



추천 101

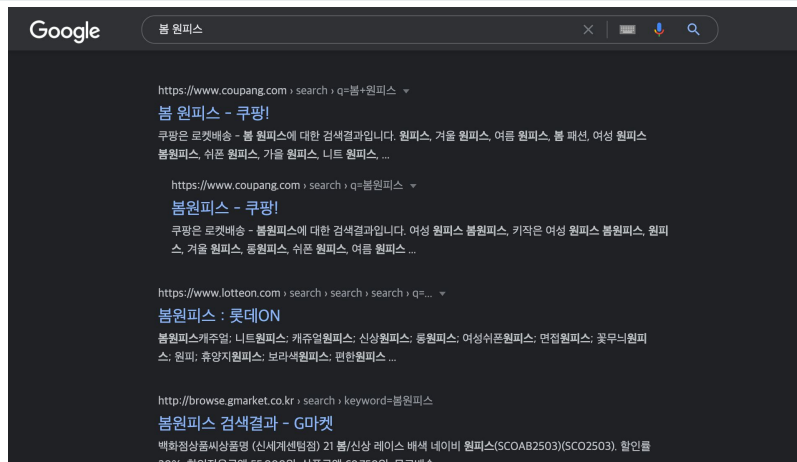


Contents





- 추천이란?
 - 정의
 - Score, User, Item
 - 추천 알고리즘 종류
 - Metric
- 대표적인 알고리즘 2개 소개
 - content-based RS
 - collaborative filtering RS
- 논문 읽어보기

추천 시스템(RS)이란

- 사용자(User)와 상품(Item)으로 구성된 시스템
- 특정 사용자가 좋아할 상품을 추천
- 비슷한 상품을 좋아할 사용자를 추천
- Item이든 User든 관심 갖을만한 정보를 추천 (핵심)



함께 보면 좋은 상품 더보기 >

 <p>신세계백화점 레코브 라인배세 포인트 니트가디건(LW31B2 KC910X) 48,646원 59,000원 ◯ ★★★★★ (4개)</p>	 <p>신세계백화점 JJ JIGOTT 하프넥 파베기 니트 가디건 (GLAN1C D10) 30,771원 79,000원 ◯ ★★★★★ (56개)</p>	 <p>신세계백화점 레코브 가든 니트가디건(LW31AYKC903X) 51,119원 62,000원 ◯ ★★★★★ (18개)</p>	 <p>신세계백화점 플라스틱아일랜드 카라단락 꽃베세 가디건(PM4KG103) 31,200원 39,000원 ◯ ★★★★★ (8개) 매장픽업 가능점도 ◯</p>
--	--	---	--



추천 시스템(RS)이란

- 서비스의 성장과 정보의 다양화
- 인터넷에서 찾을 수 있는 정보가 매우 많음
- 수 많은 데이터 속에서 사용자가 적절한 정보를 찾는데 시간이 매우 오래 걸림
- 사용자가 정보를 수집하고 찾는 시간을 줄여주는 것이 목적 (핵심)

검색 서비스	추천 서비스
사용자가 요구하여 작동	사용자가 요구하기 전 작동
사용자 스스로 원하는 바를 알고 있음	사용자 스스로 원하는 바를 정확히 알지 못함



추천 시스템을 사용해야 되는 이유

1. 더 많은 아이템 판매 가능
 - 기업이 추천 시스템을 사용하는 가장 큰 이유
2. 더 다양한 아이템 판매 가능
 - 소비자가 보지 못한 상품 판매 가능
3. 소비자 만족도 증가
 - 플랫폼을 사용하며 만족도가 증가하며 플랫폼에 더 머무르는 이유가 됨
4. 충성도 높은 고객이 증가
5. 고객이 원하는 것이 무엇인지 알 수 있다 (니즈 파악)
 - 데이터가 쌓이면 쌓일 수록 추천 알고리즘의 성능 향상

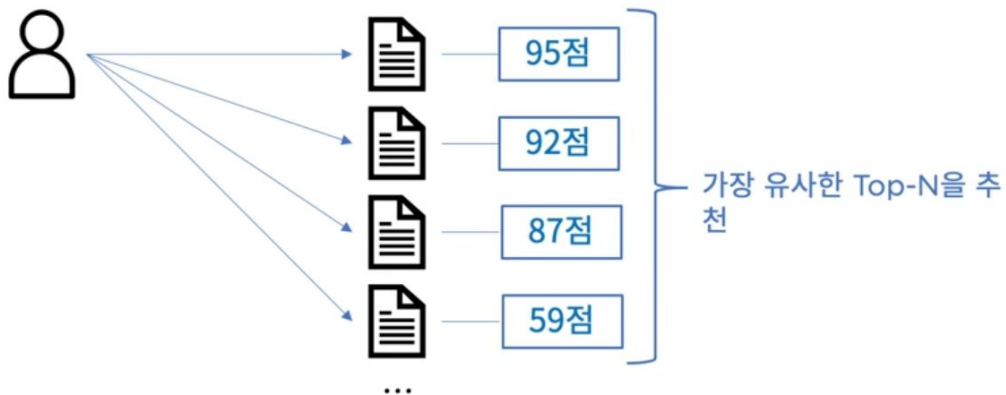


추천 시스템이 풀고자 하는 문제

1. 랭킹 문제
 - 특정 유저가 특정 아이템에 대한 평점 (or 점수)을 정확하게 예측할 필요 없음
 - 특정 아이템을 좋아할만한 Top-K 유저를 선정할 수 있음
 - 특정 유저가 좋아할만한 Top-K 아이템을 선정할 수 있음
 - 순서가 중요하다는 의미
2. 예측 문제
 - 유저-아이템 조합에서 평점 (or 점수)를 예측
 - 유저-아이템 행렬을 채우는 문제
 - EX) 유저가 특정 영화에 대해 몇점을 평점으로 줬을지 예측
 - 유저(m 명) - 아이템(n 개의) $m * n$ 행렬, 그러나 비어있는 부분 존재
 - 관측값(Observed value)은 모델 학습에 사용
 - 결측값(Missing value)은 모델 예측에 사용
3. 추천 시스템은 랭킹 문제와 예측 문제를 적절히 조합해 서비스, 데이터를 고려하여 문제 정의를 해야 함

추천점수(Score)란?

- 사용자 또는 아이템을 추천하기 위해 각각의 아이템 또는 사용자에게 대한 정량화된 기준 필요
- 분석된 사용자와 아이템 정보를 바탕으로 추천점수 계산
- 사용자 또는 아이템 프로필에서 어떤 정보를 사용할지에 따라 추천알고리즘 결정
- 추천 알고리즘의 목적은? 점수화 (Scoring) 하는 것





사용자(User)와 상품(Item)

- 사용자와 아이템 사이의 관계를 분석하고 연관관계를 찾아 점수화 => 사용자로부터 선호도(preferences)의 정도를 데이터화
 - 사용자 정보
 - 사용자 또는 사용자 그룹을 분석 가능한 요소로 프로파일링
 - 사용자 고유 정보 (나이, 성별, 지역 등 개인 신상 정보)
 - 사용자 행동 패턴 (Cookie, 웹 페이지 방문 기록, 클릭 정보 등)
 - 사용자 정보를 수집하는 방법
 - 직접적(Explicit) 방법 : 설문조사, 평가, 피드백 등
 - 간접적(Implicit) 방법 : 웹페이지 머무는 시간, 클릭 패턴, 검색 로그 등
 - 아이템 정보
 - 플랫폼마다 정의하는 아이템의 정보가 다름
 - 쇼핑몰, 여행지, 동영상 추천 등등
 - 아이템 프로필에 속하는 정보
 - 아이템 고유 정보 (가격, 색상, 내용 등)
 - 아이템을 좋아하거나 구매한 사용자 정보

Explicit & Implicit Feedback

	Explicit	Implicit
장점	유저가 직접 입력하여 명확한 선호도를 알 수 있다	다양한 방법을 데이터를 얻을 수 있다
단점	데이터를 구하기 어렵다	데이터의 실제 의미를 파악하기 어렵다 노이즈, 결측값이 존재한다 부정적인 피드백을 얻기 어렵다

	i_1	i_2	i_3	\dots	i_N
u_1	0	1	0	...	2
u_2	2	0	4	...	5
u_3	3	0	0	...	0
\vdots	\vdots	\vdots	\vdots	...	\vdots
u_M	0	0	2	...	3

$\mathbb{R}^{M \times N}$

(a) Explicit feedback

	i_1	i_2	i_3	\dots	i_N
u_1	0	1	0	...	1
u_2	1	0	1	...	1
u_3	1	0	0	...	0
\vdots	\vdots	\vdots	\vdots	...	\vdots
u_M	0	0	1	...	1

$\mathbb{R}^{M \times N}$

(b) Implicit feedback



추천 시스템 연구 현황

- 정보검색 (Information Retrieval) 에서 파생, 비슷하지만 비교적 새로운 연구 분야
- 다양한 형태의 추천 시스템 연구 / 여러 기업과 학회에서 다양하게 연구 진행 중
 - SIGIR, WWW, ACM RecSys, IEEE 등
 - 유튜브, 넷플릭스, 왓차, 쿠팡, 아마존, 멜론 등
- Netflix Prize 추천 대회
 - 넷플릭스의 온라인 DVD 대여와 영화 스트리밍 서비스
 - 480,189명의 사용자가 17,770개의 영화 -> 100,480,507개 평점 부여
 - MF 알고리즘 방법으로 넷플릭스 기존 알고리즘 대비 +10.6% 향상

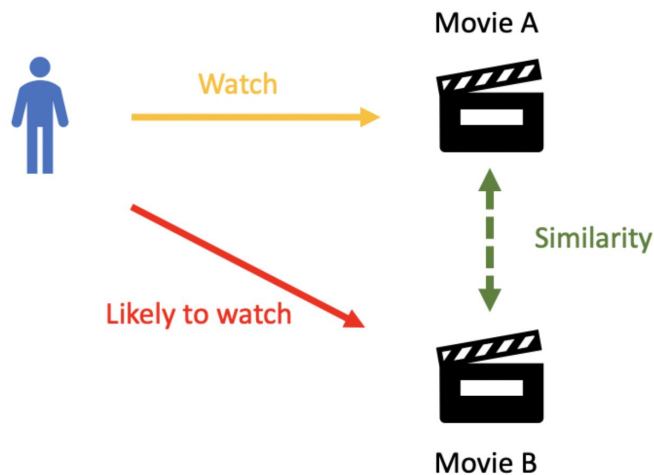


추천 알고리즘 종류

- 고전적 방법론
 - Contents-based Recommender System
 - Collaborative Filtering
 - Hybrid Recommender System
 - Knowledge-based Recommender System
 - Context-based Recommender System
 - Community-based Recommender System
- 딥러닝 기반 방법론

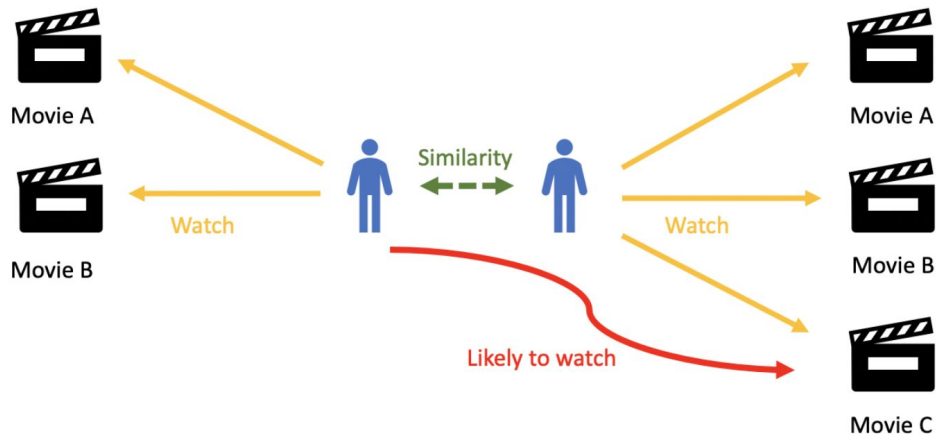
Contents-based Recommender System (컨텐츠기반 추천시스템)

- 사용자가 과거에 좋아했던 아이템을 파악하고, 그 아이템과 비슷한 아이템을 추천
- EX) 스파이더맨에 4.5점 평점을 부여한 유저는 타이타닉보다 마블 영화를 더 좋아할 것이다.



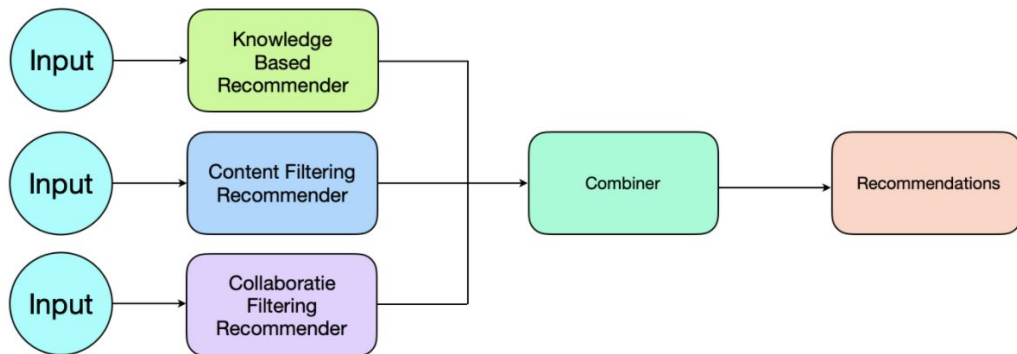
Collaborative Filtering (협업 필터링)

- 비슷한 성향 또는 취향을 갖는 다른 유저가 좋아한 아이템을 현재 유저에게 추천
- EX) 스파이더맨에 4.5점을 준 2명의 유저 -> 유저 A가 과거에 좋아했던 마블 영화를 유저 B에게 추천



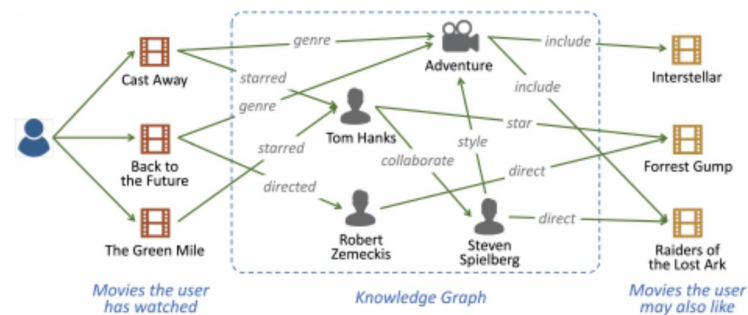
Hybrid Recommender System

- EX) Content-based와 Collaborative Filtering의 장, 단점을 상호보완
- Collaborative Filtering은 새로운 아이টে에 대한 추천을 하기 어려움 (cold-start 문제)
- Content-based 기법으로 cold-start 문제 보완
- 다른 추천 알고리즘도 함께 병합하여 사용할 수 있음



Other Recommendation Algorithms

- Context-based Recommendation
 - 유저 - 아이템 정보 이외의 context 정보까지 고려
 - Location-based, Time-sensitive, mood(감정 정보)
- Knowledge-based Recommendation
 - 유저의 선호도와 아이템들의 속성(attribute)를 함께 고려하여 추천
- Community-based Recommendation
 - 사용자의 친구 또는 속한 커뮤니티의 선호도를 바탕으로
 - SNS의 뉴스피드, SNS 네트워크 데이터



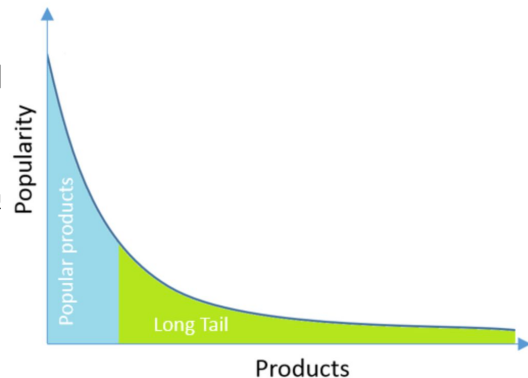


추천 시스템의 한계

- Scalability
 - 실제 서비스 상황은 다양한 종류의 데이터
 - 학습 또는 분석에 사용한 데이터와는 전혀 다른 실전 데이터 (트렌드의 변화로 새로운 데이터 발생)
- Proactive Recommender System
 - 특별한 요청이 없어도 사전에 먼저 제공하는 것이 추천 서비스
 - 유저에게 끊임없이 좋은 정보를 추천할 수 있는가 (주어진 데이터를 기반으로 학습하고 추천하는 모델 입장에서 어려운 일)
- Cold-Start Problem
 - 추천 서비스를 위한 데이터 부족 (100개의 상품 중 유저가 10개만 구입한다면 90개의 데이터가 부족한 것)
 - 기본적인 성능을 보장하는 CF 모델 구축이 쉽지 않음
- Privacy preserving Recommender System
 - 개인정보 등 유저 정보가 가장 중요하지만, 직접적으로 사용하기 어려움

추천 시스템의 한계

- Mobile device and Usage Contexts
 - 개별 상황 또는 환경(디바이스 및 위치 기반) 등에 따라 다른 컨텍스트를 반영하기 어려움
- Long-term and Short-term user preference
 - 추천받고 싶은 아이템이 현재도 관심이 있는지 과거 어느 시기에만 관심이 있는지 파악하기 어려움
- Generic User models and Cross Domain Recommender System
 - 하나의 모델을 여러가지 데이터에 적용하기 어려움
 - 비슷한 도메인의 데이터를 활용해도 동일한 성능의 추천시스템을 기대하기
- Starvation and Diversity
 - Starvation: 필요한 컴퓨터 자원을 끊임없이 가져오지 못하는 상황
 - 유저/아이템이 다양하여 모든 경우를 커버해야 하나 그러기에 리소스가 부





추천 시스템을 만들기 위해 고려해야 할 것

- 목적
 - 비즈니스 역량 확대, 매출 증대, CTR 증가 등 다양한 목적
- 추천을 위한 적절한 데이터
- 추천 알고리즘
 - 성능, 속도, 설명 가능한지 고려
- 플랫폼 내에서의 추천 시스템의 역할
 - 추천 시스템이 주요 기능인지, 필수인지 아니면 옵션인지
- 플랫폼 사용자
 - 사용자는 어떤 환경에서, 어떻게 서비스를 이용하는지, 사용자에게 어떤 정보를 받는지

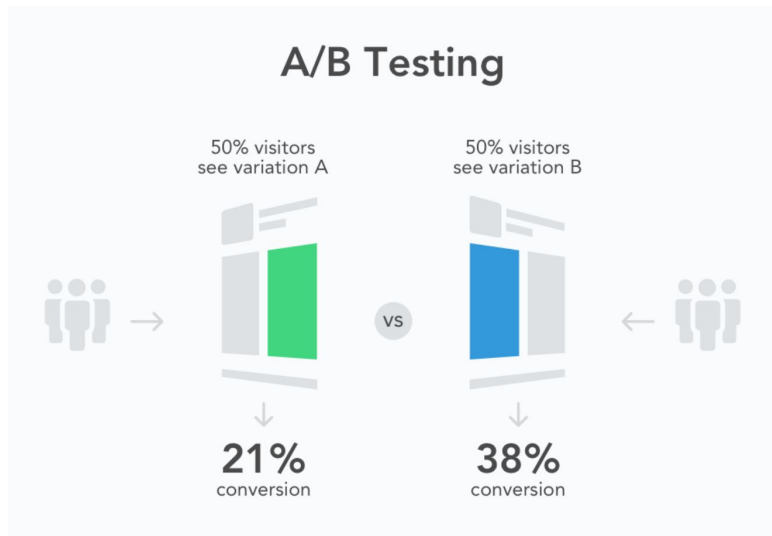


추천 시스템 평가

- Offline Evaluation
 - 데이터를 train/valid/test로 나누어 평가
 - 정량적인 평가지표를 활용한 객관적인 평가 가능
 - 수집된 데이터 상에서 평가가 이루어짐 -> 실제 서비스 상황에서 다르게 적용될 수 있음
 - 다양한 추천 알고리즘을 쉽고 빠르게 평가할 수 있음
 - 추천 시스템은 정답이 없기 때문에 어떤 지표가 더 낫다라고 이야기 하기 어려움
 - 추천 시스템의 목표와 평가 기준을 잘 설계해야 함
- Online Evaluation
 - 추천 시스템이 적용된 플랫폼에서 실제 사용자의 피드백, 평점 등 활용
 - 수집할 수 있는 데이터의 한계가 있으나, 실제 사용자의 데이터로 평가 가능
 - 추천 서비스 향상에 직접적인 도움을 줄 수 있음

A/B Testing

- Online Evaluation 방법
- 사용자를 두 그룹으로 나누고, A(기존 모델-대조군), B(새로운 모델-실험군) 두 가지 모델을 사용하여 실험 진행
- 새로운 알고리즘이 실제 데이터 상에서 잘 동작하는지 확인 가능





Offline Metric

- Binary based metric
 - accuracy
 - rmse
 - precision, recall, f1-score
 - precision@K, recall@K
- Rank aware metric
 - MRR (mean reciprocal rank)
 - MAP (mean average precision)
 - NDCG (normalized discounted cumulative gain)



Normalized Discounted Cumulative Gain

- 랭킹 추천에 많이 사용되는 평가 지표
- Top-N 랭킹 리스트에서 더 관련있는(관심있는) 아이템을 덜 관련있는 것 보다 더 상위에 노출시키는지 평가
- 가장 이상적인 랭킹(정답 랭킹)과 현재 점수를 활용한 랭킹 사이의 점수를 cumulative하게 비교
- 1에 가까울 수록 좋은 랭킹
- 유저가 아이템에 대한 평가를 하지 않는 경우, 관련도를 어떻게 처리해야 할지 정의해야 함



Normalized Discounted Cumulative Gain

- **CG**: 추천한 아이템의 relevance의 합
 - 순서를 고려하지 않은 값, 비중도 동일
- **DCG**: CG에 순서에 따른 할인 개념 추가
 - log normalization을 적용해 하위권 penalty 부여
- **IDCG**: 최선의 추천을 했을 때 받는 DCG 값 (이상적인)
- **NDCG**: DCG를 IDCG로 나누어 정규화 적용
 - DCG는 추천 아이템의 갯수가 많을 수록 커지기 때문에 정확한 성능 평가를 위해 Scale을 맞춰주는 것

$$CG_p = \sum_{i=1}^p rel_i$$

$$DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2 i}$$

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

Normalized Discounted Cumulative Gain

4 documents: d_1, d_2, d_3, d_4

i	Ground Truth		Ranking Function ₁		Ranking Function ₂	
	Document Order	r _i	Document Order	r _i	Document Order	r _i
1	d4	2	d3	2	d3	2
2	d3	2	d4	2	d2	1
3	d2	1	d2	1	d4	2
4	d1	0	d1	0	d1	0
NDCG _{GT} =1.00		NDCG _{RF1} =1.00		NDCG _{RF2} =0.9203		

$$DCG_{GT} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF1} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF2} = 2 + \left(\frac{1}{\log_2 2} + \frac{2}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.2619$$

$$MaxDCG = DCG_{GT} = 4.6309$$

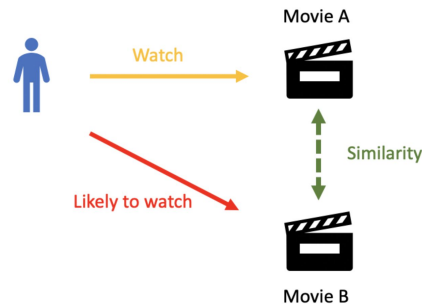


Additional Metric

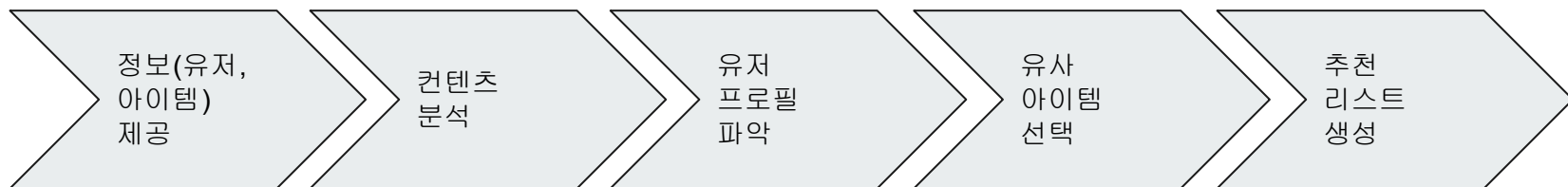
- coverage
 - 전체 아이템 중 얼마나 추천 되었는지/추천할 수 있을지
- popularity
 - 특정 아이템들이 많이 추천 되었는지 (인기많은, 익숙한 아이템들 위주로 추천 되었는지)
- novelty
 - popularity의 반대 개념 (생소한 아이템들이 추천 되었는지)
- serendipity
 - 예상되지 않은 아이템이 추천 되었는지
 - 아예 예상하지 못했던 새로운 아이템을 추천한다는 점에서 novelty와 다름
- diversity
 - 추천 리스트 내에 얼마나 다양한 아이템이 들어있는지
- personalization
 - 각 유저에게 추천된 아이템이 얼마나 다른지

컨텐츠기반 추천시스템

- 사용자가 높은 평점을 주거나 큰 관심을 가졌던 아이템과 유사한 아이템을 “현재 시점”에 추천
 - A와 비슷한 F는 현재 시점에 없을 수 있음
- 예시
 - 웹사이트, 블로그, 뉴스: 비슷한 컨텐츠의 게시글(item)을 찾아서 추천
 - 영화: 같은 배우, 같은 장르, 같은 영화 감독 등 비슷한 특징을 갖는 영화(item)을 찾아
- Point
 - 사용자가 높게 평가한 아이템과, 아직 보지않은 아이템 간의 “유사성을 어떻게 파악”
 - 유사성을 파악하기 위해 아이템의 어떤 아이템 feature들을 활용해야 할 것인가?
 - feature를 추출하는 “알고리즘”이 중요



컨텐츠기반 추천시스템 Flow



텍스트와 같은
비정형
데이터로부터 관련
있는 정보를 얻는
작업 (feature
extraction, vector
representation)

유저가 선호하는
아이템, 취향 파악

머신러닝 등
알고리즘을 통해
유저 데이터 일반화

아이템 중 가장
적절하게
match하는 아이템
선택

관련있는
아이템으로 최종
추천 리스트 만들기

Item Representation

- 아이템의 특징(attributes)을 분석하여 Feature를 추출하고 Vector로 표현 (set of features)
 - 유저의 관심사를 나타낼 많은 정보를 얻어야 유저의 관심사를 파악할 수 있음
- Item-Item 간의 유사도는 보통 Cosine Similarity 활용

스파이더맨: 노 웨이 홈

Spider-Man: No Way Home, 2021

관람객 **★★★★★ 9.12** 기자-평론가 **★★★★★ 7.09**

네타즌 **★★★★★ 8.87** 내 평점 **★★★★★** 등록 >

개요 액션, 모험, SF 미국 148분 2021.12.15 개봉

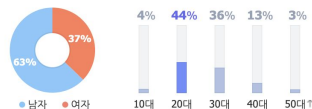
감독 존 왓스

출연 톰 홀랜드(피터 파커/스파이더맨), 켄데이아 콜먼(MJ), 베네딕트 콕스(토니 스타크) ... 더보기 >

등급 [국내 12세 관람가]

총행 예매율 7위 누적관객 7,471,995명(02.11 기준)

성별·나이별 관람추이



예매하기 **17,191**

줄거리

미스터리오의 계략으로 세상에 정체가 탄로난

스파이더맨 피터 파커는 하루 아침에 평범한 일상을 잃게 된다.

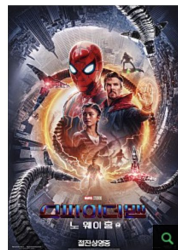
문제를 해결하기 위해 닥터 스트레인지자를 찾아가 도움을 청하지만

뜻하지 않게 멀티버스가 열리면서 각기 다른 차원의 불청객들이 나타난다.

닥터 옥토퍼스를 비롯해 스파이더맨에게 깊은 원한을 가진 숙적들의 강력한 공격에

피터 파커는 사상 최악의 위기를 맞게 되는데...

제작노트 보기 >



주요정보 배우/제작진 포토 동영상 평점 리뷰 상영시간표 행대사/연관영화

네타즌 평점 **★★★★★ 8.87** 19,138명

관람객 평점 **★★★★★ 9.12** 4,159명

이 영화는 **20대 남성**이 좋아하는 **스토리아**가 뛰어난 영화입니다.



그래프는 5분 주기로 업데이트 됩니다.

관람객 평점 **19,080**건 내 평점 등록

공감순 최신순 평점 높은 순 평점 낮은 순

스포일러 보기 관람객 평점만 보기 >

★★★★★ 10 **[관람객]** 상심 대의 내가 싫 대, 아심 대의 나를 만났대

말미나손맛마카롱(seoh****) | 2021.12.15 11:53 신고

9478 457

★★★★★ 10 **[관람객]** 스파이더맨 1 부터 본 사람이면 재미없을 수가 없다

kwan**** | 2021.12.15 11:27 신고

4656 220



User Profile

- 유저의 취향 파악
 - 설문조사, 자발적 키워드 입력 등으로 얻은 프로필
 - Like/Dislike, Ratings, 유저가 직접 작성한 댓글
 - 검색 로그, 아이템 선택 후 구매 또는 선택, 특정 아이템 관련 내용에 비슷한 아이템 tag 이력
 - 이 외에도 다양함
- 유저를 선호하는 아이템 Feature Set의 가중치를 병합한 값으로 표현 (방식은 다양할 수 있음)

Item Representation / User Profile 예시

The Avengers	9
Spiderman	7
Fantastic Four	4

1) user rating
데이터

	Superhero	Comedy	Adventure	Sci-Fi
The Avengers	1	1	1	1
Spiderman	1	0	1	0
Fantastic Four	1	1	0	1

2) item representation

	Superhero	Comedy	Adventure	Sci-Fi
The Avengers	9	9	9	9
Spiderman	7	0	7	0
Fantastic Four	4	4	0	4

3) weighted matrix 생성

(rating * item representation)

Item Representation / User Profile 예시

	Superhero	Comedy	Adventure	Sci-Fi
User Profile	0.32	0.2	0.3	0.2
	↓	↓	↓	↓
	Superhero	Comedy	Adventure	Sci-Fi
Justice League	1	0	1	1
Interstellar	0	0	1	1
The Dark Knight Rises	1	0	0	0

4) user profile (weighted matrix의 summation + normalization)

5) item representation (user가 rating하지 않은 나머지)

에 user profile 을 곱하여 가중치를 얻음

Item Representation / User Profile 예시

	Superhero	Comedy	Adventure	Sci-Fi		Final Weights
Justice League	0.32	0	0.3	0.2	→	0.82
Interstellar	0	0	0.3	0.2	→	0.5
The Dark Knight Rises	0.32	0	0	0	→	0.32

6) 각 row를 합하여 최종 score를 구함

(User의 선호도 표현)



컨텐츠기반 추천시스템 Flow

- 예시: 음원 사이트
 - 신곡이 출시되면 음악을 분석하여 장르, 비트, 음색 등의 항목을 분석하여 feature 추출
 - 유저로부터 'like'를 받은 음악의 특색과 해당 유저의 프로필 준비
 - 음악의 특징과 사용자의 프로필을 바탕으로 선호하는 음악 추천
 - 위 과정을 반복하여 모델이 점차적으로 개선



컨텐츠기반 추천시스템

- 장점

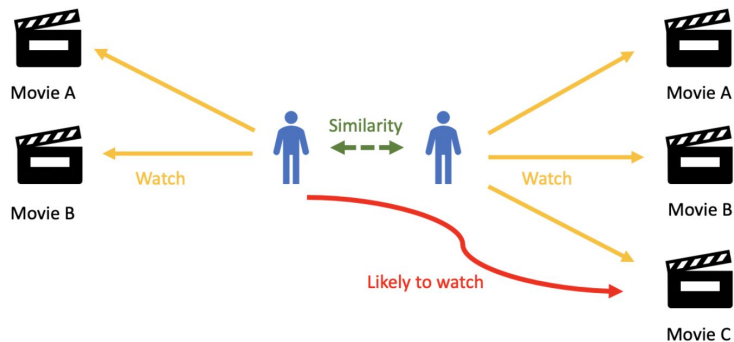
- 다른 유저가 어떤 아이템에 관심을 가졌는지 중요하지 않다
- 추천할 수 있는 아이템의 범위가 넓다
 - 100개 아이템 중 유저가 10개를 봤다면 나머지 90개에 대해서 추천 가능
 - Unique, New, Unpopular 아이템 모두 가능
 - cold start 문제 커버 가능
- 추천하는 이유를 제시할 수 있다
 - 아이템의 features로 컨텐츠 분석 가능
 - 특정 feature가 추천의 이유가 됐다고 설명 가능 (유사도가 높은 것을 추천하기 때문)

- 단점

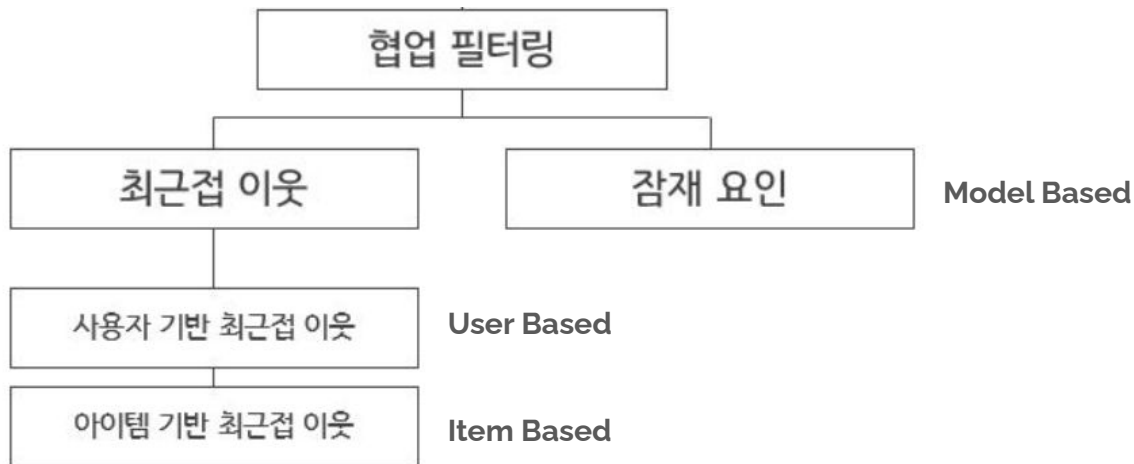
- 적절한 **features**를 찾기 어렵다
 - 수 많은 **feature**들 중에서 성능을 높일 수 있을 조합을 찾는게 어려움
- 선호하는 특성을 가진 항목을 반복 추천한다
 - **Overspecialization**
 - 유저 프로필외 추천 불가 (다양한 취향 반영 어려움)
 - **EX)** 특정 유저가 100개의 영화 중 50개에 평가를 했는데 다 스릴러물 이라면 로맨스물 추천을 할 수 없음
- 아이템의 잠재적 의미를 활용하기 어렵다
 - 눈에 보이는 특징만 활용 가능

협업 필터링

- 협동하여 필터링 (집단 지성)
- 다른 유저의 정보까지 종합적으로 고려하여 추천
 - 나와 비슷한 과거, 선호를 지닌 사람은 현재에도 나와 비슷한 아이템을 좋아할 것이다



협업 필터링





Neighborhood-based Collaborative Filtering

- Memory-based Collaborative Filtering
- User-Item 간의 평점 등 주어진 데이터로 새로운 아이템 예측
- 모델을 학습시키는게 아닌 유사도 기반으로 동작
- User-based Collaborative Filtering
 - 두 사용자가 얼마나 유사한 아이템을 좋아했는지 바탕으로 추천
 - 취향이 비슷한 사용자끼리의 데이터를 바탕으로 추천
- Item-based Collaborative Filtering
 - 과거 아이템 선호도 데이터를 기반으로 상호 연관성 높은 다른 아이템 추천
 - 콘텐츠 기반 추천시스템과 다름

Neighborhood-based Collaborative Filtering

Table 2.1: User-user similarity computation between user 3 and other users

Item-Id ⇒ User-Id ↓	1	2	3	4	5	6	Mean Rating	Cosine($i, 3$) (user-user)	Pearson($i, 3$) (user-user)
1	7	6	7	4	5	4	5.5	0.956	0.894
2	6	7	?	4	3	4	4.8	0.981	0.939
3	?	3	3	1	1	?	2	1.0	1.0
4	1	2	2	3	3	4	2.5	0.789	-1.0
5	1	?	1	2	3	3	2	0.645	-0.817

User 3의 예측점수

- 가장 가까운 User 2의 평점으로 예측 가능
- 가장 가까운 top N의 평점의 가중치 평균으로 예측 가능

$$\hat{r}_{31} = \frac{7 * 0.894 + 6 * 0.939}{0.894 + 0.939} \approx 6.49$$

$$\hat{r}_{36} = \frac{4 * 0.894 + 4 * 0.939}{0.894 + 0.939} = 4$$

Item-based CF는 위의 방식을 Item 기준으로 진행



Neighborhood-based Collaborative Filtering

- 정확도
 - 아이템 수가 더 많을 경우 -> User-based / 유저 수가 더 많을 경우 -> Item-based
 - 수가 더 적은 쪽의 벡터가 덜 sparse 할 것이므로
- 설명력
 - 비슷한 아이템, 유저의 가중치와 함께 설명 가능
 - 그러나 User-based의 경우 특정 유저와 비슷한 유저로 분류된 유저의 실제 취향을 알 수 없는 부분이 있음
- 새로운 추천
 - User-based가 더욱 다양한 유저의 데이터를 보기 때문에 상대적으로 더 새로운 추천 가능
 - 아이템의 종류는 도메인에 따라 한정되어 있어 데이터의 다양성이 상대적으로 부족
- Cold-start 문제
 - 유저에 대한 아무런 기록이 없다면, 새로운 아이템에 대한 정보가 없다면 추천 불가능
- Long-Tail Economy
 - 관심이 상대적으로 부족한 아이템은 추천되지 못함

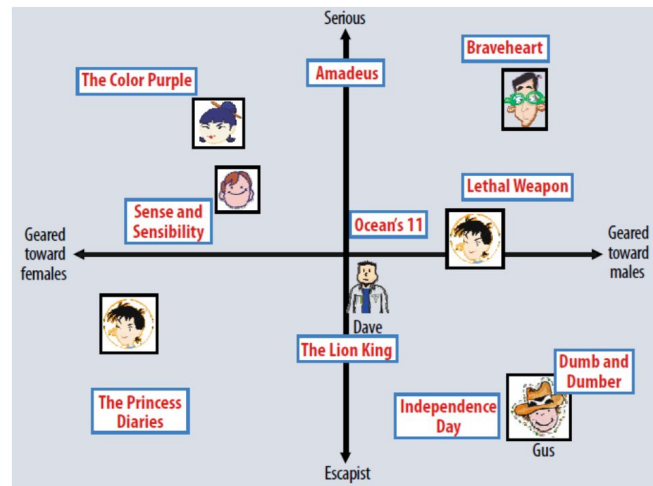


Model-based Collaborative Filtering

- 주어진 데이터를 활용하여 모델 학습
- 유사성을 이용하는 것이 아닌 데이터의 패턴을 학습
- 데이터의 잠재적 특성(취향)을 파악하는 모델링(Latent Factor Model) 방식이 유명
- 종류
 - Matrix Factorization (Latent Factor 기반)
 - Probabilistic Models
 - Deep Learning 기반
 - ML 기반 모델
 - Association Rule 기반
 - 등등

Latent Factor Model

- User-Item 행렬에서 사용자와 아이템을 factor의 벡터로 표현하는 방법
- 사용자와 아이템을 같은 벡터 공간에 표현 (Latent space 상에 mapping)
- 같은 벡터 공간에서 사용자와 아이템이 가까우면 유사, 멀리 떨어져 있으면 유사하지 않음

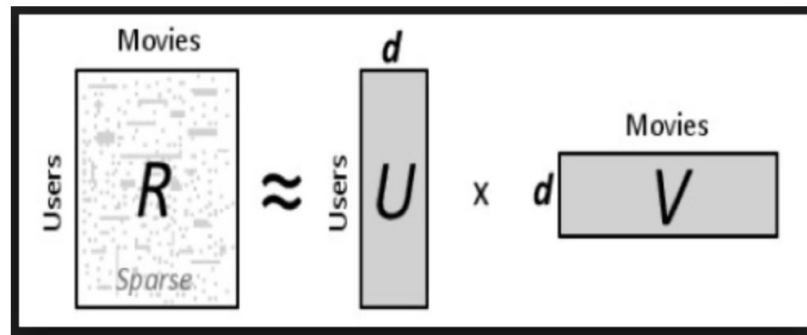


[Example] 이 표는 잠재 요인 모델 접근방법에 대한 설명을 나타낸 표로 남자 대 여자, Serious 대 Escapist-두 축을 사용하여 사용자와 영화들 모두의 특성들을 표현했다.

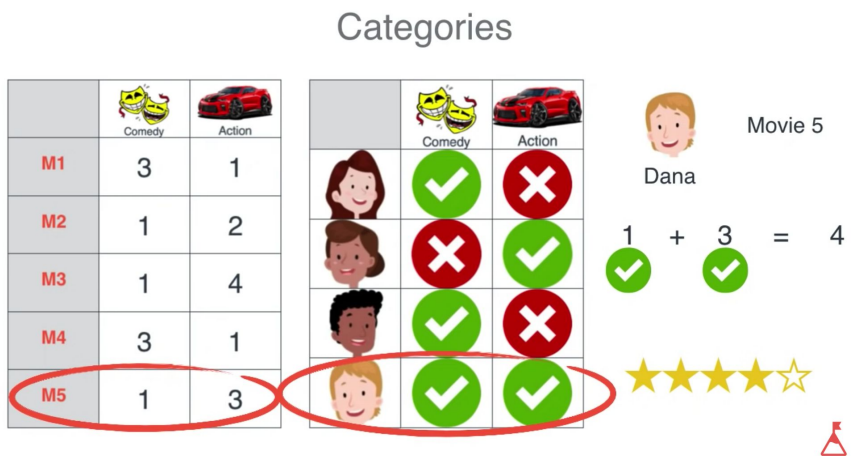
Matrix Factorization

- Rating Matrix를 분해하는 과정
- R: Rating Matrix (Users * Movies)
- U(P): User latent matrix
- V(Q): Item latent matrix
- d: latent factor (임의의 차원 수)
- 목표는 R hat을 만들어 전체 데이터를 추정하는 것
- 관측된 rating data를 사용하여 R과 R hat이 서로 유사하도록 latent factor를 학습

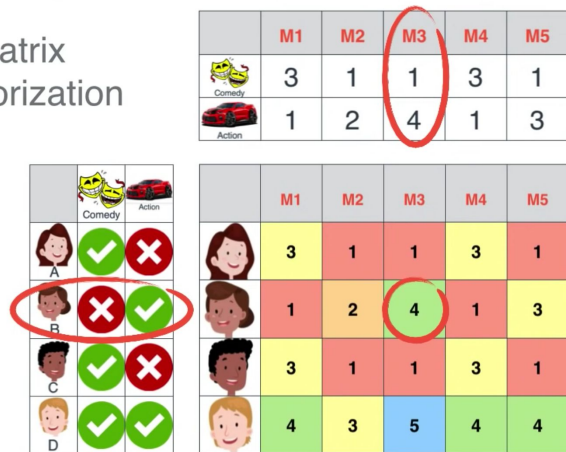
$$R \approx P \times Q^T = \hat{R}$$



Matrix Factorization



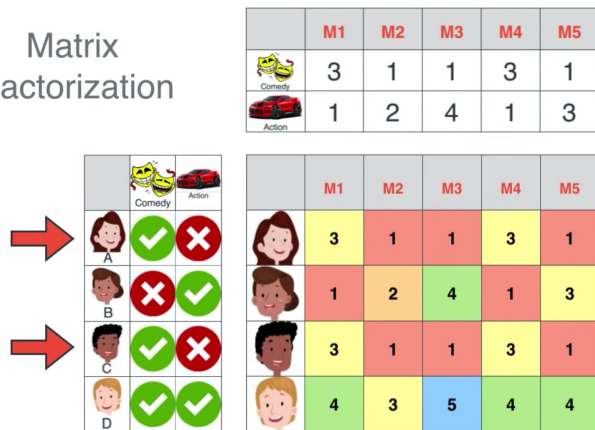
Matrix Factorization



latent factor (위 영상에서는 장르)를 이용하여 원래 rating matrix를 추정할 수 있음

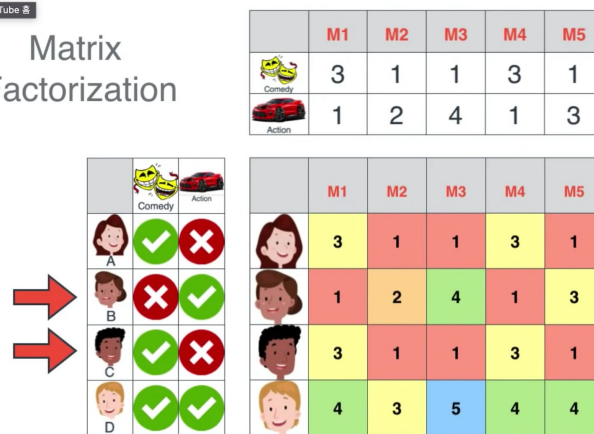
Matrix Factorization

Matrix Factorization



YouTube

Matrix Factorization



유사한 latent factor를 가진 유저는 비슷한 rating을 갖고 반대의 latent factor를 가진 유저는 반대의 rating을 갖게 됨

Matrix Factorization

	M1	M2	M3	M4	M5
	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Matrix
Factorization

training 과정 (20:42 ~ 30:12)

<https://www.youtube.com/watch?v=ZspR5PZemcs>



협업 필터링

- 장점

- 아이템의 종류와 관계없이 추천의 성능이 보장된다
- rating 정보만 이용하므로 다른 feature가 필요 없다

- 단점

- cold-start 문제
 - 한번도 평가되지 않은 아이템을 추천할 확률이 적다
 - 초기(데이터가 부족한) 유저에 대한 추천 품질이 떨어진다
- Long tail
 - 인기있는 아이템만 추천하는 경향이 있다
- latent factor에 대한 해석이 어렵다



논문 읽어보기

- 추천 시스템에서 딥러닝을 적용한 논문을 같이 읽어보기
 - Neural Collaborative Filtering
 - <https://arxiv.org/pdf/1708.05031.pdf>

Q&A