

## 추천 시스템

# 추천 시스템의 중요성

- 아마존 등과 같은 전자상거래 업체부터 넷플릭스, 유튜브 등 콘텐츠 포털까지 추천 시스템을 통해 사용자의 취향을 이해하고, 맞춤 상품과 콘텐츠를 제공하고 있다.
- 이에 따라 조금이라도 오랫동안 서비스에 고객을 오래 머무르게 하기 위해 전력을 기울이고 있다.

# 추천 엔진의 필요성

- 아마존 및 넷플릭스 같은 상품 콘텐츠 서비스를 제공하는 측에서는 너무나 많은 상품이 존재한다.
- 이에 따라 사용자는 어떤 상품을 골라야 할지 선택의 압박이 생기게 된다.
- 추천 엔진은 사용자가 무엇을 원하는지 빠르게 찾아내어 사용자의 온라인 쇼핑 및 콘텐츠 선택에 있어 즐거움이 생기게 된다.

## 추천 시스템 방식

# 추천 시스템 방식

- 추천 시스템은 크게 콘텐츠 기반 필터링(Content Based Filtering), 협업 필터링(Collaborative Filtering) 두 가지 방식으로 나뉘게 된다.
  - 콘텐츠 기반 필터링은 상품의 속성에 개인의 취향을 반영하여 사용자에게 콘텐츠가 가지고 있는 여러 요소이 사용자에게 어울리겠구나 라는 것을 판별해가 제공한다. 예를 들어 어떤 사용자가 특정 영화 감독의 영화를 좋아한다면 이 영화 감독이 참여한 영화를 중점적으로 추천하는 경우를 들 수 있다.
  - 협업 필터링은 나와 비슷한 사람이 선택한 상품을 선택한다는 개념으로가 예를 들어 A라는 사람과 B라는 사람의 쇼핑 목록이 비슷할 때 A가 사고, B가 사지 않은 물건을 B에게 추천해 주는 경우를 들 수 있다.
- 추천 시스템은 이들 방식 중 1가지를 선택하거나 이들을 결합하여 하이브리드 방식으로 사용한다.

## 컨텐츠 기반 필터링

# 콘텐츠 기반 필터링 ( Contents Based Filtering )



# 콘텐츠 기반 필터링 ( Contents Based Filtering )



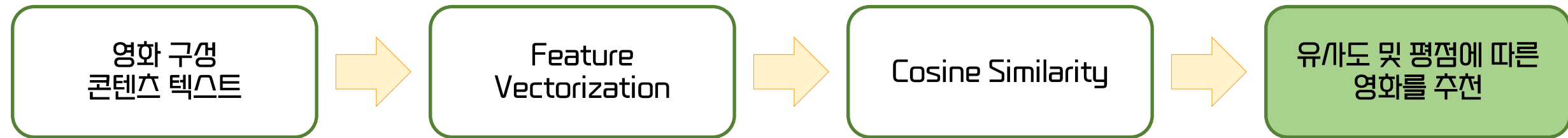


# 콘텐츠 기반 필터링 ( Contents Based Filtering )

- 콘텐츠 기반 필터링을 수행하기 위해 반드시 필수 형식의 데이터만 필요하지는 않음
- 예를 들어 영화의 경우 감독, 배우, 영화 설명, 장르 등 영화를 구성하는 다양한 콘텐츠들을 텍스트 기반 문서 유사도로 비교하여 추천할 수도 있음

# 콘텐츠 기반 필터링 ( Contents Based Filtering )

- 콘텐츠 기반 필터링을 수행하기 위해 반드시 필수 형식의 데이터만 필요하지는 않음
- 예를 들어 영화의 경우 감독, 배우, 영화 설명, 장르 등 영화를 구성하는 다양한 콘텐츠들을 텍스트 기반 문서 유사도로 비교하여 추천할 수도 있음



# 콘텐츠 기반 필터링 실습 (TMDB 5000)

- 미국의 유명한 영화 평점 사이트인 IMDB 데이터 세트를 캐글에서 다운 받기
  - <https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata/code>
- 콘텐츠 기반 필터링은 다음과 같은 프로세스에 의해 구현함
  1. 콘텐츠에 대한 여러 텍스트 정보들을 피처 벡터화
  2. 코사인 유사도로 콘텐츠별 유사도 계산
  3. 콘텐츠 별로 가중 평점을 계산
  4. 유사도가 높은 콘텐츠 중에 평점이 좋은 콘텐츠 순으로 추천

## 협업 필터링

# 협업 필터링 ( Collaborative Filtering )

- 이번에 개봉하는 영화를 볼까 말까 고민할 때 그 영화를 봤었던 친구에게 물어보고자 한다.
  - 이 때 주의 할 점은 취향이 비슷한 친구에게 물어보아야 한다는 것!

# 협업 필터링의 유형

- 최근접 이웃 기반(Nearest Neighbor)
  - 사용자 기반(User-user CF)
    - 만약 내가 사용자 A라면 나와 취향이 비슷한 다른 사용자 들을 찾는 것
  - 아이템 기반(Item-item CF)
    - 내가 선호하는 아이템 A와 비슷한 다른 아이템들을 찾는 것
- 잠재 요인 기반(Latent Factor)
  - 행렬 분해 기반(Matrix Factorization)
    - 사용자-아이템 행렬을 분해하여(일반적으로는 SVD) 다시 재결합 후 재결합된 행렬을 이용하여 예측 평점을 추출하는 방식

# 협업 필터링의 특징

- 유저의 행동(User behavior – 상품 구매 이력, 영화 평점 이력 등)에만 기반하여 추천 알고리즘 들을 전반적으로 지칭
- 상품, 영화 등 사용자가 아직 평가하지 않은 상품(item)에 대한 평가(rating)을 예측하는 것이 주요 역할

	item1	item2	item3	...	Item M
User 1	3		3		?
User 2	4	2			3
User 3		1	2		2
User 4	1				
...		3	1		
User N	4	2			5

User1은 item1, 3에 대한 평가 자료만 존재한다. Item M에 대한 평가를 예측 할 수 있을까?

# 협업 필터링의 특징

- 유저의 행동(User behavior – 상품 구매 이력, 영화 평점 이력 등)에만 기반하여 추천 알고리즘 들을 전반적으로 지칭
- 상품, 영화 등 사용자가 아직 평가하지 않은 상품(item)에 대한 평가(rating)을 예측하는 것이 주요 역할

	item1	item2	item3	...	Item M
User 1	3		3		?
User 2	4	2			3
User 3		1	2		2
User 4	1				
...		3	1		
User N	4	2			5

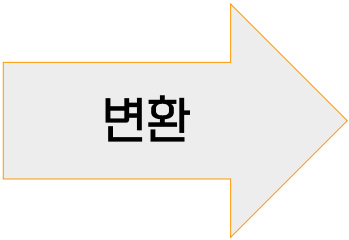
결국은 평가하지 않은 데이터에 대한 예측을 하게 된다. 추천은 평점이 높으면 추천, 평점이 높지 않으면 추천을 해준다.



# 협업 필터링을 위한 데이터 세트 만들기 - 사용자 행/아이템 열

일반적인 사용자-아이템 평점 데이터

User ID	Item ID	Rating
User 1	Item 1	3
User 1	Item 3	3
User 2	Item 1	4
User 2	Item 2	1
User 3	Item 4	5



판다스의 pivot\_table

사용자가 행, 아이템이 열로 구성된  
사용자-아이템 평점 데이터

	item1	item2	item3	Item 4
User 1	3		3	
User 2	4	1		
User 3				5

# 사용자 기반과 아이템 기반 협업 필터링 이해



사용자 기반  
(User – User)

- 특정 사용자와 비슷한 고객들을 기반으로 이 비슷한 고객들이 선호하는 다른 상품을 추천
- 특정 사용자와 비슷한 상품을 구매한 고객들은 비슷한 고객으로 간주

당신과 비슷한 고객들도 다음 상품도 구매했습니다.

	item1	item2	item3	Item 4
User 1	5	4	4	?
User 2	4	4	5	5
User 3	1	1	1	3

User 1이 높은 평점을 준 상품들과 비슷한 User2가 5점을 부여한 Item4를 User 1에게 추천할 수 있다.

# 사용자 기반과 아이템 기반 협업 필터링 이해



아이템 기반  
(Item - Item)

- 특정 상품과 유사한 좋은 평가를 받은 다른 비슷한 상품을 추천
- 사용자들로 부터 특정 상품과 비슷한 평가를 받은 상품들은 비슷한 상품으로 간주  
이 상품을 선택한 다른 고객들은 다음 상품도 구매했습니다.

	User 1	User 2	User 3	User 4
Item 1	5	4	4	5
Item 2	4	4	5	5
Item 3	1	1	1	3

Item 1과 Item2의 평점이 매우 비슷하다. Item1과 Item2의 벡터 유사도가 비슷하기 때문에 Item1을 구매한 사용자에게 Item2도 추천할 수 있다.

# 사용자 기반 협업 필터링

- 사용자 A에게 추천될 영화는 프로메테우스 일까? 스타워즈 일까?

	다크 나이트	인터스텔라	엣지 오브 투모로우	프로메테우스	스타워즈
사용자 A	5	4	4		
사용자 B	5	3	4	5	3
사용자 C	4	3	3	2	5

# 사용자 기반 협업 필터링

- 사용자 A에게 추천될 영화는 프로메테우스 일까? 스타워즈 일까?

	다크 나이트	인터스텔라	엣지오브 투모로우	프로메테우스	스타워즈
사용자 A	5	4	4		
사용자 B	5	3	4	5	3
사용자 C	4	3	3	2	5

상호간 유사도가 높다

- 사용자 A는 사용자 C보다 사용자 B와 영화 평점 측면에서 유사도가 높다.
- 따라서 사용자 A에게는 사용자 B가 재미있게 본(평점을 높게 준) **프로메테우스**를 추천 하는 것이 좋다.

# 아이템 기반 협업 필터링

- 사용자 D가 다크나이트를 재미있게 보았다. 다크나이트와 유사한 영화를 추천하려고 하는데, 어떤 영화가 어울릴까?

	사용자 A	사용자 B	사용자 C	사용자 D	사용자 E
다크나이트	5	4	5	5	5
프로메테우스	5	4	4		5
스타워즈	4	3	3		4

# 아이템 기반 협업 필터링

- 사용자 D가 다크나이트를 재미있게 보았다. 다크나이트와 유사한 영화를 추천하려고 하는데, 어떤 영화가 어울릴까?

상호간  
유사도가 높다

	사용자 A	사용자 B	사용자 C	사용자 D	사용자 E
다크나이트	5	4	5	5	5
프로메테우스	5	4	4		5
스타워즈	4	3	3		4

- 여러 사용자들의 평점을 기준으로 볼 때 다크나이트와 가장 유사한 영화는 프로메테우스이다.
- 사용자 D가 다크나이트를 재미있게 봤기 때문에, 이와 비슷하게 프로메테우스도 높은 평점을 줄 것으로 예상된다.

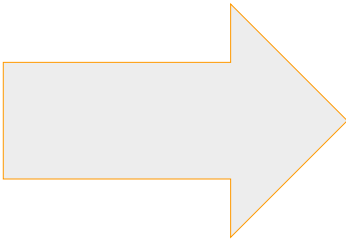
# 사용자 기반 VS 아이템 기반

- 일반적으로 사용자 기반 보다 아이템 기반 방식이 더 선호된다.
- 사람간의 특성은 상대적으로 다양한 요소들에 기반하기 때문이다.
  - 단순히 동일한 상품을 구입하였다고 유사한 사람이라고 판단하기 어려운 경우가 더 많기 때문이다.



# 협업 필터링을 위한 코사인 유사도

기준 행				
0	1	7	2	
1	1	2	4	
2	0	8	3	
3	2	0	3	



cosine\_similarities

비교 대상 기준 행	0	1	2	3
0	1	0.68 (0 - 1 유사도)	0.99 (0 - 2 유사도)	0.3 (0 - 3 유사도)
1	0.68 (1 - 0 유사도)	1	0.72 (1 - 2 유사도)	0.85 (1 - 3 유사도)
2	0.99 (2 - 0 유사도)	0.72 (2 - 1 유사도)	1	0.29 (2 - 3 유사도)
3	0.3 (3 - 0 유사도)	0.85 (3 - 1 유사도)	0.29 (3 - 2 유사도)	1

# 아이템 기반 협업 필터링의 개인화된 영화 추천

- 아이템 기반의 협업 필터링에서 개인화된 평점 예측은 Weighted Rating sum 방식을 이용한다.
  - 사용자  $u$ 의 아이템  $i$ 에 대한 평점 예측을 사용자  $u$ 가 아이템  $i$ 와 유사한 다른 아이템들 ( $N$ 개의 다른 아이템)의 합으로 계산하되, 아이템  $i$ 와 다른 아이템들 간의 유사도를 반영한 합으로 계산한다.

$$\hat{R}_{u,i} = \sum(S_{i,N} \times R_{u,N}) / \sum(|S_{i,N}|)$$

- $\hat{R}_{u,i}$  : 사용자  $u$ , 아이템  $i$ 의 개인화된 예측 평점 값
- $S_{i,N}$  : 아이템  $i$ 와 가장 유사도가 높은 Top- $N$ 개 아이템의 유사도 벡터
- $R_{u,N}$  : 사용자  $u$ 의 아이템  $i$ 와 가장 유사도가 높은 Top- $N$ 개 아이템에 대한 실제 평점 벡터

사용자  $u$ 가 아이템  $i$ 에 대해 몇 점의 평점을 줄까?를 예측해보자.

# 아이템 기반 협업 필터링의 개인화된 영화 추천

- 아이템 기반의 협업 필터링에서 개인화된 평점 예측은 Weighted Rating sum 방식을 이용한다.
  - 사용자  $u$ 의 아이템  $i$ 에 대한 평점 예측을 사용자  $u$ 가 아이템  $i$ 와 유사한 다른 아이템들 ( $N$ 개의 다른 아이템)의 합으로 계산하되, 아이템  $i$ 와 다른 아이템들 간의 유사도를 반영한 합으로 계산한다.

$$\hat{R}_{u,i} = \sum(S_{i,N} \times R_{u,N}) / \sum(|S_{i,N}|)$$

사용자  $u$ 의 상품 평점  $R_{u,N}$

Item $j$	Item $k$	Item $l$	Item $m$	Item $n$
5	4	1	3	2

상품  $i$ 와 비슷한 5개 상품에 대한 점수

# 아이템 기반 협업 필터링의 개인화된 영화 추천

- 아이템 기반의 협업 필터링에서 개인화된 평점 예측은 Weighted Rating sum 방식을 이용한다.
  - 사용자  $u$ 의 아이템  $i$ 에 대한 평점 예측을 사용자  $u$ 가 아이템  $i$ 와 유사한 다른 아이템들 ( $N$ 개의 다른 아이템)의 합으로 계산하되, 아이템  $i$ 와 다른 아이템들 간의 유사도를 반영한 합으로 계산한다.

$$\hat{R}_{u,i} = \sum(S_{i,N} \times R_{u,N}) / \sum(|S_{i,N}|)$$

사용자  $u$ 의 상품 평점  $R_{u,N}$

Item $j$	Item $k$	Item $l$	Item $m$	Item $n$
5	4	1	3	2

상품  $i$ 와 비슷한 상품들의 유사도

$sim(i,j)$	$sim(i,k)$	$sim(i,l)$	$sim(i,m)$	$sim(i,n)$
0.2	0.1	0.4	0.1	0.2

상품  $i$ 와 다른 상품들 과의 유사도

# 아이템 기반 협업 필터링의 개인화된 영화 추천

- 아이템 기반의 협업 필터링에서 개인화된 평점 예측은 Weighted Rating sum 방식을 이용한다.
  - 사용자  $u$ 의 아이템  $i$ 에 대한 평점 예측을 사용자  $u$ 가 아이템  $i$ 와 유사한 다른 아이템들 ( $N$ 개의 다른 아이템)의 합으로 계산하되, 아이템  $i$ 와 다른 아이템들 간의 유사도를 반영한 합으로 계산한다.

$$\hat{R}_{u,i} = \sum(S_{i,N} \times R_{u,N}) / \sum(|S_{i,N}|)$$

사용자  $u$ 의 상품 평점  $R_{u,N}$

Item $j$	Item $k$	Item $l$	Item $m$	Item $n$
5	4	1	3	2

상품  $i$ 와 비슷한 상품들의 유사도

$sim(i,j)$	$sim(i,k)$	$sim(i,l)$	$sim(i,m)$	$sim(i,n)$
0.2	0.1	0.4	0.1	0.2

$$5 \times 0.2 + 4 \times 0.1 + 1 \times 0.4 + 3 \times 0.1 + 2 \times 0.2 = 2.5$$

# 아이템 기반 협업 필터링의 개인화된 영화 추천

- 아이템 기반의 협업 필터링에서 개인화된 평점 예측은 Weighted Rating sum 방식을 이용한다.
  - 사용자  $u$ 의 아이템  $i$ 에 대한 평점 예측을 사용자  $u$ 가 아이템  $i$ 와 유사한 다른 아이템들 ( $N$ 개의 다른 아이템)의 합으로 계산하되, 아이템  $i$ 와 다른 아이템들 간의 유사도를 반영한 합으로 계산한다.

$$\hat{R}_{u,i} = \sum(S_{i,N} \times R_{u,N}) / \sum(|S_{i,N}|)$$

사용자  $u$ 의 상품 평점  $R_{u,N}$

Item $j$	Item $k$	Item $l$	Item $m$	Item $n$
5	4	1	3	2

상품  $i$ 와 비슷한 상품들의 유사도

$sim(i,j)$	$sim(i,k)$	$sim(i,l)$	$sim(i,m)$	$sim(i,n)$
0.2	0.1	0.4	0.1	0.2

Item $j$	Item $k$	Item $l$	Item $m$	Item $n$
5	4	1	3	2

0.2	$sim(i,k)$
0.1	$sim(i,k)$
0.4	$sim(i,l)$
0.1	$sim(i,m)$
0.2	$sim(i,n)$

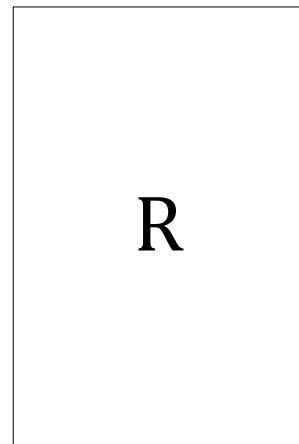
# 아이템 기반 협업 필터링 실습 (IMDB)

- 미국의 유명한 영화 평점 사이트인 IMDB 데이터 세트를 캐글에서 다운 받기
  - <https://files.grouplens.org/datasets/movielens/ml-latest-small.zip>
- 아이템 기반 협업 필터링은 다음과 같은 프로세스에 의해 구현함
  1. 사용자-아이템 행렬 데이터를 아이템-사용자 행렬 데이터로 변환
  2. 아이템 간의 코사인 유사도로 아이템 유사도 계산
  3. 사용자가 관람(구매)하지 않은 아이템들 중에서 아이템간 유사도를 반영한 예측 점수 계산
  4. 예측 점수가 가장 높은 순으로 아이템 추천

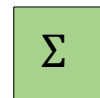
# 잠재 요인 협업 필터링 개요

- 잠재(Latent) 요인 협업 필터링은 사용자-아이템 평점 행렬 속에 숨어 있는 잠재 요인을 추출해 추천 예측을 시도
- 대규모 다차원 행렬을 SVD와 같은 행렬 분해(Matrix Factorization) 기법으로 분해하는 과정에서 잠재 요인 추출
- 추출된 잠재 요인을 기반으로 사용자-아이템 평점 행렬을 재 구성하면서 추천을 구현

원본 사용자-아이템 평점 행렬  
(희소 행렬)



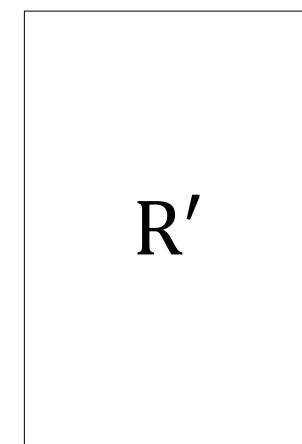
$\approx$



Latent-Item Matrix

$=$

잡음이 제거된 형태로 재구성  
(값이 모두 채워진 행렬)



User-Latent Matrix



# 잠재 요인 협업 필터링 이해

- 잠재 요인 협업 필터링은 사용자-아이템 평점 행렬 속에 숨어 있는 잠재 요인을 추출해 추천 예측을 할 수 있게 한다.
  - 잠재 요인 자체를 정확하게 무엇이다 라고 정의 하기는 힘들다.
  - 사용자나 영화의 숨어 있는 내재된 특징을 표현하기란 쉬운 일이 아니기에, 어떤 내재된 특징이 있을 것이라고 짐작만 한다.
  - 따라서 사용자 레벨의 잠재 요인을 장르라고 생각하면, 아이템 레벨의 잠재 요인도 장르라고 생각한다.

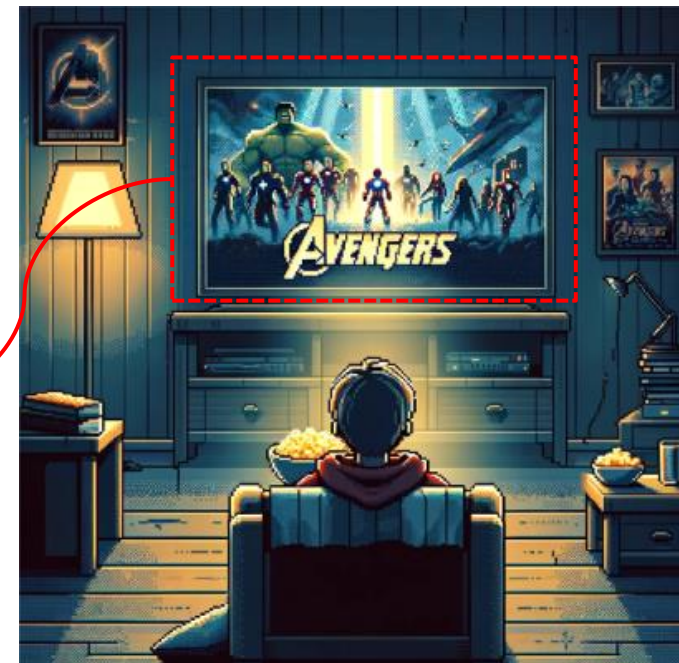


사용자 레벨의 잠재 요인

- Super Hero : 0.7
- Action : 0.6
- Adventure : 0.5
- Sci-Fi : 0.3
- Fantasy : 0.5
- Drama : 0.3
- Romance : 0.1

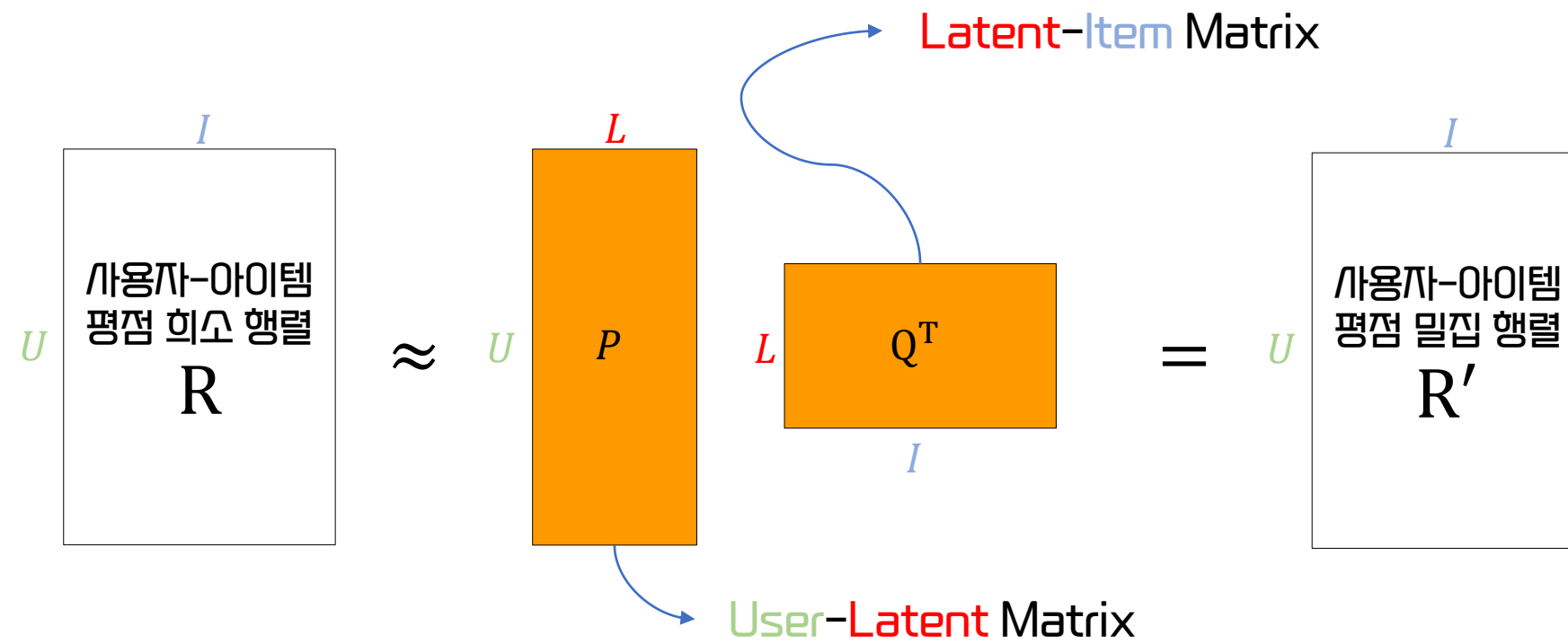
아이템 레벨의 잠재 요인

- Super Hero : 0.8
- Action : 0.7
- Adventure : 0.6
- Sci-Fi : 0.5
- Fantasy : 0.4
- Drama : 0.4
- Romance : 0.0



# 행렬 분해를 통한 잠재 요인 협업 필터링

- 잠재 요인 협업 필터링의 행렬 분해 목표는 희소 행렬 형태의 사용자-아이템 평점 행렬을 밀집(Dense) 행렬 형태의 사용자-잠재 요인 행렬과 잠재 요인-아이템 행렬로 분해 한 뒤 이를 재 결합하여 밀집 행렬 형태의 사용자-아이템 평점 행렬을 생성하여 사용자에게 새로운 아이템을 추천하는 것이다.



# 행렬 분해를 통한 잠재 요인 협업 필터링

원본 사용자-아이템 평점 행렬  
(희소 행렬)

	item1	item2	item3	item4	item5
User 1	4			2	
User 2		5		3	
User 3			3	4	4
User 4	5	2	1	2	

≈

사용자-잠재 요인 행렬

	factor1	factor2
U1	0.94	0.96
U2	2.14	0.08
U3	1.93	1.79
u4	0.58	1.56

잠재 요인-아이템 행렬

	item1	item2	item3	Item4	item5
factor1	1.7	2.3	1.41	1.36	0.41
factor2	2.49	0.41	0.14	0.75	1.77

원본 사용자-아이템 행렬이 사용자-잠재 요인 행렬과 잠재 요인-아이템 행렬로 분해되었다. 이를 내적시키면 원래 모양의 밀집 행렬이 만들어 진다.

# 행렬 분해를 통한 잠재 요인 협업 필터링

사용자-잠재 요인 행렬

	factor1	factor2
U1	0.94	0.96
U2	2.14	0.08
U3	1.93	1.79
u4	0.58	1.56

잠재 요인-아이템 행렬

	item1	item2	item3	Item4	item5
factor1	1.7	2.3	1.41	1.36	0.41
factor2	2.49	0.41	0.14	0.75	1.77

=

원본 사용자-아이템 평점 행렬  
(밀집 행렬)

	item1	item2	item3	item4	item5
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

# 행렬 분해를 통한 잠재 요인 협업 필터링 예시

사용자 1의 영화 1에 대한 점수  $R[1, 1]$

사용자-아이템 평점 행렬 R

	item1	item2	item3	item4	item5
User 1	4	?		2	
User 2		5		3	
User 3			3	4	4
User 4	5	2	1	2	

4  $\approx$

사용자 1의-장르 선호도 행렬  $P[1, :]$

	Action	SF
U1	0.94	0.96

영화 1의 장르 요소 행렬  $Q.T[:, 1]$

	item1
Action	1.7
SF	2.49

\*

$$= 0.94 \cdot 1.7 + 0.96 \cdot 2.49 = 3.98$$

사용자 1의 장르 선호도 행렬 P와 영화 1의 장르 요소 행렬 Q.T를 내적했더니 원본 데이터 행렬의 값인 4와 유사한 값이 나왔다. 이를 이용해 영화 2에 대한 선호도를 구해보자

# 행렬 분해를 통한 잠재 요인 협업 필터링 예시

사용자 1의 영화 2에 대한 점수  $R[1, 2]$   
아직 몇 점을 주었는지에 대한 정보가 없다.

사용자-아이템 평점 행렬 R

	item1	item2	item3	item4	item5
User 1	4	?		2	
User 2		5		3	
User 3			3	4	4
User 4	5	2	1	2	

?  $\approx$

사용자 1의-장르 선호도 행렬  $P[1, :]$

	Action	SF
U1	0.94	0.96

영화 1의 장르 요소 행렬  $Q.T[:, 2]$

	item1
Action	2.3
SF	0.41

\*

$$= 0.94 \times 2.3 + 0.96 \times 0.41 = 2.56$$

영화 2는 영화1과 다르게 Action과 SF에 대한 잠재 값이 낮은 것을 알 수 있다. 따라서 영화 1에 비해 사용자가 낮은 점수를 예측할 것이라는 것을 짐작할 수 있다.

# 행렬 분해 이슈

	item1	item2	item3	...	Item M
User 1	3	?	3	?	?
User 2	4	2	?	?	3
User 3	?	1	2	?	2
User 4	1	?	?	?	?
...	...	...	...	...	...
User N	4	2	?	?	5

- 희소 행렬은 값 자체가 많이 들어있지 않은 행렬이다.
- 따라서 사용자-영화 평점 행렬도 값이 매우 희소한 행렬이다.
- SVD는 값이 모두 들어 있어야 수행할 수 있기 때문에 행렬 P와 행렬 Q를 일반적인 방법으로는 분해할 수가 없다.

P와 Q를 모르는데 어떻게 R을 예측 할 수 있을까?

# 행렬 분해 이슈

	item1	item2	item3	...	Item M
User 1	3	?	3	?	?
User 2	4	2	?	?	3
User 3	?	1	2	?	2
User 4	1	?	?	?	?
...	...	...	...	...	...
User N	4	2	?	?	5

- 희소 행렬은 값 자체가 많이 들어있지 않은 행렬이다.
- 따라서 사용자-영화 평점 행렬도 값이 매우 희소한 행렬이다.
- SVD는 값이 모두 들어 있어야 수행할 수 있기 때문에 행렬 P와 행렬 Q를 일반적인 방법으로는 분해할 수가 없다.

경사 하강법을 이용하면 된다!

- 경사 하강법을 이용하여 P와 Q에 기반한 예측 R 값이 실제 R 값과 가장 최소의 오류를 가질 수 있도록 비용함수 최적화를 통해 P와 Q를 최적화 할 수 있도록 유추한다.



# 경사 하강법 기반의 행렬 분해

- 경사 하강법을 이용한 행렬 분해 방법은 P와 Q 행렬로 계산된 예측행렬 R 행렬 값이 실제 R 행렬 값과 가장 최소의 오류를 가질 수 있도록 반복적인 비용 함수 최적화를 통해 P와 Q를 유추해 나가는 것이다.
  - P와 Q를 임의의 값을 가진 행렬로 설정
    - 보통은 정규 분포 랜덤 값으로 없는 부분을 채운다.
  - P와 Q.T 값을 곱해 예측 R 행렬을 계산하고 예측 R 행렬과 실제 R 행렬에 해당하는 오류 값을 계산
  - 이 오류 값을 최소화 할 수 있도록 P와 Q 행렬을 적절한 값으로 각각 업데이트
  - 만족할 만한 오류 값을 가질 때까지 2, 3번 작업을 반복하면서 P와 Q를 업데이트하여 근사화한다.

# 경사 하강법 기반의 행렬 분해

- 경사 하강법을 이용한 행렬 분해 방법은 P와 Q 행렬로 계산된 예측행렬 R 행렬 값이 실제 R 행렬 값과 가장 최소의 오류를 가질 수 있도록 반복적인 비용 함수 최적화를 통해 P와 Q를 유추해 나가는 것이다.
- P와 Q를 임의의 값을 가진 행렬로 설정
    - 보통은 정규 분포 랜덤 값으로 없는 부분을 채운다.
  - P와 Q.T 값을 곱해 예측 R 행렬을 계산하고 예측 R 행렬과 실제 R 행렬에 해당하는 오류 값을 계산
  - 이 오류 값을 최소화 할 수 있도록 P와 Q 행렬을 적절한 값으로 각각 업데이트
  - 만족할 만한 오류 값을 가질 때까지 2, 3번 작업을 반복하면서 P와 Q를 업데이트하여 근사화한다.

실제 값과 예측값의 오류 최소화와 L2 규제를 고려한 비용 함수 식

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

# 경사 하강법 기반의 행렬 분해

- 경사 하강법을 이용한 행렬 분해 방법은 P와 Q 행렬로 계산된 예측행렬 R 행렬 값이 실제 R 행렬 값과 가장 최소의 오류를 가질 수 있도록 반복적인 비용 함수 최적화를 통해 P와 Q를 유추해 나가는 것이다.
- P와 Q를 임의의 값을 가진 행렬로 설정
    - 보통은 정규 분포 랜덤 값으로 없는 부분을 채운다.
  - P와 Q.T 값을 곱해 예측 R 행렬을 계산하고 예측 R 행렬과 실제 R 행렬에 해당하는 오류 값을 계산
  - 이 오류 값을 최소화 할 수 있도록 P와 Q 행렬을 적절한 값으로 각각 업데이트
  - 만족할 만한 오류 값을 가질 때까지 2, 3번 작업을 반복하면서 P와 Q를 업데이트하여 근사화한다.

실제 값과 예측값의 오류 최소화와 L2 규제를 고려한 비용 함수 식

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

실제값과 예측값의 오류 최소화

과적합 개선을 위한 L2 규제

# 경사 하강법 기반의 행렬 분해

- 실제 R 행렬 값과 예측 R 행렬 값의 차이를 최소화하는 방향성을 가지고 P 행렬과 Q 행렬에 업데이트 값을 반복적으로 수행하면서 최적화된 예측 R 행렬을 구하는 방식이 경사 하강법 기반의 행렬 분해이다.

비용 함수를 최소화하기 위해/  
새롭게 업데이트 되는  $p'_u$ 와  $q'_i$  식

$$p'_u = p_u + \eta(e_{u,i} \cdot q_i - \lambda \cdot p_u)$$

$$q'_i = q_i + \eta(e_{u,i} \cdot p_u - \lambda \cdot q_i)$$

- $p_u$  : P 행렬의 사용자 u행 벡터
- $q_i^t$  : Q 행렬의 아이템 i행의 전치 벡터
- $e_{u,i}$  : u행 i열에 위치한 실제 행렬 값과 예측 행렬 값의 차이 오류
  - $e_{u,i} = r_{u,i} - \hat{r}_{u,i}$
  - $r_{u,i}$  : R 행렬의 u행 i열의 값
  - $\hat{r}_{u,i} : p_u \cdot q_i^t$ 로 계산되며 u행 i열에 위치한 행렬의 예측값
- $\eta$  : SGD 학습률
- $\lambda$  : L2 규제(Regularization) 계수