations from scratch. The influence of deep learning is also pervasive, recently demonstrating its effectiveness when applied to information retrieval and recommender systems research. Evidently, the field of deep learning in recommender system is flourishing. This article aims to provide a comprehensive review of recent research efforts on deep learning based recommender systems. More concretely, we provide and devise a taxonomy of deep learning based recommendation models, along with providing a comprehensive summary of the state-of-the-art. Finally, we expand on current trends and provide new perspectives pertaining to this new exciting development of the field.

CCS Concepts: •**Information systems** → **Recommender systems;**

Additional Key Words and Phrases: Recommender System; Deep Learning; Survey

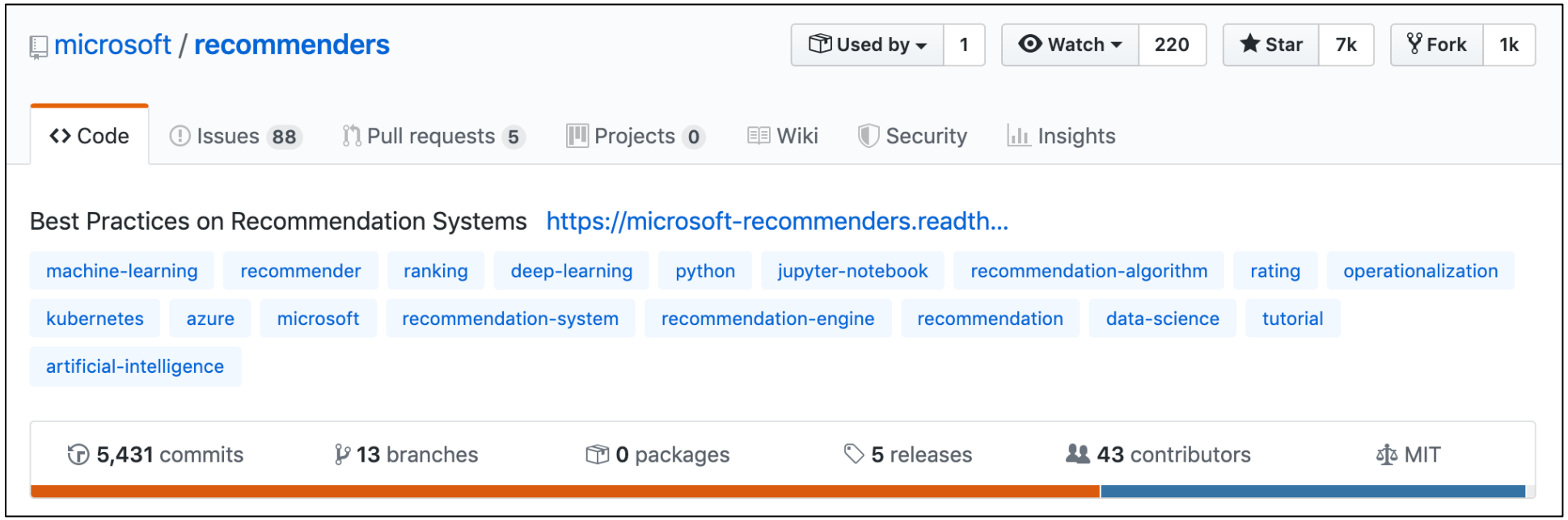
ACM Reference format:
Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2018. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 1, 1, Article 1 (July 2018), 35 pages.
DOI: 0000001.0000001

Deep Learning 관련된 추천시스템 모델이 무엇이 있는지를 요약한 모델

PAI
PUBLIC AI

이 수업을 설계하면서 다루었던 참조자료들

Microsoft에서 제공하는 추천시스템에 관련된 Sample Codes Repository



https://github.com/microsoft/recommenders

PAI
PUBLIC AI

이 수업을 설계하면서 다루었던 참조자료들

Google에서 제공하는 추천시스템

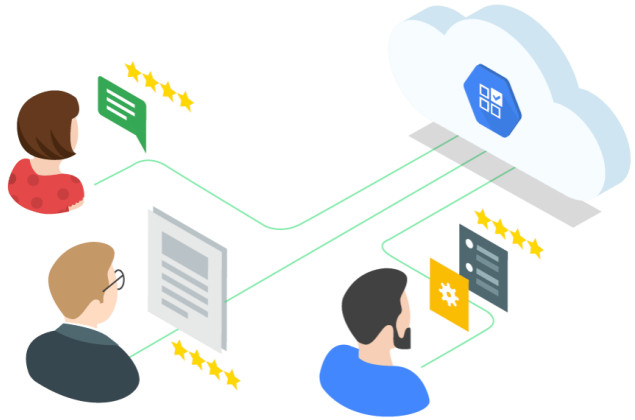
Recommendations AI 베타

규모에 따라 고도로 맞춤화된 제품 추천

문서 보기

고객의 취향에 따른 추천

고객을 잘 이해하고 있음을 보여주면 고객의 신뢰와 충성도를 높일 수 있습니다. Google에서는 수년 동안 Google Ads, Google Search, YouTube와 같은 대표적인 제품 전반에서 추천 콘텐츠를 제공해 왔습니다. Recommendations AI는 그러한 경험을 바탕으로 모든 터치포인트 전반에서 각 고객의 취향이나 선호도에 맞는 맞춤 추천을 제공합니다. 제한된 사용자에게만 제공되는 이 베타 서비스에 대해서는 Google 계정 관리자에게 문의하세요.



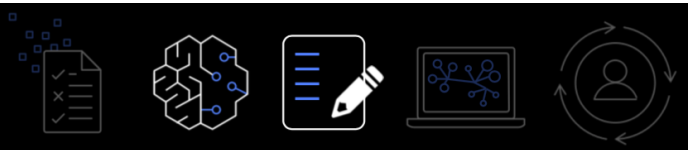
https://cloud.google.com/recommendations?hl=ko

PAI PUBLIC AI

이 수업을 설계하면서 다루었던 참조자료들

AWS에서 제공하는 추천시스템

빌트인 레시피

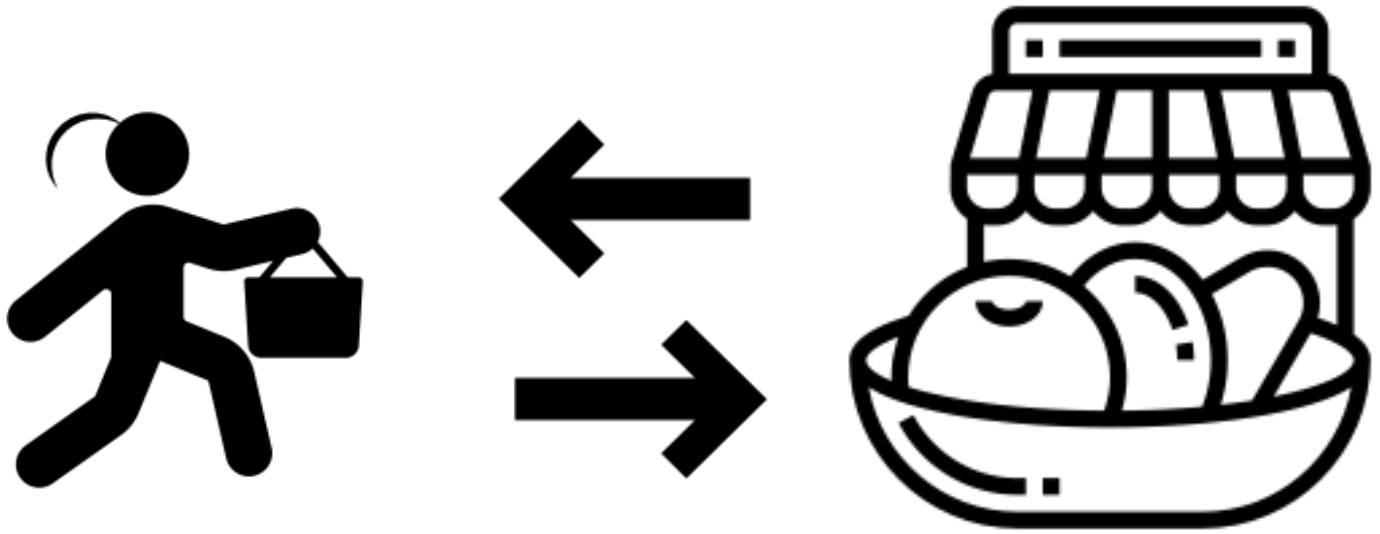


개인화 & 추천을 위한 **최신 기술 기반 알고리즘(Predefined Recipes)**을 다양하게 제공합니다.
아울러, 높은 성능의 모델 생성을 위한 **파라미터 최적화(Hyperparameter Optimization)**,
알고리즘 자동 선택(AutoML)* 기능도 제공합니다.

DeepFM* (Deep Factorization Machines)	Recommended for fast Training and Inference with good general performance
FFNN* (Feed-Forward Neural Network)	A general purpose feed-forward neural network
HRNN* (Hierarchical Recurrent Neural Network)	Recommended when user behavior is changing with time (the evolving intent problem) - **HRNN-coldstart** - **HRNN-metadata**
PersonalizeReranking	Use for personalized reranking of search results or curated lists
Popularity-baseline	Use as a baseline to compare other personalization recipes
SIMS	Item-to-Item similarities. Use for improving item discoverability and in detail pages for the fast performance

PAI
PUBLIC AI

이 수업의 주제. :추천



고객

상점

핵심 이슈 : 고객에게 어떠한 제품을 권할 것인가?

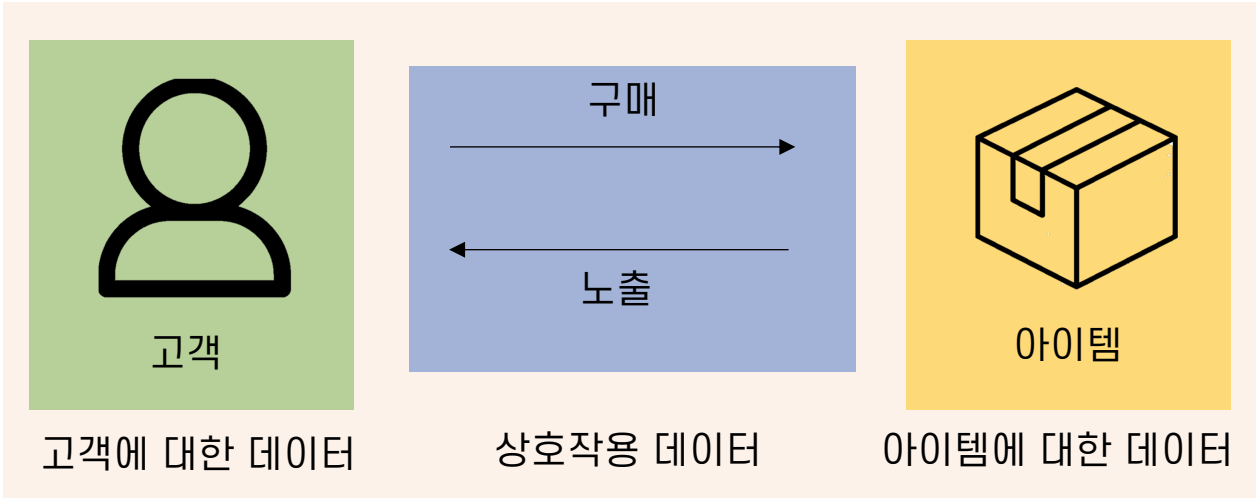
PAI
PUBLIC AI

이 수업의 구성

1. 고객이 첫 방문을 했을 때
   : 고객에 대한 정보가 없을 때

2. 고객이 재차 방문을 했을 때
   : 고객의 이전 구매 정보가 있을 때

3. 고객이 단골이 되었을 때
   : 고객 신원 정보가 있을 때



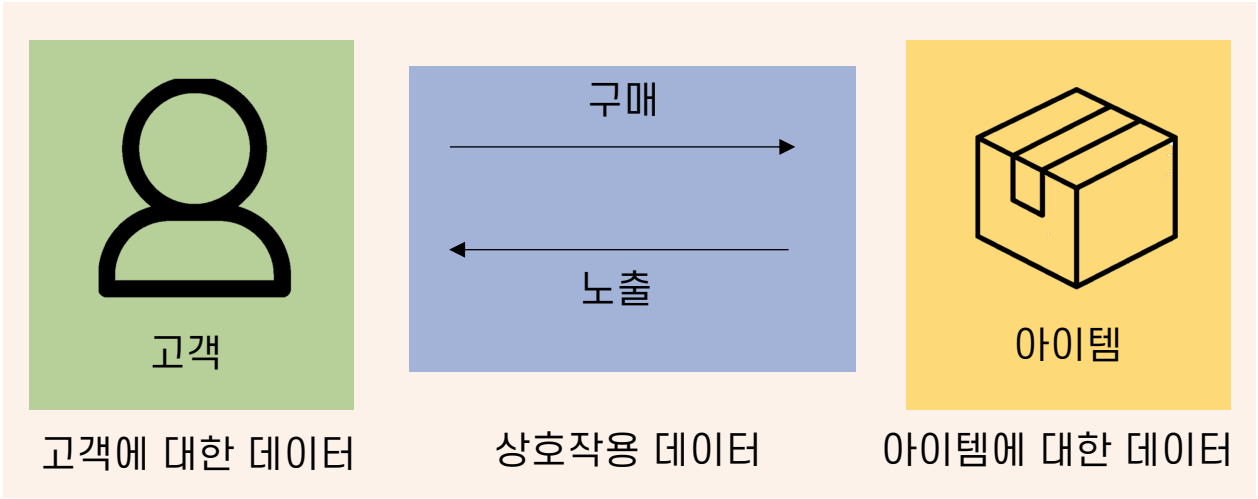
PAI
PUBLIC AI

이 수업의 주요 토픽

1. 고객이 첫 방문을 했을 때
   - 비개인화 추천
   - 연관 분석

2. 고객이 재차 방문을 했을 때
   - Collaborative Filtering

3. 고객이 단골이 되었을 때
   - Factorization Machine

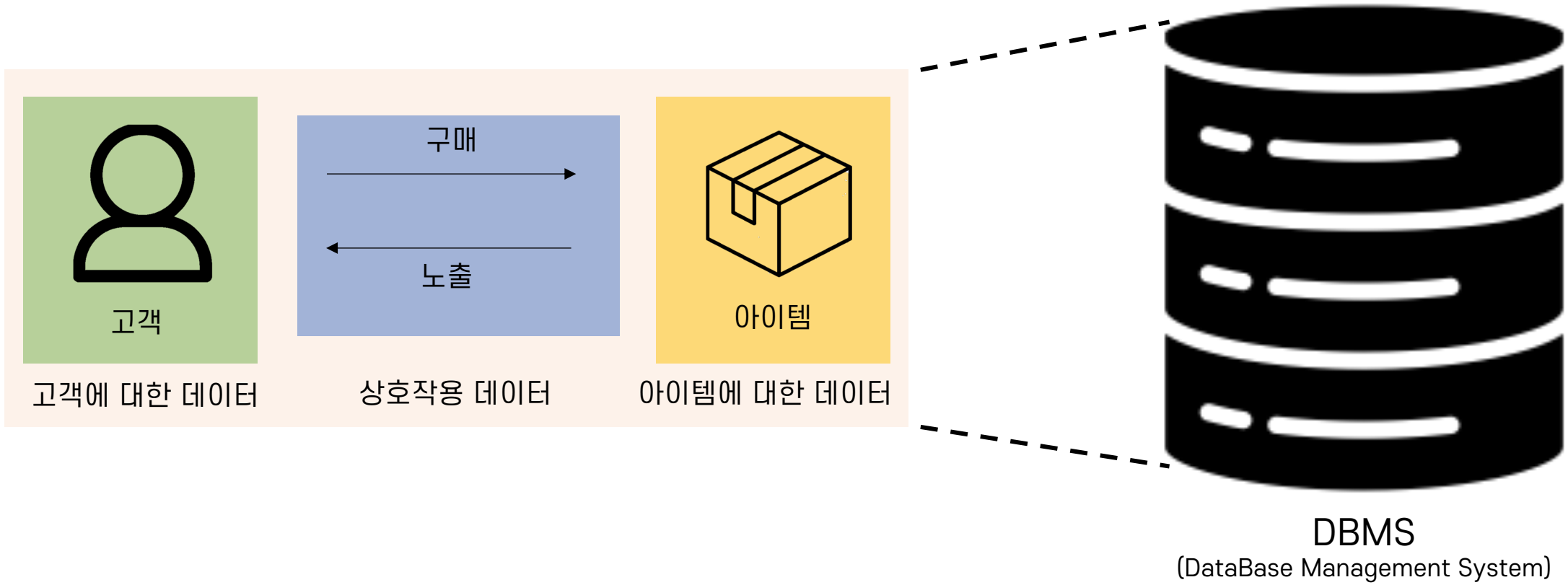


PAI
PUBLIC AI

데이터베이스
<Database>

PAI
PUBLIC AI

추천 시스템에서의 데이터



PAI
PUBLIC AI

오늘 배우게 되는 SQL 구문들

원하는 데이터 셋을 가져오기 위한 방법들

```
SELECT c1, c2 FROM t;
```
Query data in columns c1, c2 from a table

```
SELECT * FROM t;
```
Query all rows and columns from a table

```
SELECT c1, c2 FROM t
WHERE condition;
```
Query data and filter rows with a condition

```
SELECT DISTINCT c1 FROM t
WHERE condition;
```
Query distinct rows from a table

```
SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
```
Sort the result set in ascending or descending order

```
SELECT c1, c2 FROM t
ORDER BY c1
LIMIT n OFFSET offset;
```
Skip *offset* of rows and return the next n rows

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
```
Group rows using an aggregate function

PAI
PUBLIC AI

연관 분석
<Association Analysis>

PAI
PUBLIC AI

연관 분석

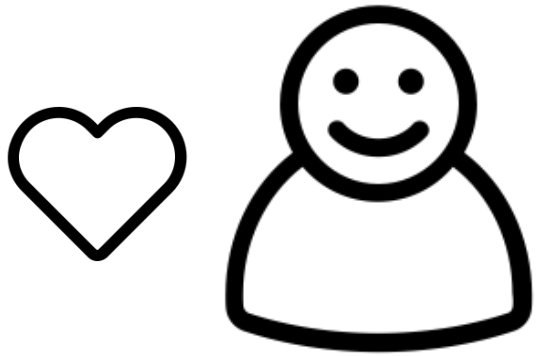
만약 X를 선호(구매)했다면,
Y도 선호(구매)할 것이다

의 연관 관계를 도출하는 것

PAI
PUBLIC AI

스타워즈 2를 너무나 재밌게 본 유저에게
어떤 영화를 추천하는 것이 좋을까?

(1) 지지도 : $\boldsymbol{Support(X)} = \frac{\boldsymbol{freq(X)}}{\boldsymbol{N}}$
: 전체 고객 중에서 영화 X를 선호하는 사람의 비율은?



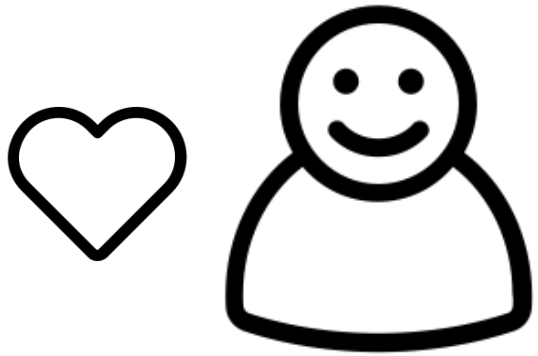
지지도를 기준으로 추천할 경우,
어떤 영화를 선호하던 항상 같은 영화(타이타닉)을
추천하는 문제가 발생

PAI
PUBLIC AI

스타워즈 2를 너무나 재밌게 본 유저에게
어떤 영화를 추천하는 것이 좋을까?

(2) 신뢰도 : $\boldsymbol{confidence(X \rightarrow Y) = \frac{freq(X,Y)}{freq(X)}}$

: 영화 X를 선호한 사람 중에서
영화 Y를 선호하는 사람의 비율은?(조건부 확률)



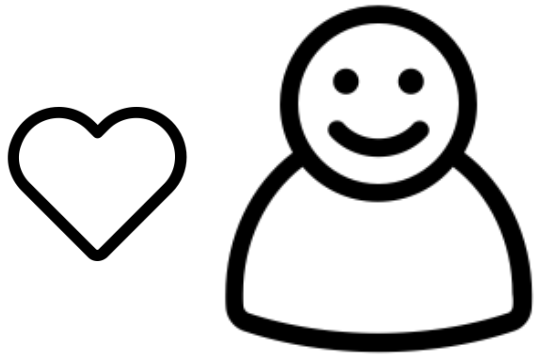
	지지도	신뢰도
	4.1%	50.2%
	3.8%	17.0%
	3.8%	13.1%
	10.0%	29.0%

신뢰도를 기준으로 추천할 경우,
대다수 사람들이 선호하는 영화가
소수의 사람들이 선호하는 영화보다 우선 추천됨

PAI
PUBLIC AI

스타워즈 2를 너무나 재밌게 본 유저에게
어떤 영화를 추천하는 것이 좋을까?

(3) 리프트 = $\mathbf{lift}(X \rightarrow Y) = \frac{confidence(X \rightarrow Y)}{support(Y)}$
: 지지도 대비 신뢰도가 얼마나 올라갔는가?



	지지도	신뢰도	리프트
	4.1%	50.2%	12.2
	3.8%	17.0%	4.4
	3.8%	13.1%	3.5
	10.0%	29.0%	2.9

리프트를 기준으로 추천할 경우,
소수의 사람들이 선호하는 영화이더라도
강한 연관관계가 있으면 영화가 우선 추천됨

PAI
PUBLIC AI

연관 분석의 3가지 지표

(1) 지지도 : $\boldsymbol{Support(X) = \frac{freq(X)}{N}}$

(2) 신뢰도 : $\boldsymbol{confidence(X \rightarrow Y) = \frac{freq(X,Y)}{freq(X)}}$

(3) 리프트 : $\mathbf{lift}\boldsymbol{(X \rightarrow Y) = \frac{confidence(X \rightarrow Y)}{support(Y)} = \frac{N * freq(X,Y)}{freq(X)freq(Y)}}$

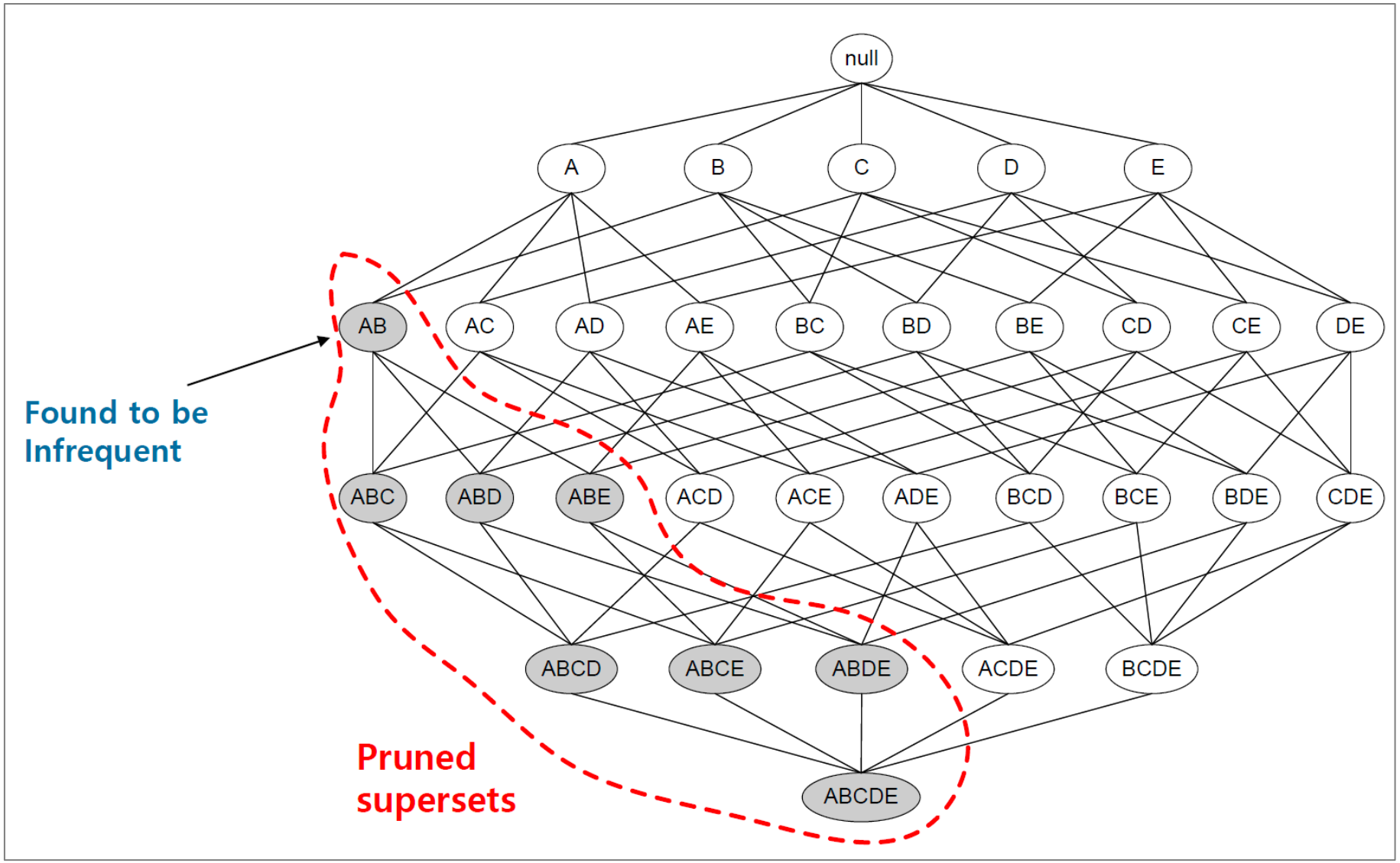
연관분석을 위해 알아야 하는 값 :

$$freq(X), freq(Y), freq(X,Y)$$

- $freq(X)$: 아이템(X)을 선호(구매)한 경우의 수
- $freq(Y)$: 아이템(Y)을 선호(구매)한 경우의 수
- $freq(X,Y)$: 아이템(X,Y)을 동시에 선호(구매)한 경우의 수

PAI
PUBLIC AI

빈발집합 찾기 알고리즘 : Apriori 알고리즘



PAI PUBLIC AI

연관 분석 활용 예제) The YouTube Video Recommendation System, 2010

Figure 1: A screenshot of the recommendations module on the YouTube home page

2.3 Generating Recommendation Candidates

To compute personalized recommendations we combine the related videos association rules with a user's personal activity on the site: This can include both videos that were watched (potentially beyond a certain threshold), as well as videos that were explicitly favorited, "liked", rated, or added to playlists. We call the union of these videos the *seed set*.

In order to obtain candidate recommendations for a given seed set S, we expand it along the edges of the related videos graph: For each video v_i in the seed set consider its related videos R_i. We denote the union of these related video sets as C_1:

$$C_1(S) = \bigcup_{v_i \in S} R_i \tag{2}$$

In many cases, computing C_1 is sufficient for generating a set of candidate recommendations that is large and diverse enough to yield interesting recommendations. However, in practice the related videos for any videos tend to be quite narrow, often highlighting other videos that are very similar to the seed video. This can lead to equally narrow recommendations, which do achieve the goal of recommending content close to the user's interest, but fail to recommend videos which are truly new to the user.

In order to broaden the span of recommendations, we expand the candidate set by taking a limited transitive closure over the related videos graph. Let C_n be defined as the set of videos reachable within a distance of n from any video in the seed set:

$$C_n(S) = \bigcup_{v_i \in C_{n-1}} R_i \tag{3}$$

where $C_0 = S$ is the base case for the recursive definition (note that this yields an identical definition for C_1 as equation (2)). The final candidate set C_{final} of recommendations is then defined as:

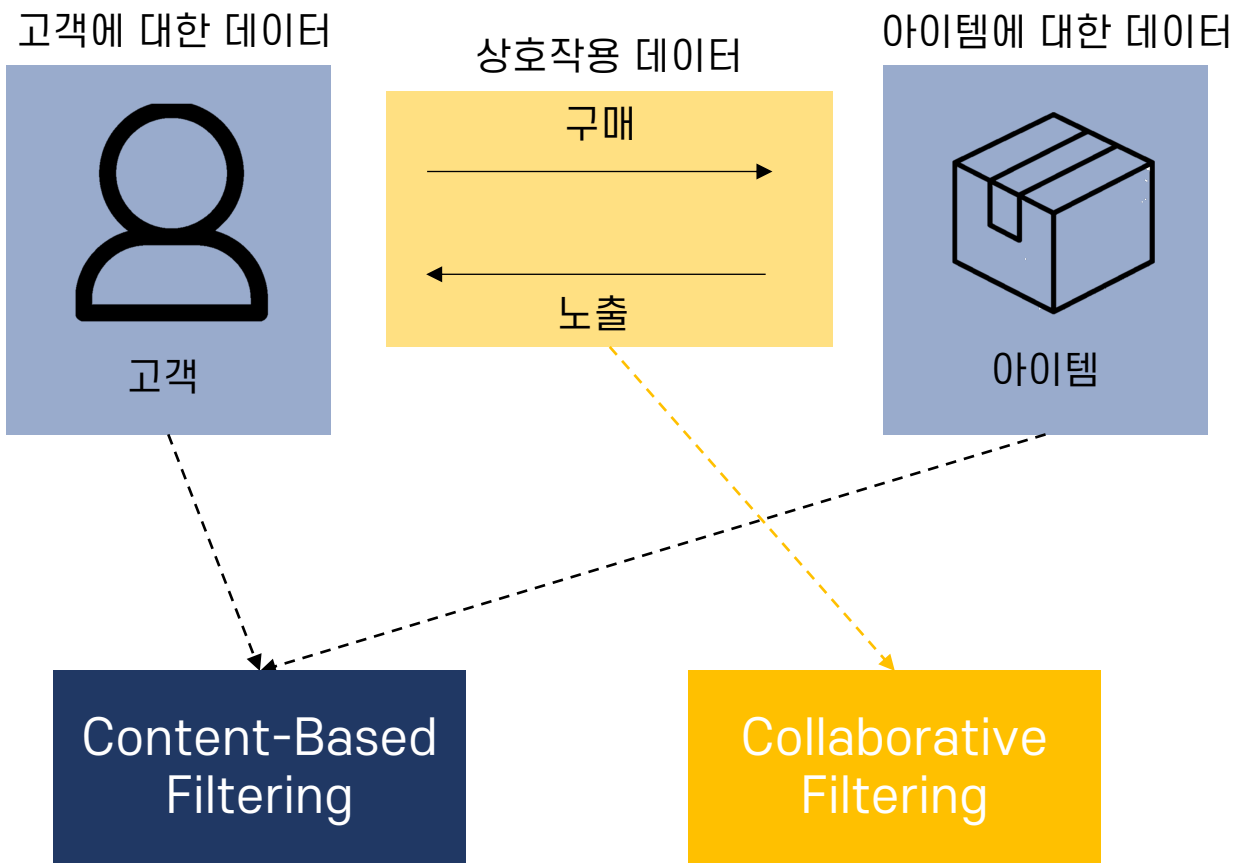
$$C_{final} = (\bigcup_{i=0}^{N} C_i) \setminus S \tag{4}$$

Due to the high branching factor of the related videos graph we found that expanding over a small distance yielded a broad and diverse set of recommendations even for users with a small seed set. Note that each video in the candidate set is associated with one or more videos in the seed set. We keep track of these seed to candidate associations for ranking purposes and to provide explanations of the recommendations to the user.

협업 필터링
<Collaborative Filtering>

추천 시스템의 알고리즘

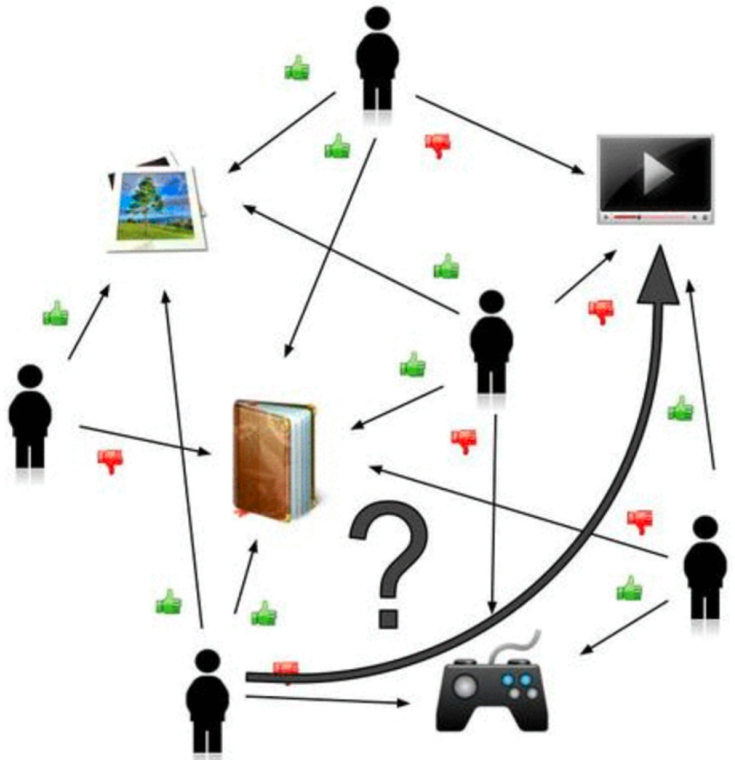
추천 시스템은 크게 콘텐츠 기반 추천과 협업 필터링 기반 추천으로 나뉘어짐



협업 필터링이란?

핵심 아이디어

> 만약 두 명의 사용자가 유사한 관심사를 가지고 있다면,
> 그들은 미래에도 유사한 취향을 가질 것이다.



PAI
PUBLIC AI

CF 알고리즘의 데이터, 상호작용 정보

	user_id	movie_id	rating	rated_at
0	1	2	3.5	1112486027
175	2	3	4.0	974820889
236	3	1	4.0	944919407
423	4	6	3.0	840879227
424	4	10	4.0	840878922
451	5	2	3.0	851527569
517	6	1	5.0	858275452
518	6	3	3.0	858275558
519	6	7	5.0	858275558
541	7	3	3.0	1011208463
542	7	7	3.0	1011208220
817	8	1	4.0	833981871
818	8	3	5.0	833981733
819	8	6	3.0	833982631
820	8	10	4.0	833981834
922	10	1	4.0	943497887

고객이 아이템에게
어떠한 액션을 취했는지에 대한 정보

예시)

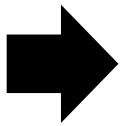
6번 고객이 7번 영화에 평점 5점을 부여하였음

PAI
PUBLIC AI

CF 알고리즘의 데이터, 상호작용 정보

	user_id	movie_id	rating	rated_at
0	1	2	3.5	1112486027
175	2	3	4.0	974820889
236	3	1	4.0	944919407
423	4	6	3.0	840879227
424	4	10	4.0	840878922
451	5	2	3.0	851527569
517	6	1	5.0	858275452
518	6	3	3.0	858275558
519	6	7	5.0	858275558
541	7	3	3.0	1011208463
542	7	7	3.0	1011208220
817	8	1	4.0	833981871
818	8	3	5.0	833981733
819	8	6	3.0	833982631
820	8	10	4.0	833981834
922	10	1	4.0	943497887

고객이 아이템에게
어떠한 액션을 취했는지에 대한 정보

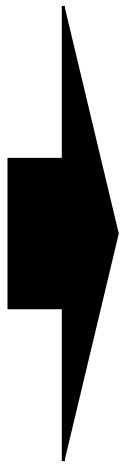


이러한 데이터의 포맷에서는
고객과 아이템 간 특징들을 파악하기 어려움

USER-ITEM Matrix

	user_id	movie_id	rating	rated_at
0	1	2	3.5	1112486027
175	2	3	4.0	974820889
236	3	1	4.0	944919407
423	4	6	3.0	840879227
424	4	10	4.0	840878922
451	5	2	3.0	851527569
517	6	1	5.0	858275452
518	6	3	3.0	858275558
519	6	7	5.0	858275558
541	7	3	3.0	1011208463
542	7	7	3.0	1011208220
817	8	1	4.0	833981871
818	8	3	5.0	833981733
819	8	6	3.0	833982631
820	8	10	4.0	833981834
922	10	1	4.0	943497887

PIVOT



movie_id	1	2	3	6	7	10
user_id						
1	NaN	3.5	NaN	NaN	NaN	NaN
2	NaN	NaN	4.0	NaN	NaN	NaN
3	4.0	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	3.0	NaN	4.0
5	NaN	3.0	NaN	NaN	NaN	NaN
6	5.0	NaN	3.0	NaN	5.0	NaN
7	NaN	NaN	3.0	NaN	3.0	NaN
8	4.0	NaN	5.0	3.0	NaN	4.0
10	4.0	NaN	NaN	NaN	NaN	NaN

행(user)과 열(item)으로 정렬한 행렬

PAI
PUBLIC AI

USER-ITEM Matrix

	user_id	movie_id	rating	rated_at
0	1	2	3.5	1112486027
175	2	3	4.0	974820889
236	3	1	4.0	944919407
423	4	6	3.0	840879227
424	4	10	4.0	840878922
451	5	2	3.0	851527569
517	6	1	5.0	858275452
518	6	3	3.0	858275558
519	6	7	5.0	858275558
541	7	3	3.0	1011208463
542	7	7	3.0	1011208220
817	8	1	4.0	833981871
818	8	3	5.0	833981733
819	8	6	3.0	833982631
820	8	10	4.0	833981834
922	10	1	4.0	943497887

유사도

movie_id / user_id	1	2	3	6	7	10
1	NaN	3.5	NaN	NaN	NaN	NaN
2	NaN	NaN	4.0	NaN	NaN	NaN
3	4.0	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	3.0	NaN	4.0
5	NaN	3.0	NaN	NaN	NaN	NaN
6	5.0	NaN	3.0	NaN	5.0	NaN
7	NaN	NaN	3.0	NaN	3.0	NaN
8	4.0	NaN	5.0	3.0	NaN	4.0
10	4.0	NaN	NaN	NaN	NaN	NaN

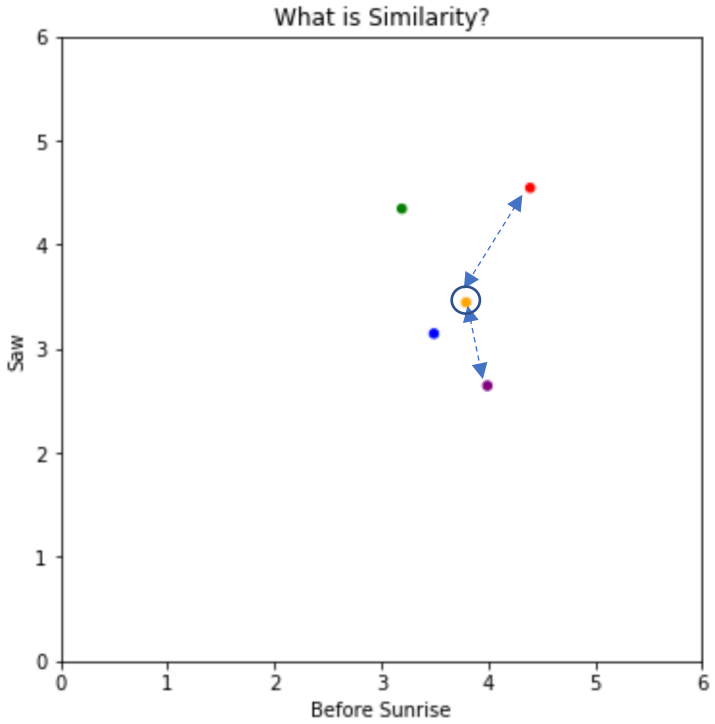
유사도를 통해 우리는 어떤 영화끼리 비슷한지를 알 수 있음

PAI PUBLIC AI

협업 필터링 연산의 핵심 : 유사도

핵심 아이디어

> 만약 두 명의 사용자가 유사한 관심사를 가지고 있다면,
> 그들은 미래에도 유사한 취향을 가질 것이다.



유사도 계산

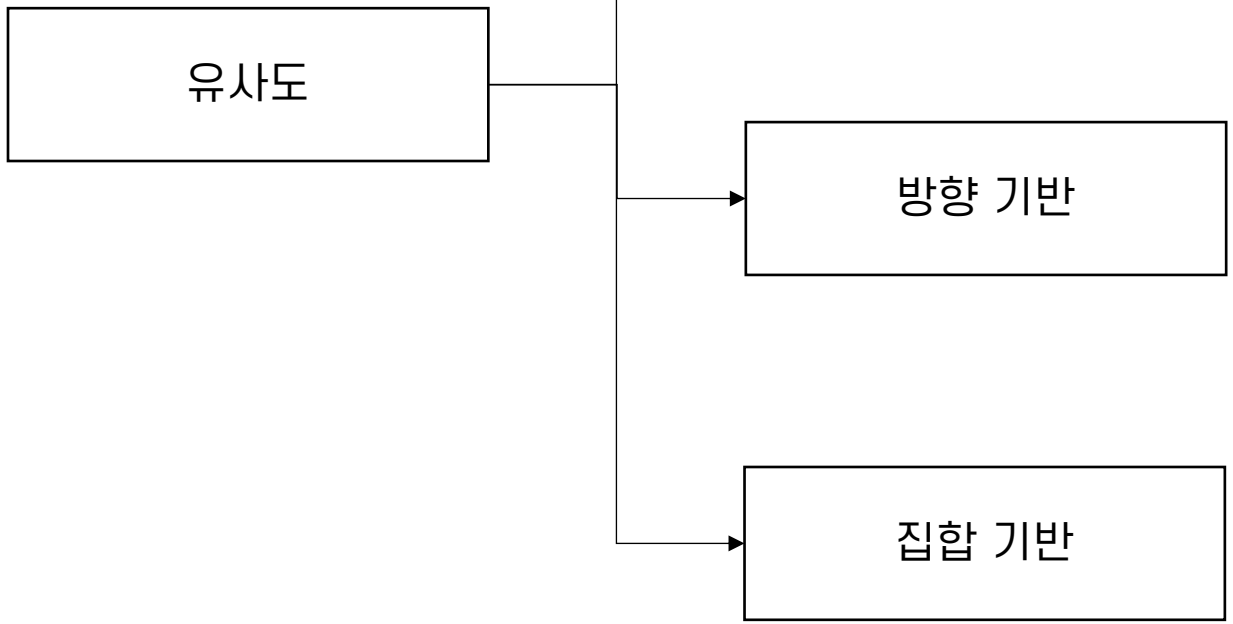
노란색 유저는

* 빨간색 유저와 유사할까?

* 보라색 유저와 유사할까?

PAI
PUBLIC AI

유사도의 다양한 기준들



$$sim(X, Y) = \frac{1}{1 + dist(X, Y)}$$

맨해튼 유사도 $$dist(X, Y) = \sum_{i=1}^{n}(|x_i - y_i|)$$

유클리디안 유사도 $$dist(X, Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

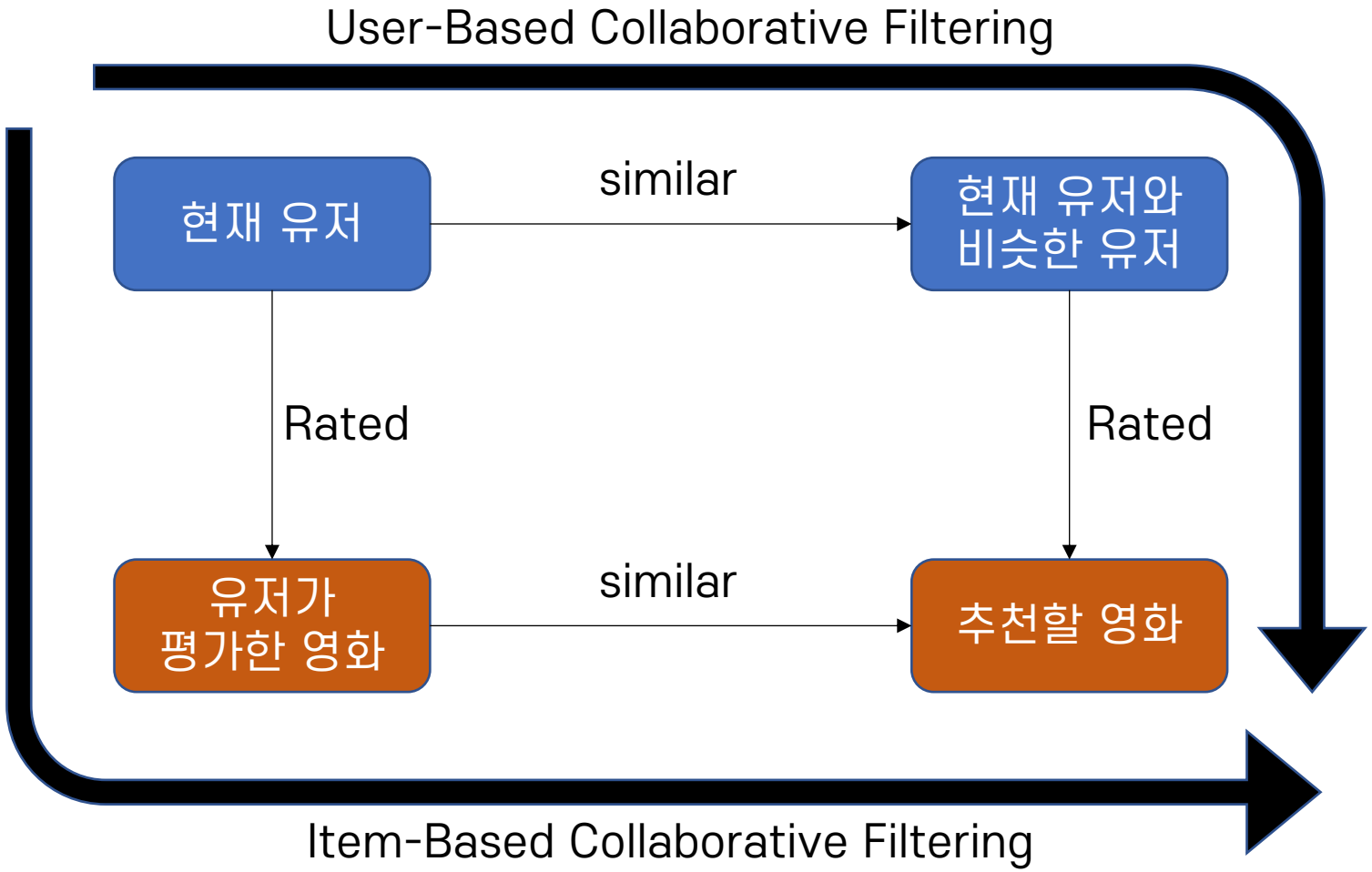
코사인 유사도 $$sim(X, Y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

자카드 유사도 $$sim(X, Y) = \frac{(X \cap Y)}{(X \cup Y)}$$

PAI
PUBLIC AI

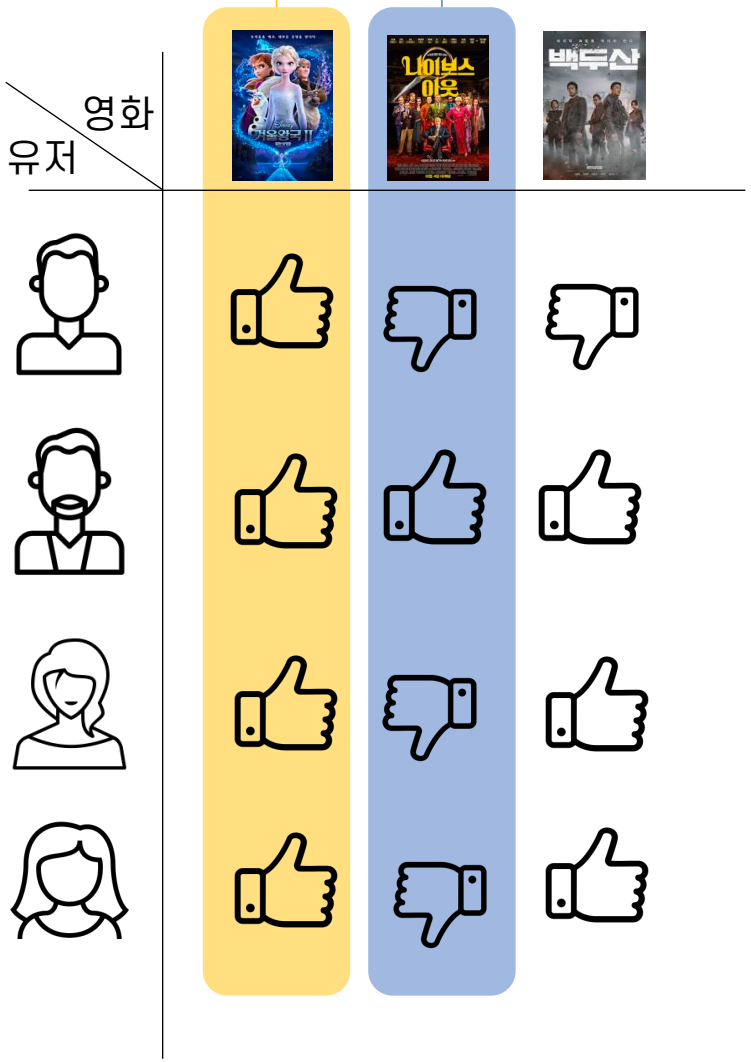
Memory-Based 협업 필터링의 종류

Memory-Based 협업 필터링은 크게
User-Based Collaborative Filtering과 Item-Based Collaborative Filtering으로 나뉘어짐

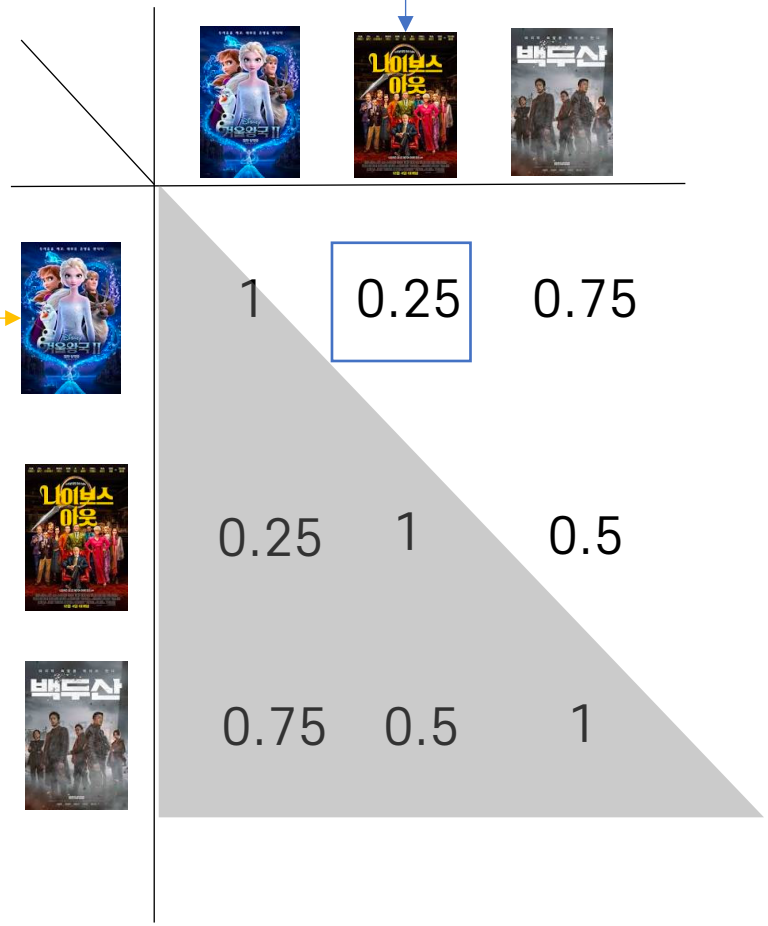


Item-Based Collabortive Filtering

User-Item Matrix



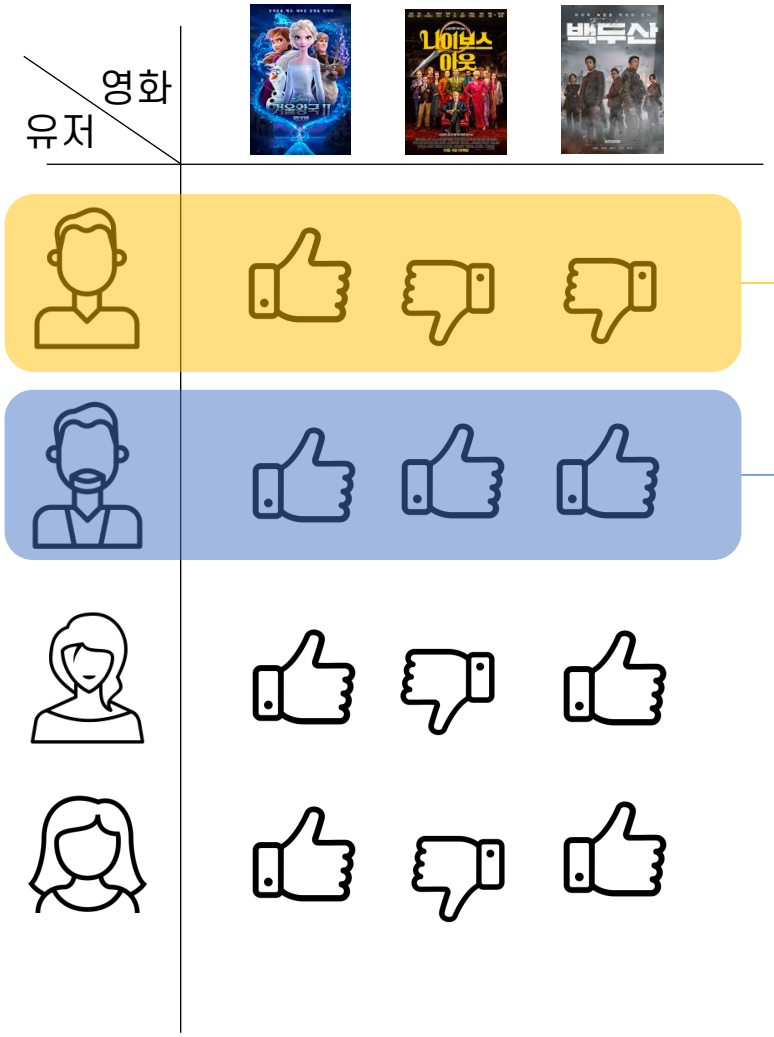
Item Similarity Matrix



PAI
PUBLIC AI

User-Based Collabortive Filtering

User-Item Matrix



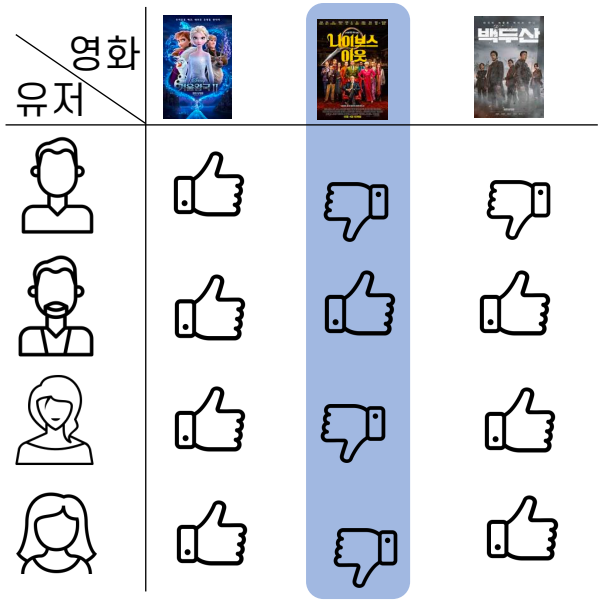
User Similarity Matrix

PAI
PUBLIC AI

Item-User Matrix

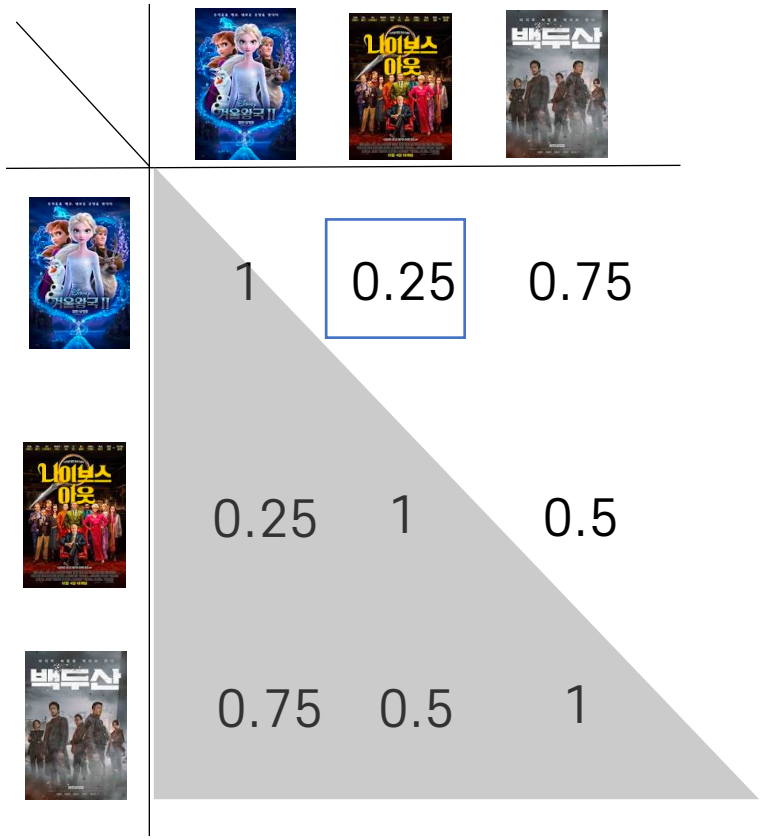
행렬 곱

User-Item Matrix



=

Similarity Matrix



PAI
PUBLIC AI

Item Based CF 활용 예제) Amazon.com 추천 시스템

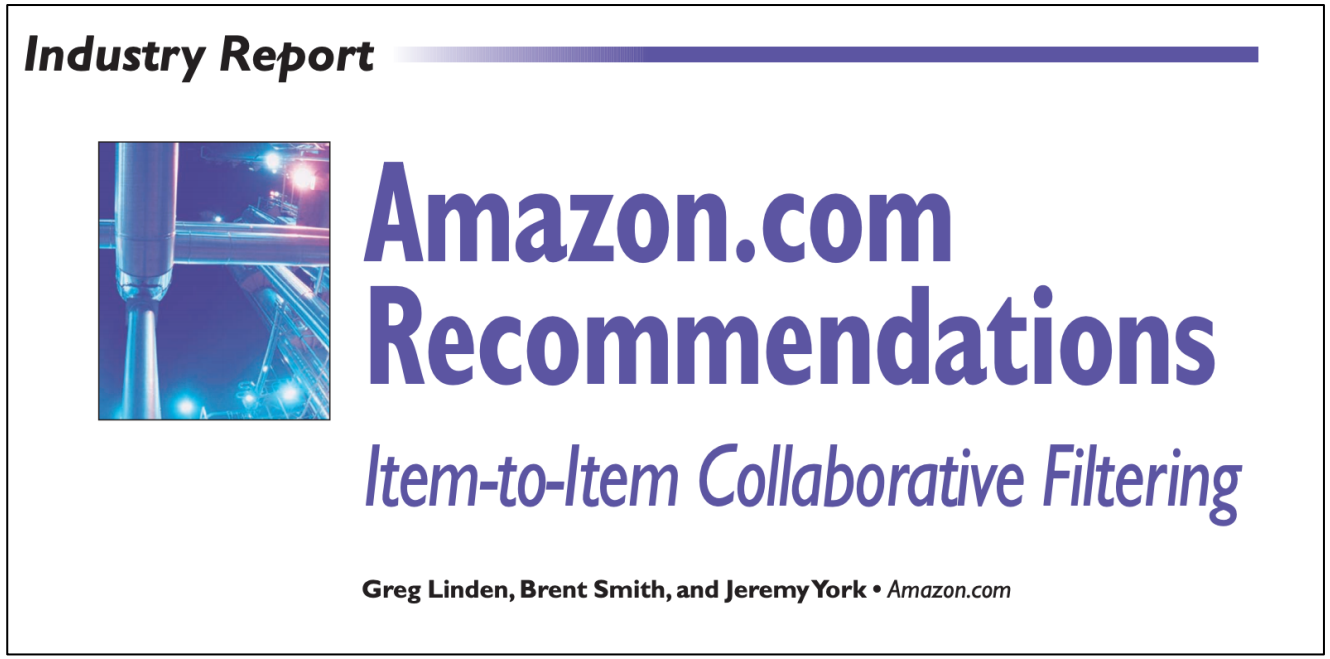


Figure 1. The "Your Recommendations" feature on the Amazon.com homepage. Using this feature, customers can sort recommendations and add their own product ratings.

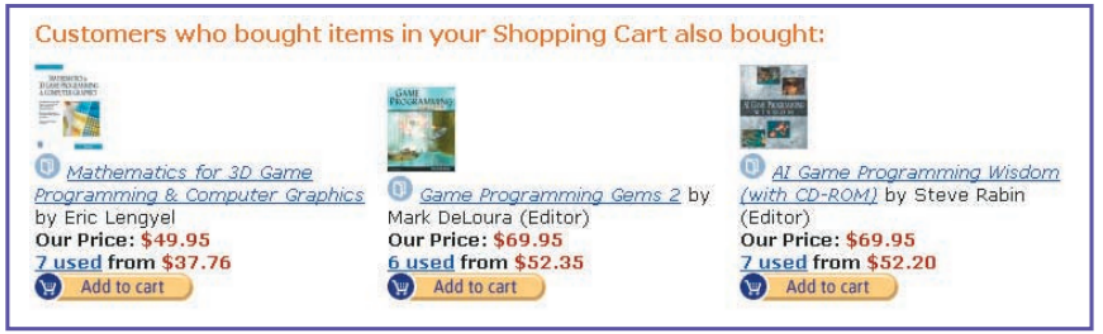


Figure 2. Amazon.com shopping cart recommendations. The recommendations are based on the items in the customer's cart: The Pragmatic Programmer *and* Physics for Game Developers.

Model based Collaborative Filtering
<Matrix Factorization>

모델 기반 협업 필터링

모델 기반 협업 필터링 핵심 아이디어

User – Item Matrix를 인수분해(Factorization)하자

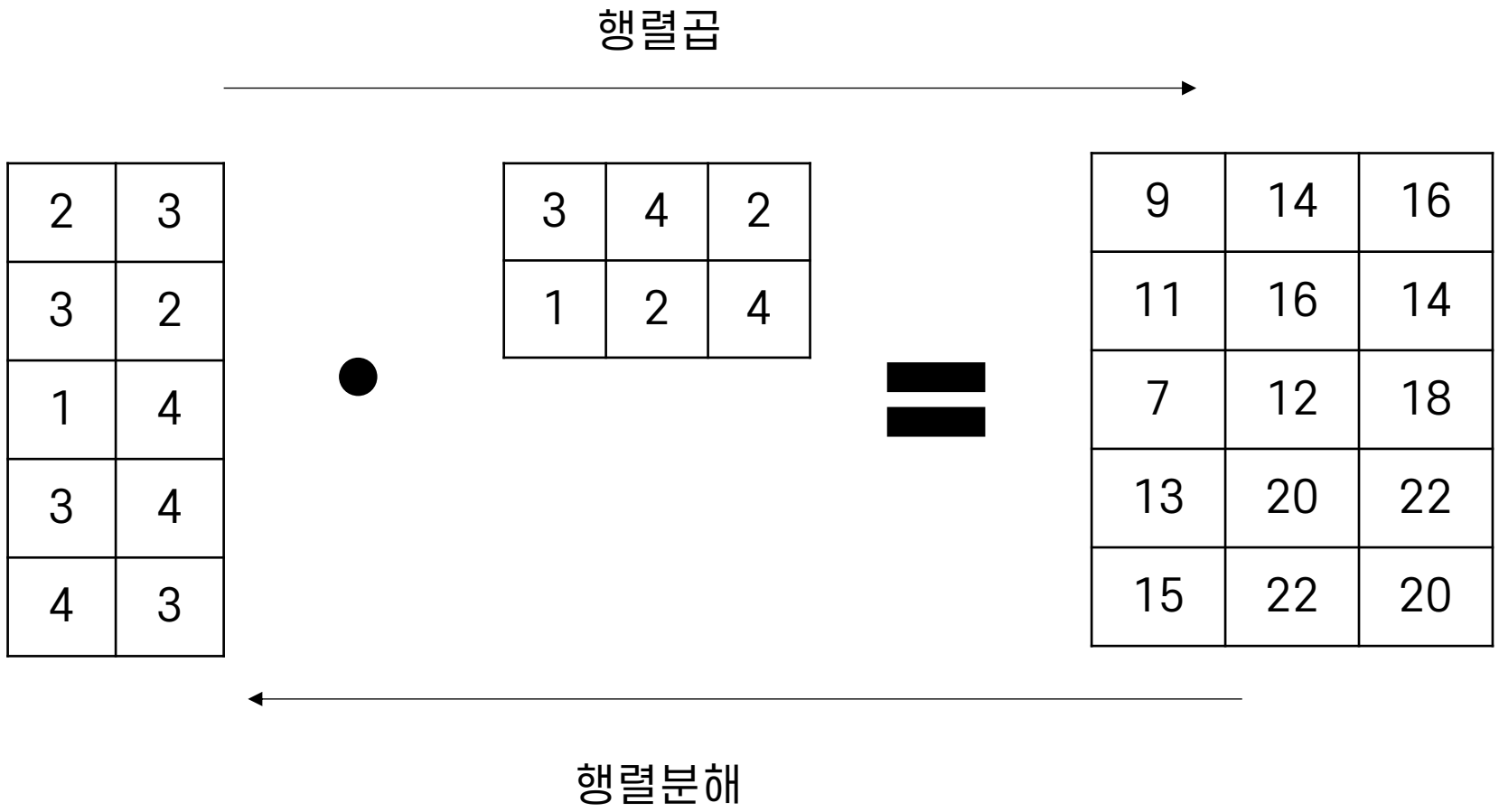
유저 \ 음악				
		4.5	2.0	
	4.0		3.5	
		5.0		2.0
		3.5	4.0	1.0

이 행렬을 크게 두가지 인수, 고객에 대한 행렬과 아이템에 대한 행렬로 분해하자

PAI
PUBLIC AI

행렬분해 (Matrix Decomposition)

행렬분해는 행렬곱의 반대로, 하나의 행렬을 두 개의 인수행렬로 나누는 것



오늘의 주제 : 모델 기반 협업 필터링

우리의 목표 :

유저-아이템 행렬을 유저의 행렬과 아이템의 행렬로 분해하는 것

User-Artist 행렬

유저 \ 음악				
		4.5	2.0	
	4.0		3.5	
		5.0		2.0
		3.5	4.0	1.0

≈

$Embedding_{user}$

유저의 취향 행렬




?	?
?	?
?	?
?	?

$Embedding_{artist}$

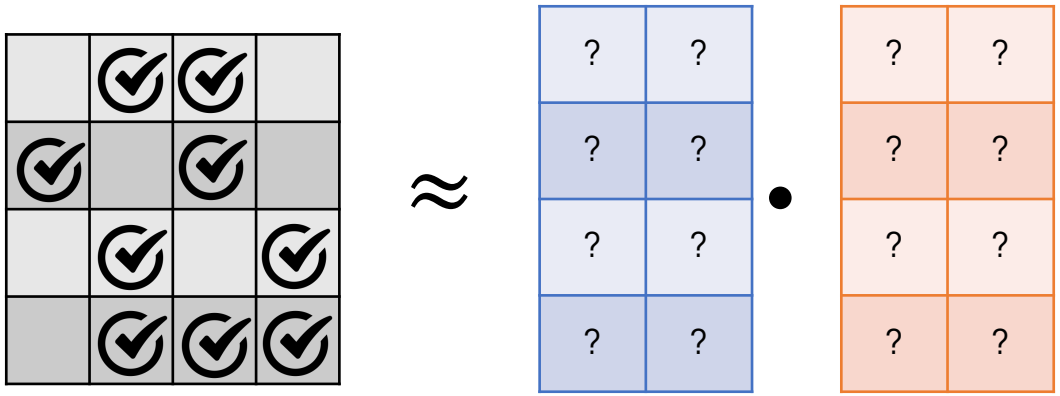
아티스트의 특성 행렬




?	?
?	?
?	?
?	?

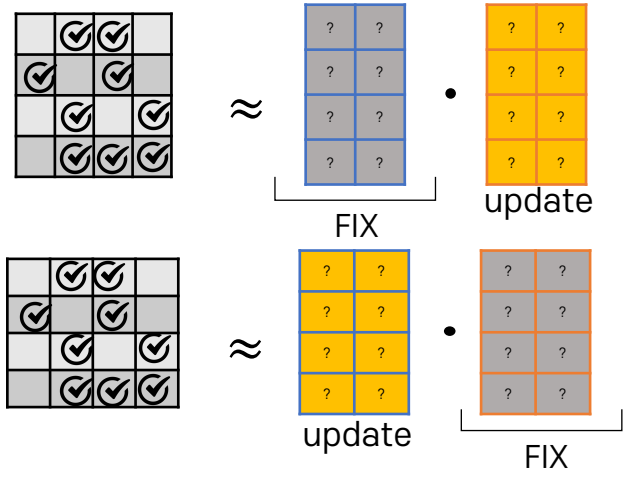
PAI
PUBLIC AI

Matrix Factorization (For 암묵 데이터)



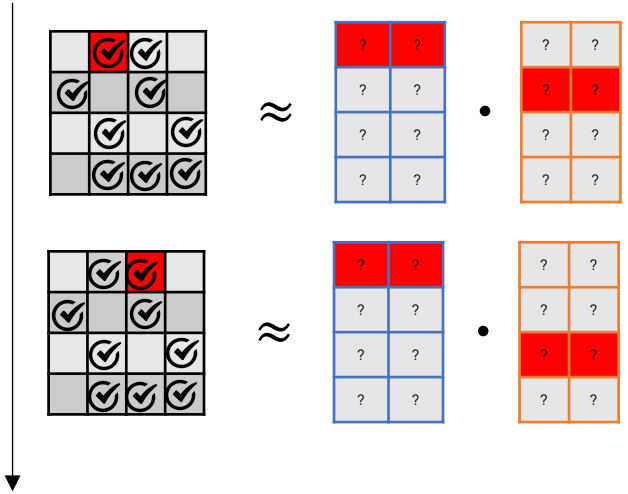
ALS(Alternating Least Square)

전체 데이터로
교대로 행렬을 업데이트하기



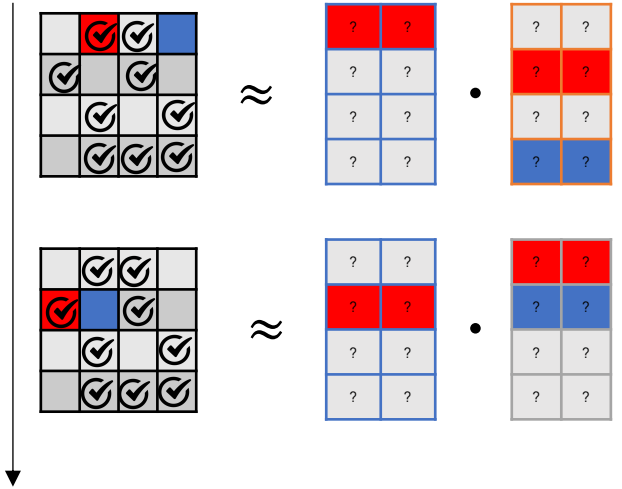
SGD(Stochastic Gradient Descent)

각 케이스 별로
행렬 값을 업데이트하기



BPR(Bayesian Personalized Ranking

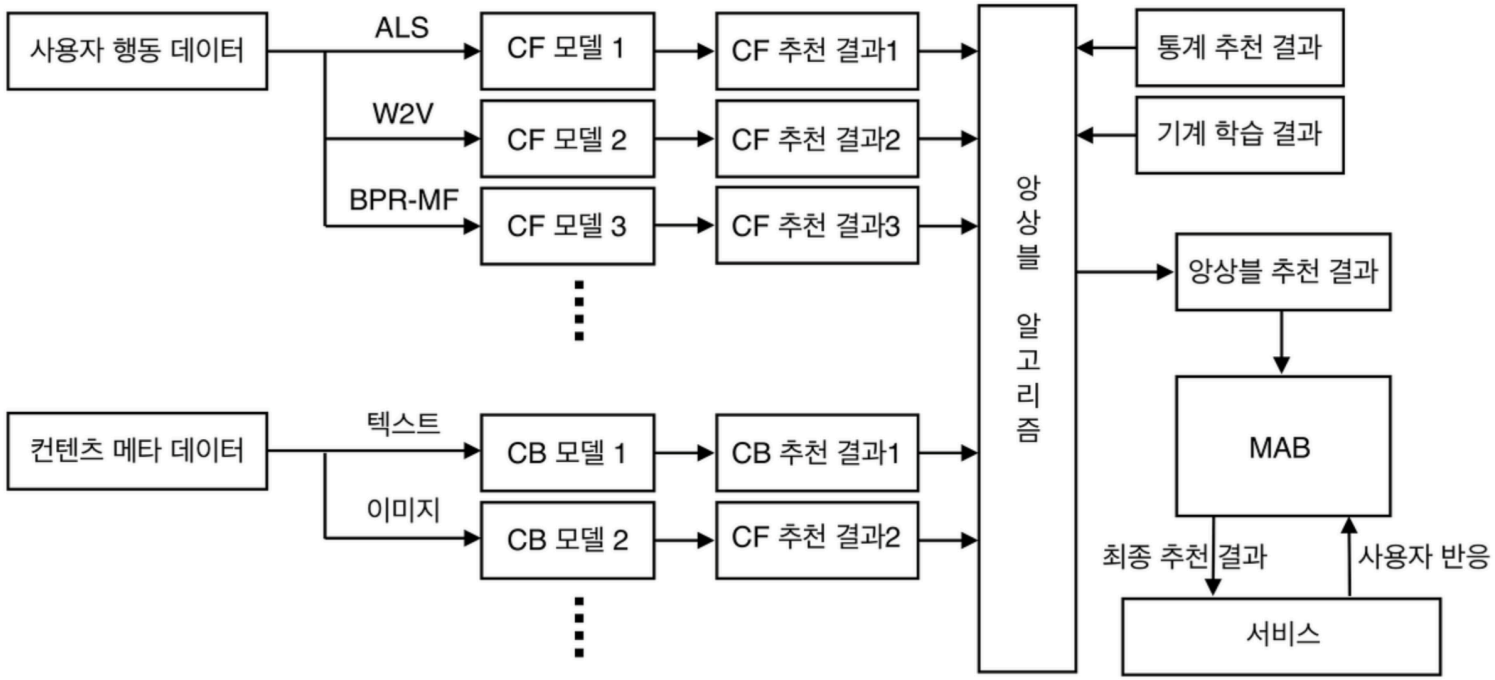
케이스 쌍 별로
행렬 값을 업데이트하기



PAI
PUBLIC AI

BPR 활용 예제) 카카오 추천 시스템, 토로스

토로스는 CF, CB, 통계 모델, 일반적인 기계학습 모델 등 다양한 모델들에서 추천 결과를 뽑고 뽑은 추천 결과를 앙상블하여 하나의 추천 결과로 병합한다. 만들어진 추천 결과가 사용자들에게 노출되기 시작하면 MAB를 사용해 가장 좋은 추천 결과가 무엇일지 찾아낸다.



[그림 4] 토로스 추천 개요

https://brunch.co.kr/@kakao-it/72

PAI PUBLIC AI

Deep Learning Basis

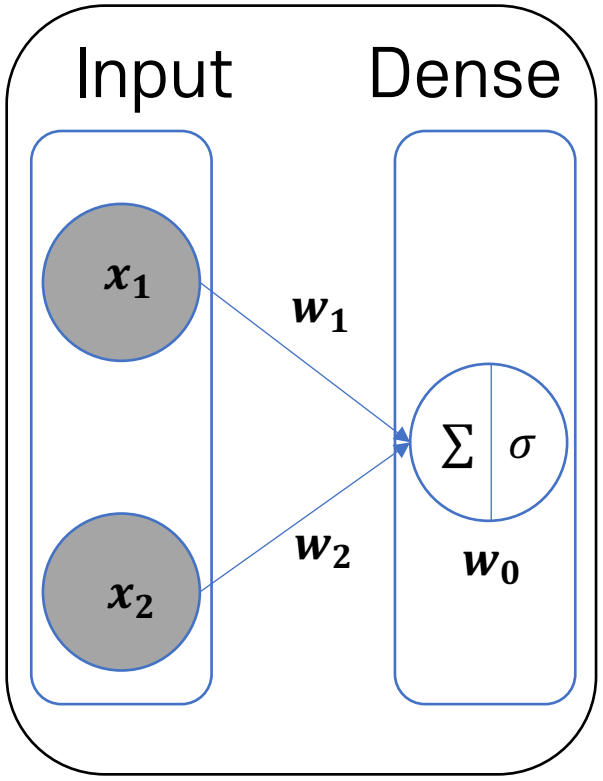
PAI
PUBLIC AI

Keras로 모델을 구성하기 (1) 뉴런 하나 구성하기

수식

$$z = w_1 x_1 + w_2 x_2 + w_0$$
$$y = relu(z)$$

층 구조도



코드

```
# (1) 입력층의 형태 결정하기
inputs = Input(2, name='x')

# (2) 출력층의 형태 결정하기
dense_layer = Dense(1, activation='relu', name='output')

# (3) 레이어를 연결하기
output = dense_layer(inputs)

# (4) 모델 구성하기
model = Model(inputs, output, name='model')
```

PAI
PUBLIC AI

Keras로 모델을 구성하기 (2) 인공신경망 구성하기

수식

$$z^{[1]} = XW^{[1]} + b^{[1]}$$
$$a^{[1]} = relu(z^{[1]})$$
$$y = a^{[1]}W^{[2]} + b^{[2]}$$

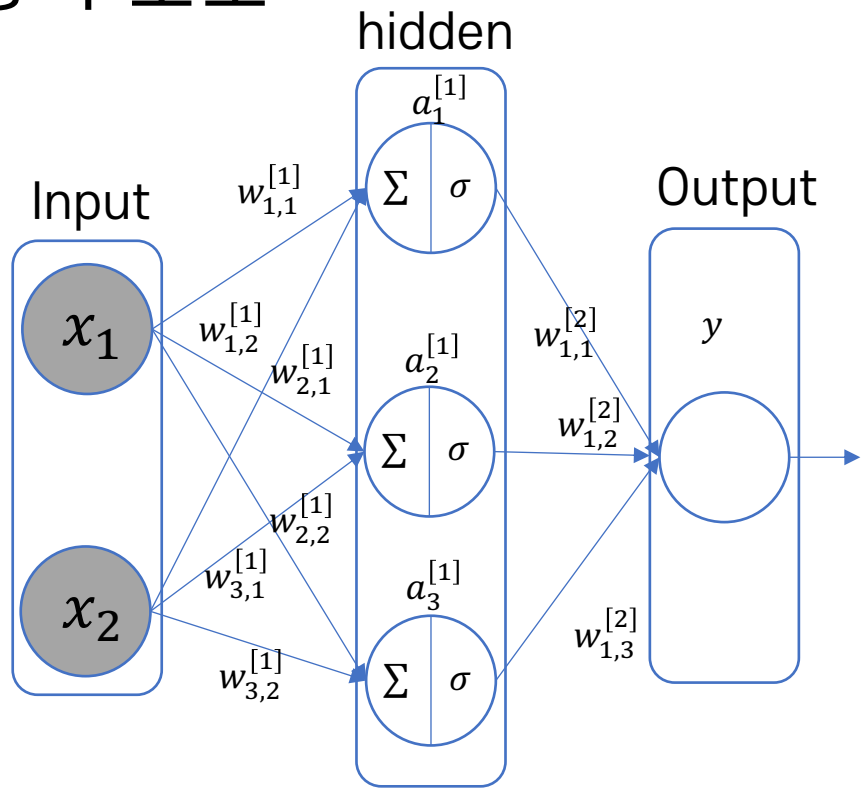
행렬 크기

$X : (None, 2)$

$W^{[1]} : (2, 3)$ $b^{[1]} : (3,)$

$W^{[2]} : (3, 1)$ $b^{[2]} : (1,)$

층 구조도



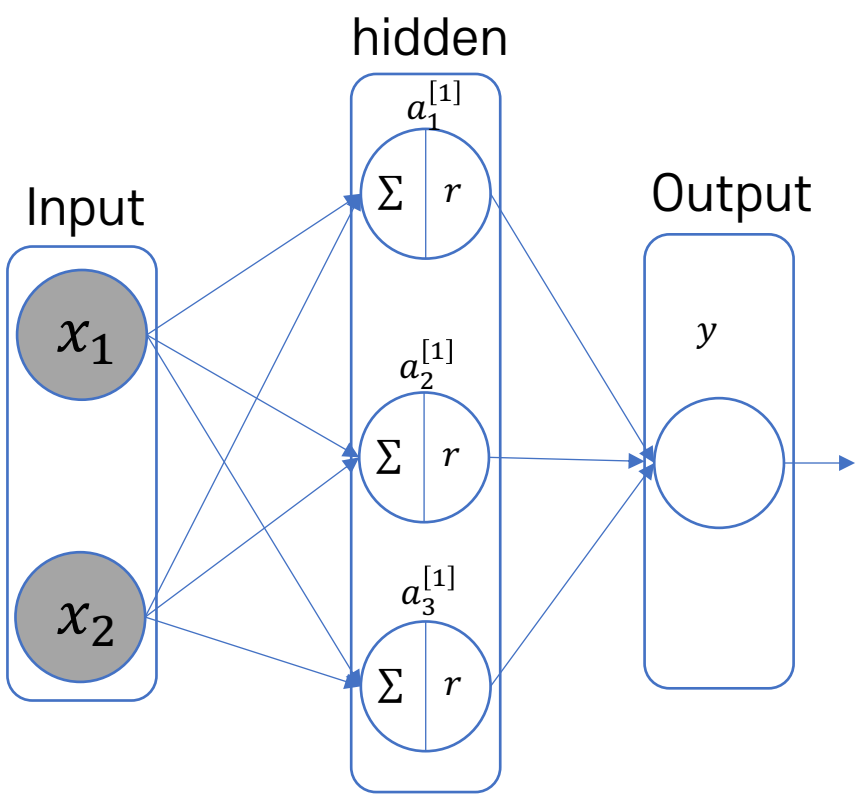
코드

```
# (1) 입력층의 형태 결정하기
inputs = Input(2, name='x')

# (2) 은닉층 구성하기
hidden = Dense(3, activation='relu',
               name='hidden')(inputs)

# (3) 출력층 구성하기
output = Dense(1, name='output')(inputs)

# (4) 모델 구성하기
model = Model(inputs, output, name='model')
```

PAI PUBLIC AI

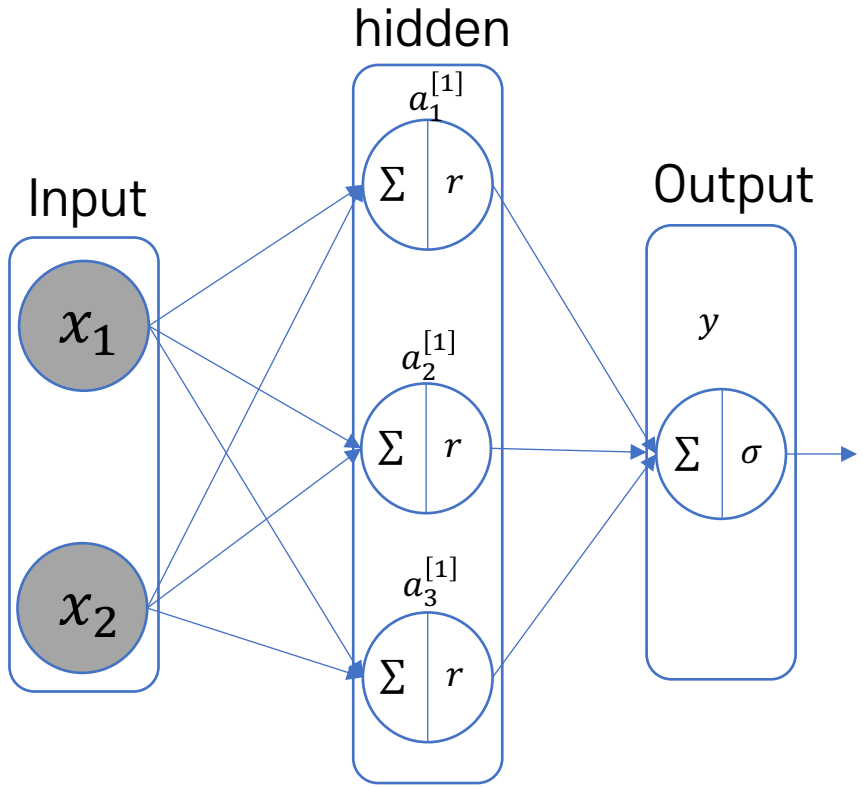
Keras로 모델 컴파일하기

Case 1) Regression 모델인 경우




```
from tensorflow.keras.losses import MeanSquaredError
from tensorflow.keras.optimizers import SGD

model.compile(optimizer=SGD(.1),
              loss=MeanSquaredError())
```

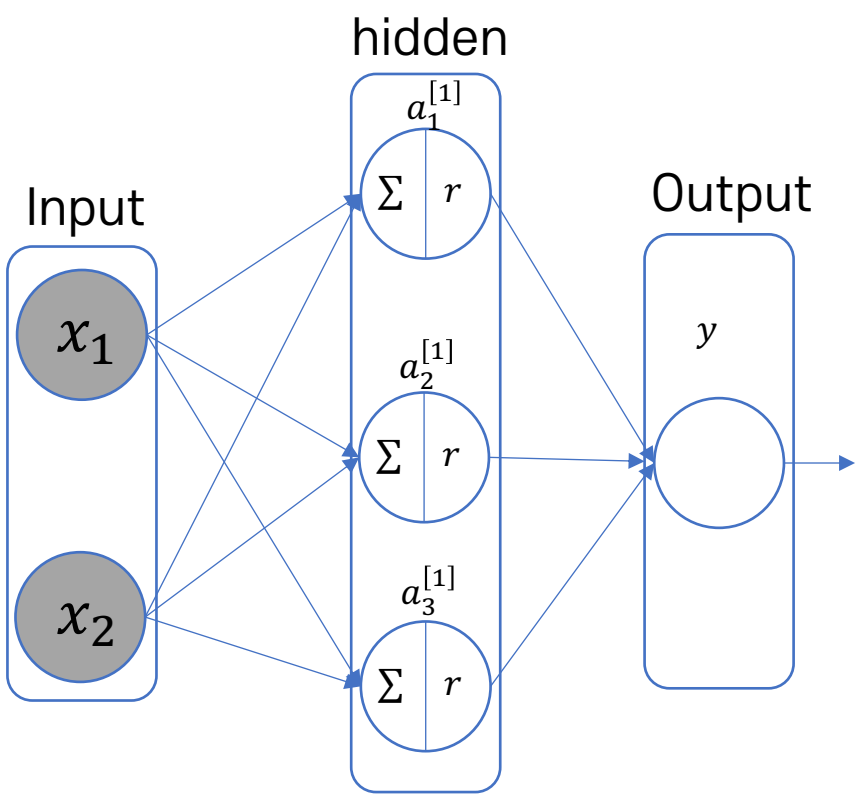
Case 2) Classification 모델인 경우




```
from tensorflow.keras.losses import BinaryCrossentropy
from tensorflow.keras.optimizers import SGD

model.compile(optimizer=SGD(.1),
              loss=BinaryCrossentropy())
```

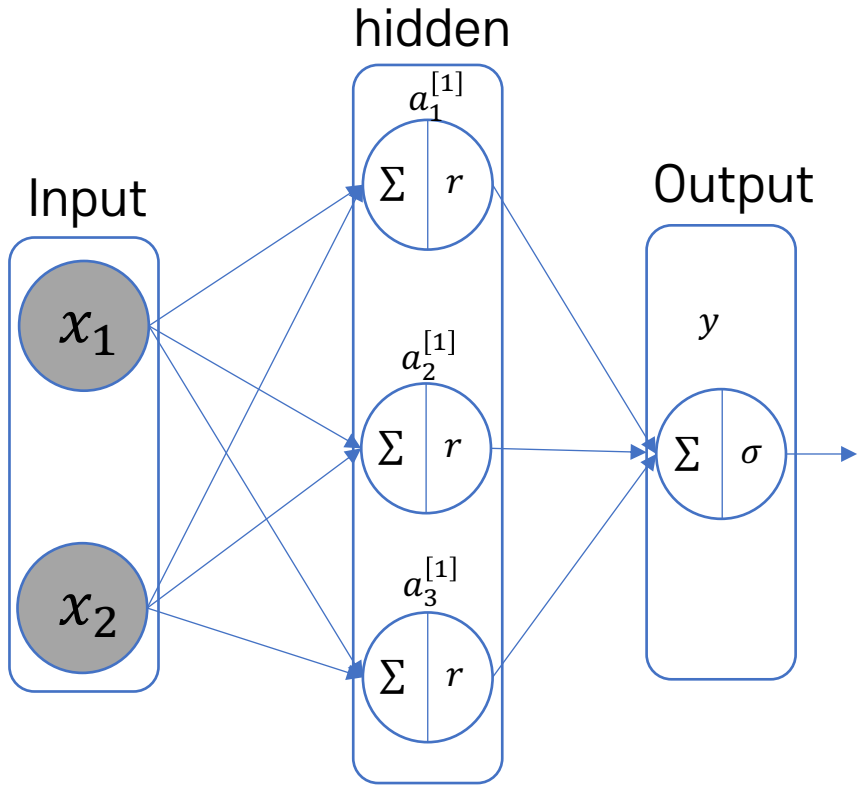
Keras로 모델 컴파일하기

Case 1) Regression 모델인 경우




```
from tensorflow.keras.losses import MeanSquaredError
from tensorflow.keras.optimizers import SGD

model.compile(optimizer=SGD(.1),
              loss=MeanSquaredError())
```

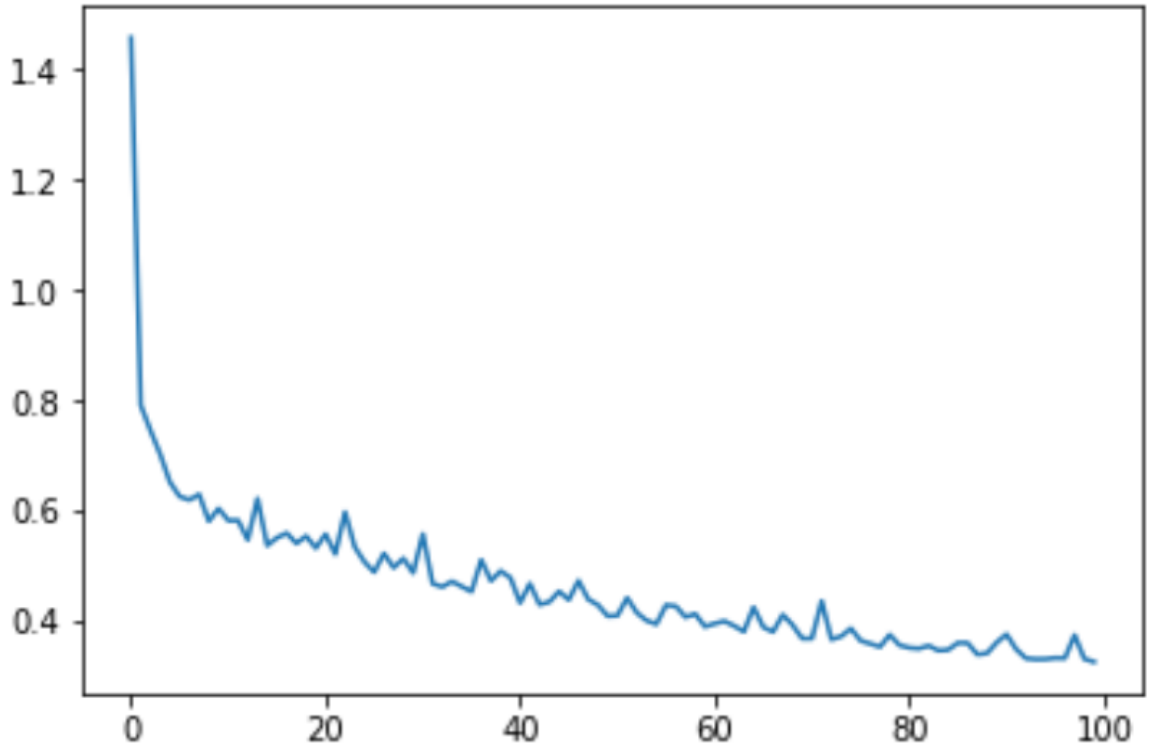
Case 2) Classification 모델인 경우




```
from tensorflow.keras.losses import BinaryCrossentropy
from tensorflow.keras.optimizers import SGD

model.compile(optimizer=SGD(.1),
              loss=BinaryCrossentropy())
```

Keras로 모델 학습하기

```
model.fit(X, y, batch_size=64, epochs=100, verbose=0)

plt.plot(model.history.history['loss'])
plt.show()
```



PAI
PUBLIC AI

Neural Collaborative Filtering

Neural Collaborative Filtering

딥러닝을 활용하여 Matrix Factorization을 시도한 대표적인 모델

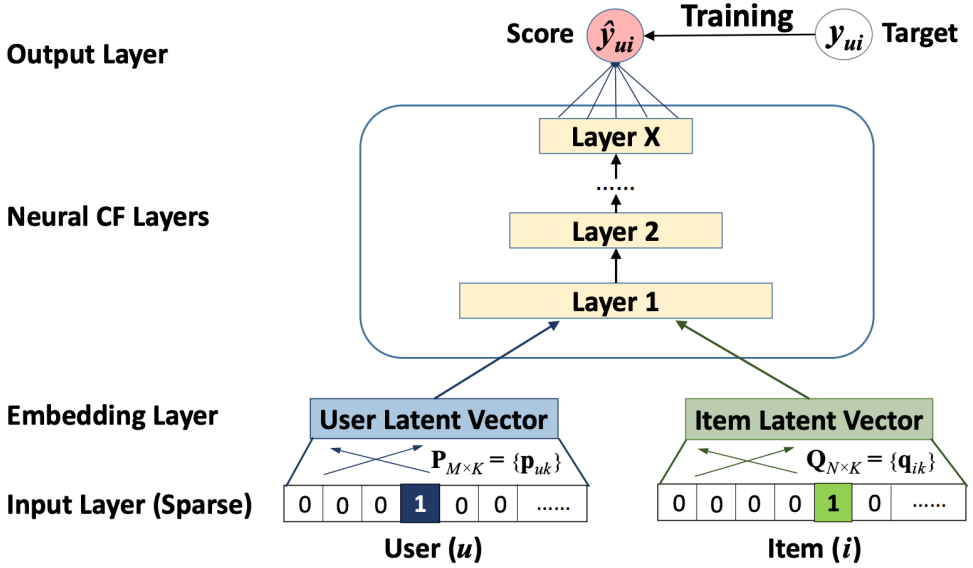


Figure 2: Neural collaborative filtering framework

논문 초록

In recent years, deep neural networks have yielded immense success on speech recognition, computer vision and natural language processing. However, the exploration of deep neural networks on recommender systems has received relatively less scrutiny. In this work, we strive to develop techniques based on neural networks to tackle the key problem in recommendation — collaborative filtering — on the basis of implicit feedback.

Although some recent work has employed deep learning for recommendation, they primarily used it to model auxiliary information, such as textual descriptions of items and acoustic features of musics. When it comes to model the key factor in collaborative filtering — the interaction between user and item features, they still resorted to matrix factorization and applied an inner product on the latent features of users and items.

PHI PUBLIC AI

Neural Collaborative Filtering

Neural Collaborative Filtering의 입력값과 출력값

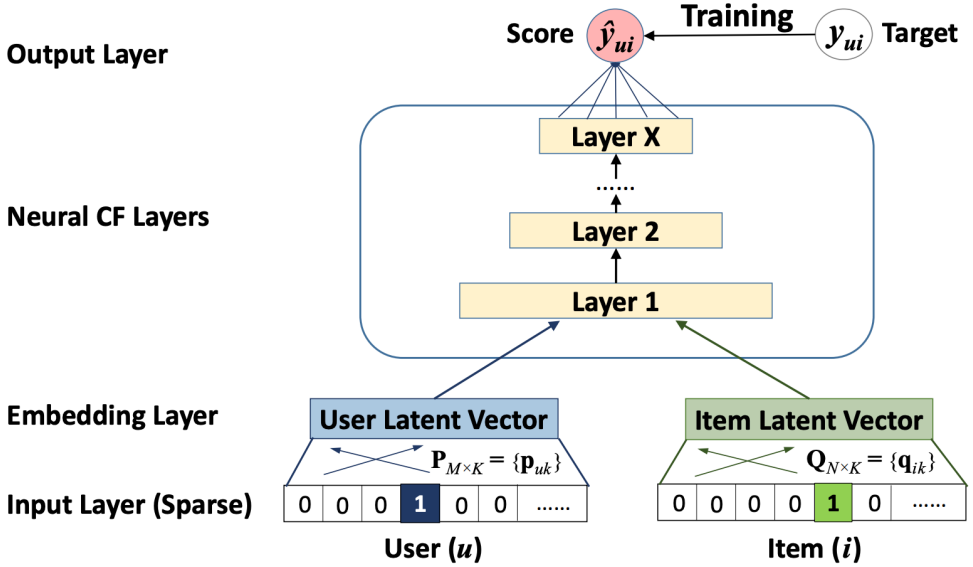
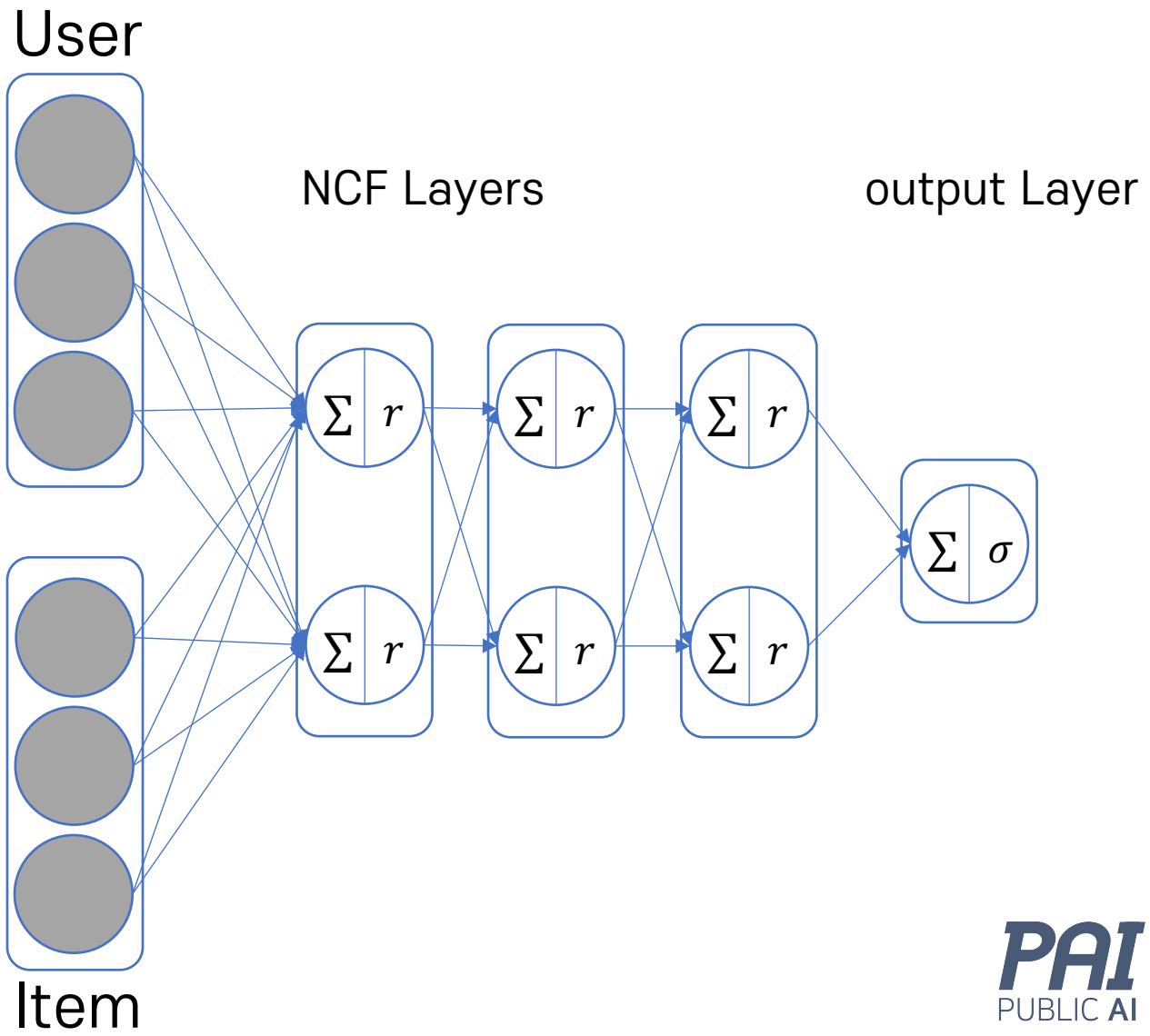


Figure 2: Neural collaborative filtering framework



입력값

- 유저
- 아이템

출력값

- 해당 유저가 아이템을 선호할 확률

PAI
PUBLIC AI

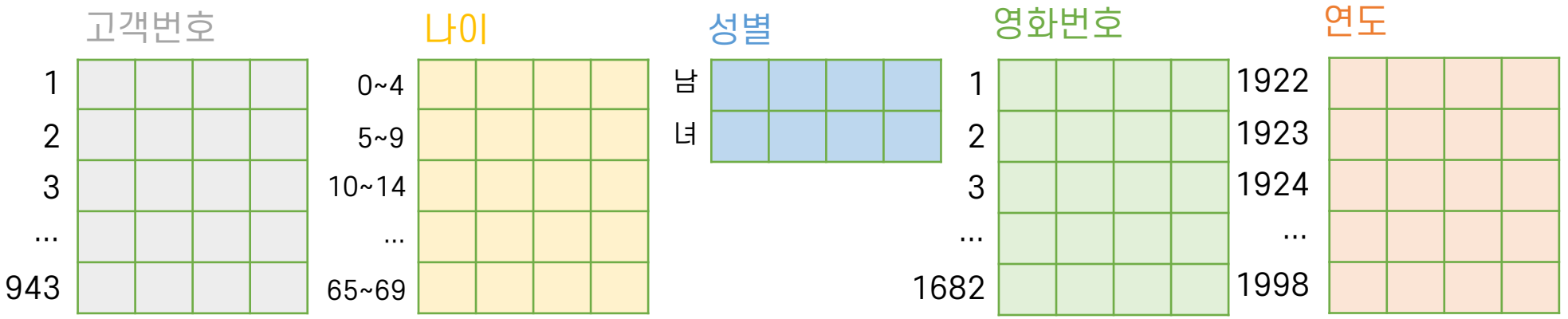
Factorization Machine

Factorization Machine의 아이디어

Factorization Machine의 수식

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

상호작용 임베딩 행렬



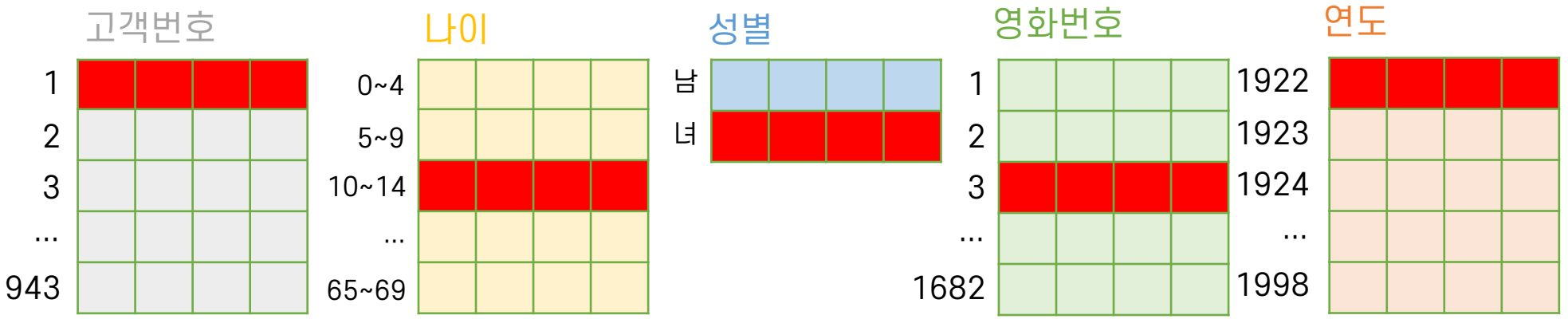
PAI
PUBLIC AI

Factorization Machine의 아이디어

Factorization Machine의 수식

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j$$

상호작용 임베딩 행렬



Q) 나이가 13세인 여성 1번 고객이 1922년에 출시한 액션 영화 3번에 대한 상호작용 효과?

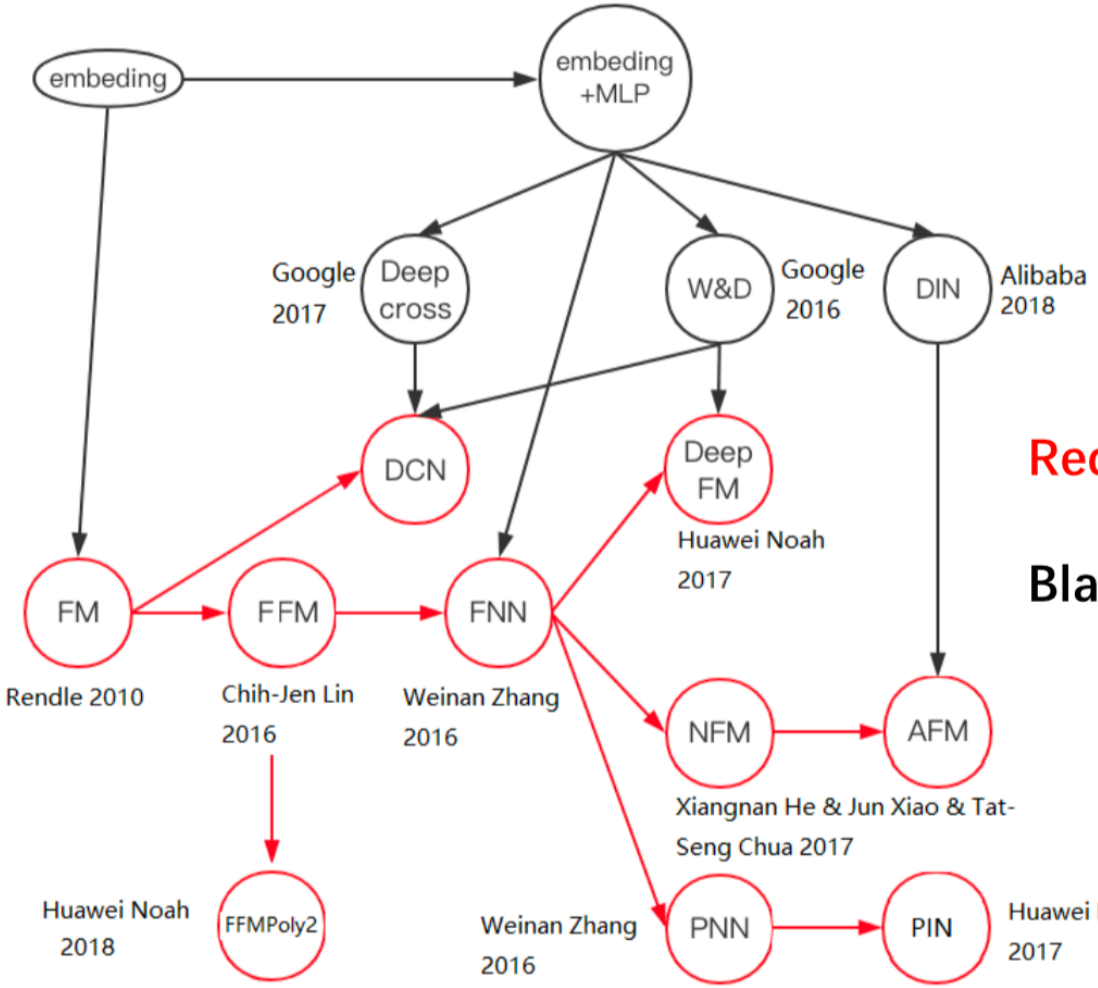
$< \text{고객}_1, \text{나이}_{10\sim14\text{세}} > + < \text{고객}_1, \text{성별}_{\text{여}} > + < \text{고객}_1, \text{영화}_3 > + < \text{고객}_1, \text{연도}_{1992} > + \cdots$.

=> 각 임베딩 벡터끼리의 DOT 연산의 합

PAI PUBLIC AI

딥러닝을 활용한 추천시스템 발전 과정

Evolution of deep learning for recommender system



Red path：FM path

Black path：embedding + MLP path

PAI PUBLIC AI