

Factorization Machine

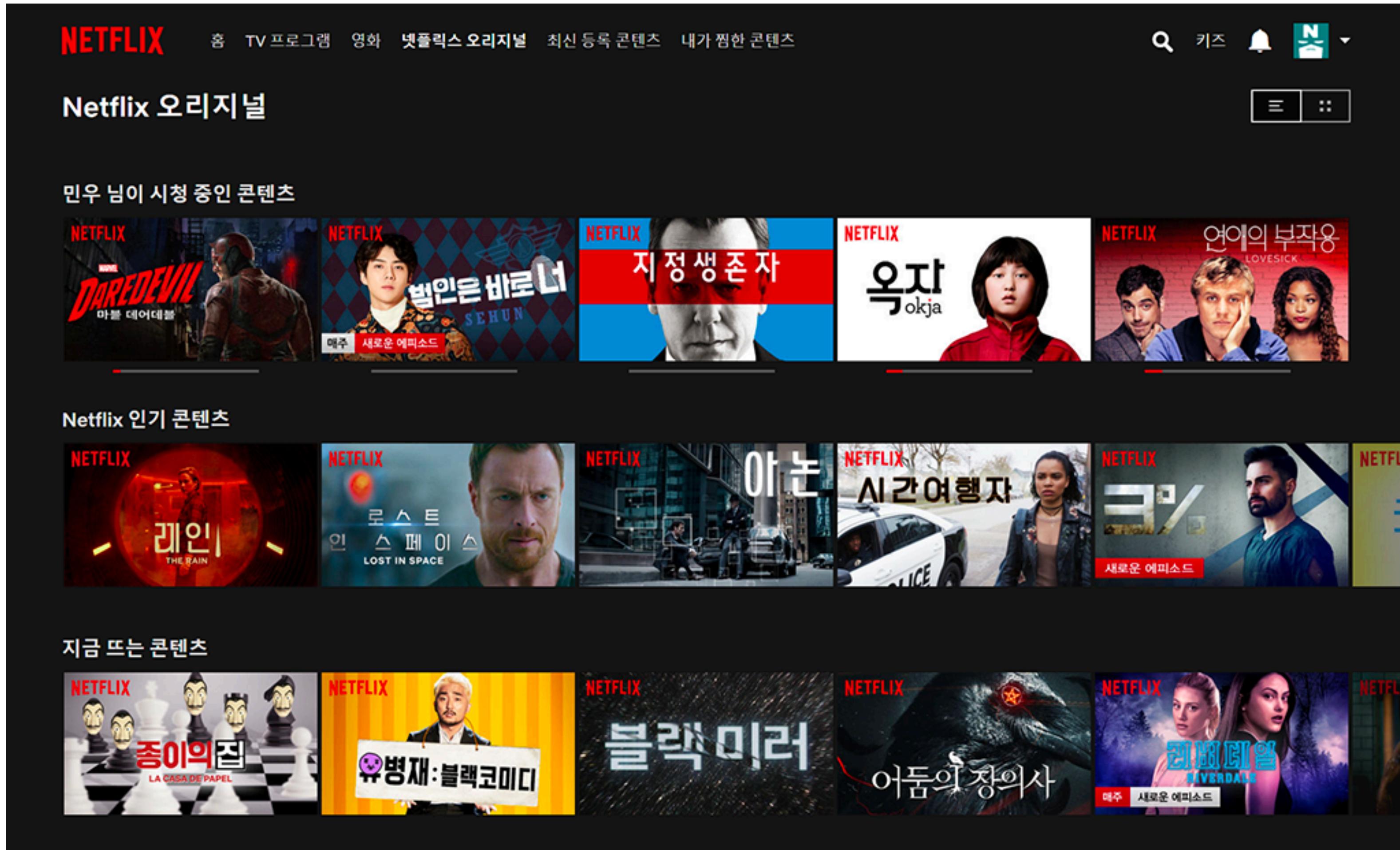
추천시스템 Overview & 논문 리뷰

2020.05.04

김태진

Overview

Netflix



Netflix

약 4200만개 컨텐츠

소비 컨텐츠의 2/3정도가
추천 로직을 경유

Overview

알수없는 유튜브 알고리즘



A screenshot of a YouTube video player showing a comment section. The comment is from a user named "니얼굴" posted 3 months ago, stating "아무도 이 영상을 검색하지 않았다." (No one searched for this video). It has 5.2k likes and 74 replies. Below the comment, there is a snippet of the video showing a mole in the soil.

Youtube

나도 모르게 보고 있는
이상한 상황

Recommender System

A recommender system, or a recommendation system is a subclass of **information filtering system** that seeks to predict the "rating" or "preference" a user would give to an item.

Recommender System

목표

Consumer

사용자의 취향, 선호도 파악

행동 기록이나,
메타데이터를 바탕으로
좋아할 만한 상품을 제공

Producer

목표 *KPI 달성

Page View 증가
Unique Visitor 증가
매출 증가

*KPI(Key Performance Indicator): 핵심 성과 지표

Recommender System

속성 구분

Best

Popularity가
높은 상품

ex) 음악 스트리밍 TOP100

Related

주어진 아이템과
연관이 높은 상품

ex) 보온재, 대체재,
이것도 같이 사면 좋아요

Personalized

개개인 별 선호도가
높은 상품

ex) 나와 비슷한 사람들이
많이 본 제품이에요

Recommender System

알고리즘 구분

	특징	Pros	Cons
Content-based filtering	아이템 속성기반 사용자 행동기록 X ex) 임베딩, TF-IDF, Category	No Item info. Cold Start	Low Accuracy
Collaborative filtering	사용자 행동기록 기반 아이템 메타데이터 X ex) 클릭로그, 평점, 구독	Better Acc	Cold Start Problem

Recommender System

Fusion

**Content-based
Filtering**

**Collaborative
Filtering**

Best

Popularity
높은 아이템

Related

해당 아이템과
같은 카테고리,
같은 속성의 상품

해당 아이템과
같이 조회, 구매한
기록이 있는 상품

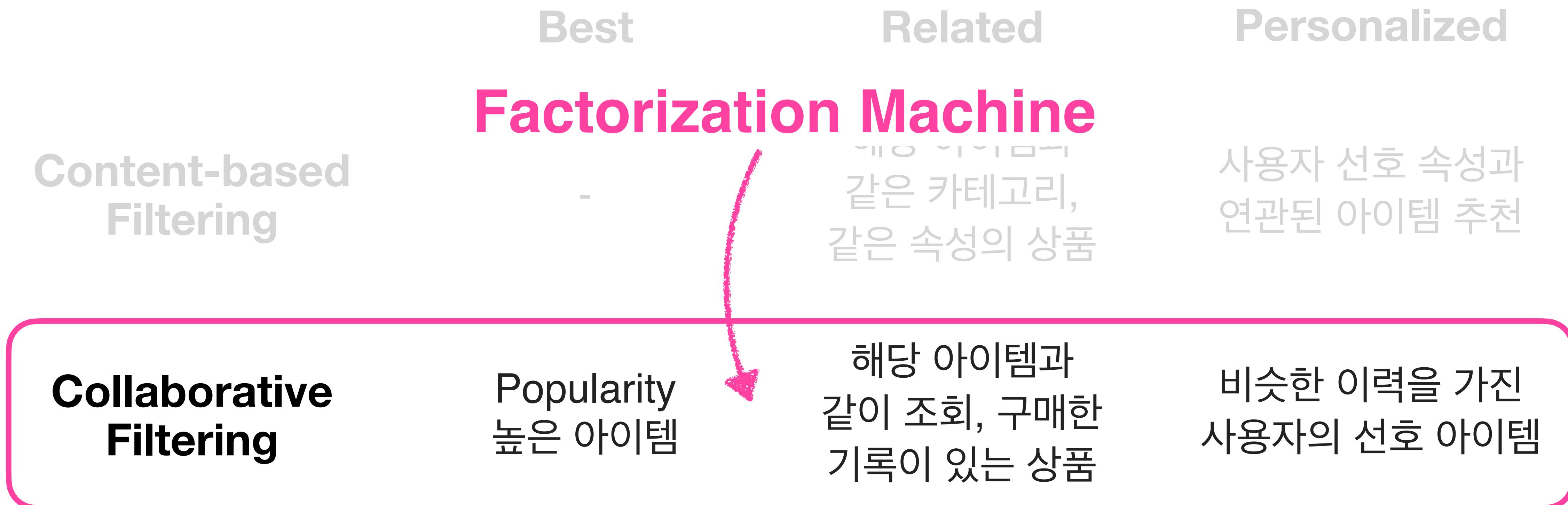
Personalized

사용자 선호 속성과
연관된 아이템 추천

비슷한 이력을 가진
사용자의 선호 아이템

Recommender System

Fusion



Factorization Machine

2010년 논문

Collaborative Filtering 방법 중 하나

Matrix Factorization + Regression 방식을 조합한 방법

Factorization Machine

소개

Input
User, Item
+ Side Features

Feature vector \mathbf{x}										Target y										
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...
					Movie					Other Movies rated					Last Movie rated					

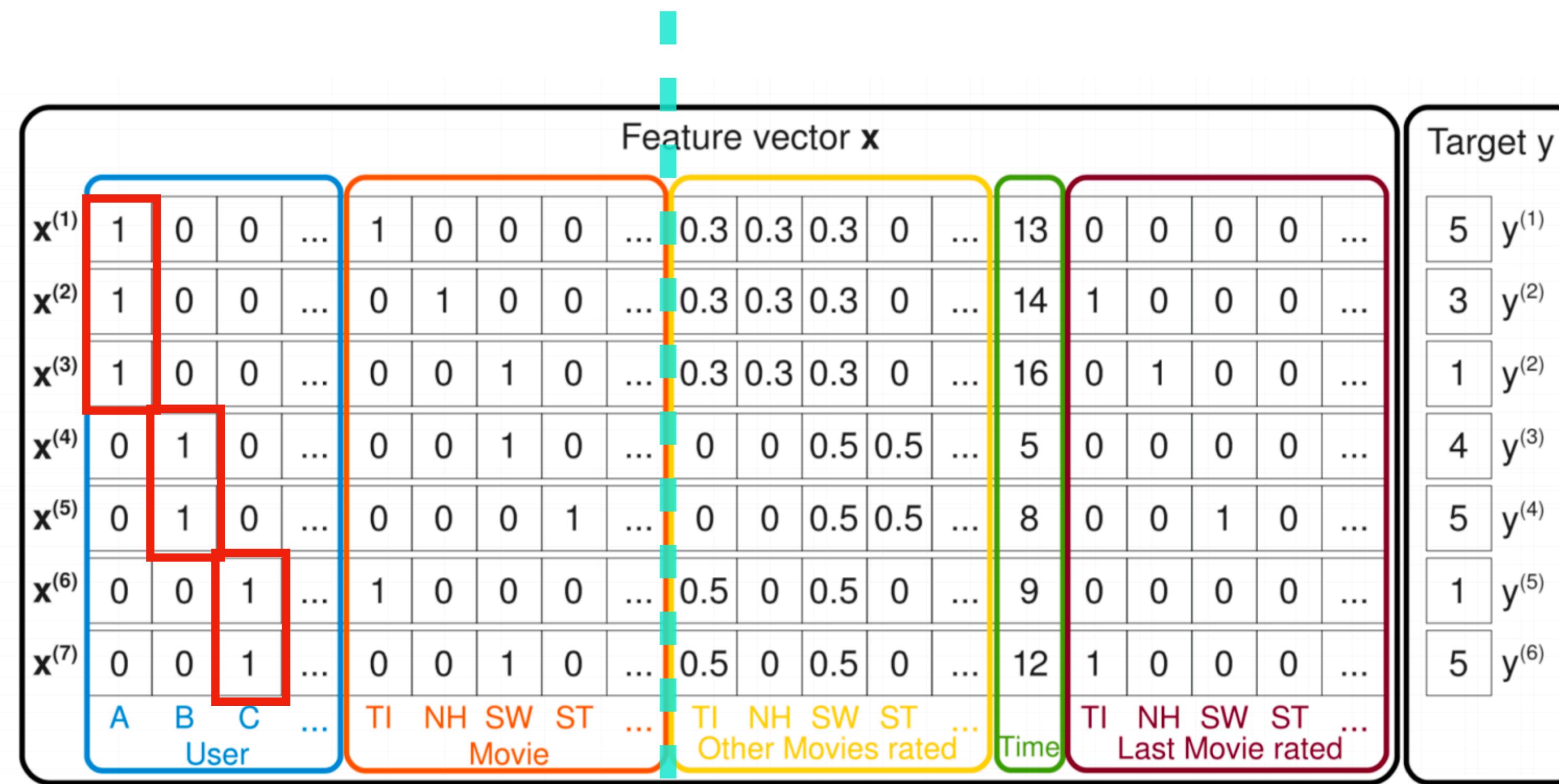
Target
Click
Rating

Dataset

User Interaction log

Factorization Machine

구성



User

Item

기본 요소

Side Features

- Category, Search Keyword, etc..

- Factorization Machine의 특징 중 하나

Factorization Machine

수식

	Feature vector \mathbf{x}																			
	User				Movie				Other Movies rated				Time	Last Movie rated						
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...

	Target y
5	$y^{(1)}$
3	$y^{(2)}$
1	$y^{(2)}$
4	$y^{(3)}$
5	$y^{(4)}$
1	$y^{(5)}$
5	$y^{(6)}$

Key Point!

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

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{V} \in \mathbb{R}^{n \times k} \quad (2)$$

w_0 : global bias

w_i : i-th variable의 weight

$\langle \mathbf{v}_i, \mathbf{v}_j \rangle$: i-th & j-th variable의 Interaction

k : Dimensionality of Factorization

Factorization Machine

Pairwise Interactions

And $\langle \cdot, \cdot \rangle$ is the dot product of two vectors of size k :

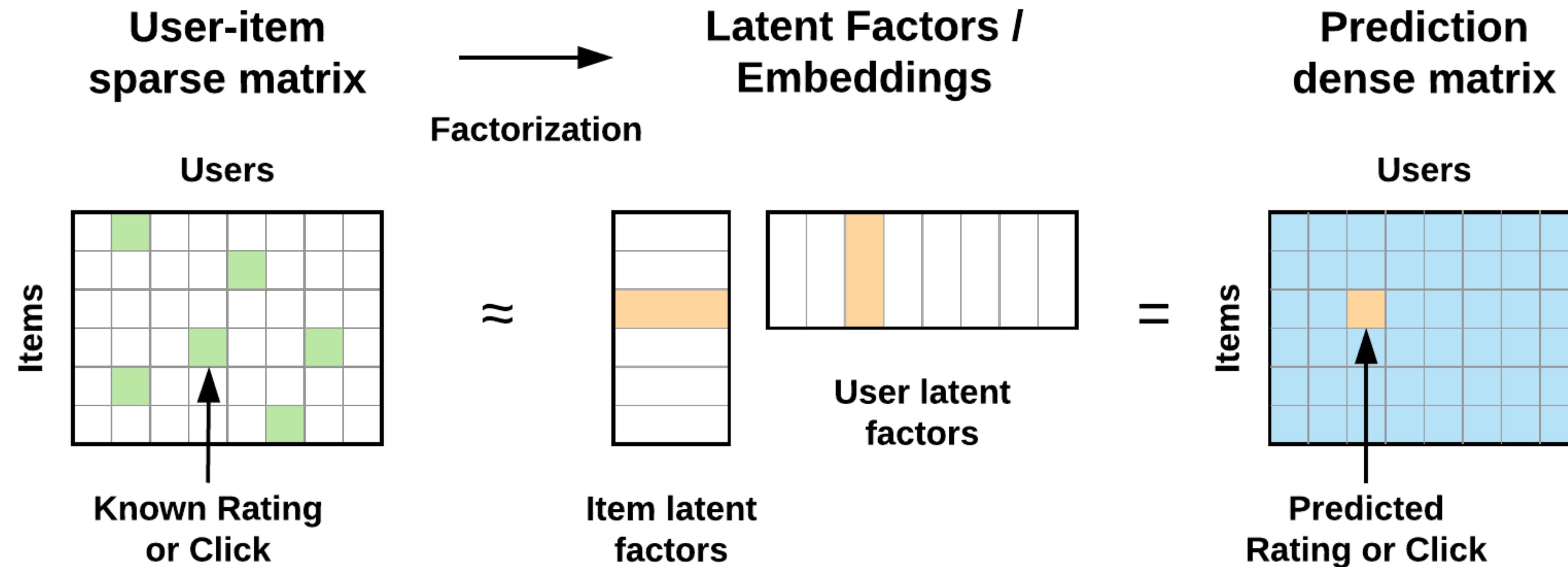
$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^k v_{i,f} \cdot v_{j,f} \quad (3)$$

$$w_{i,j} \in \mathbb{R} \quad \xrightarrow{\qquad\qquad\qquad W = V \cdot V^t \qquad\qquad\qquad} \quad \mathbf{V} \in \mathbb{R}^{n \times k}$$

Factorization

Extra.. Matrix Factorization

MF vs FM



A FM using this feature vector \mathbf{x} is identical to the matrix factorization model [2] because x_j is only non-zero for u and i , so all other biases and interactions drop:

$$\hat{y}(\mathbf{x}) = w_0 + w_u + w_i + \langle \mathbf{v}_u, \mathbf{v}_i \rangle \quad (12)$$

Parameter Estimation Under Sparsity

Sparse한 환경에서 FM의 능력

Feature vector \mathbf{x}							Target y
A	$x^{(1)}$	1 0 0 ...	1 0 0 0 ...	0.3 0.3 0.3 0 ...	13	0 0 0 0 ...	5 $y^{(1)}$
	$x^{(2)}$	1 0 0 ...	0 1 0 0 ...	0.3 0.3 0.3 0 ...	14	1 0 0 0 ...	3 $y^{(2)}$
	$x^{(3)}$	1 0 0 ...	0 0 1 0 ...	0.3 0.3 0.3 0 ...	16	0 1 0 0 ...	1 $y^{(2)}$
	$x^{(4)}$	0 1 0 ...	0 0 1 0 ...	0 0 0.5 0.5 ...	5	0 0 0 0 ...	4 $y^{(3)}$
	$x^{(5)}$	0 1 0 ...	0 0 0 1 ...	0 0 0.5 0.5 ...	8	0 0 1 0 ...	5 $y^{(4)}$
	$x^{(6)}$	0 0 1 ...	1 0 0 0 ...	0.5 0 0.5 0 ...	9	0 0 0 0 ...	1 $y^{(5)}$
	$x^{(7)}$	0 0 1 ...	0 0 1 0 ...	0.5 0 0.5 0 ...	12	1 0 0 0 ...	5 $y^{(6)}$
User		A B C ...	TI NH SW ST ...	TI NH SW ST ...	Time	TI NH SW ST ...	Last Movie rated
Movie							
Other Movies rated							

보통 **sparse**한 환경에서는 Feature간 Interaction이 매우 **독립적**,

Pairwise interaction으로 각 feature간 독립성을 깨버림

Computation

천재인가

Time Complexity

Training

Gradient Decent Method

$$\frac{\partial}{\partial \theta} \hat{y}(\mathbf{x}) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \underbrace{\sum_{j=1}^n v_{j,f} x_j - v_{i,f} x_i^2}_{\text{Precomputed}}, & \text{if } \theta \text{ is } v_{i,f} \end{cases} \quad (4)$$

- Stochastic Gradient Decent (SGD)
- Each gradient computation : $O(1)$
- All parameter update : $O(kn)$

Summary

(대충 내가 괜찮은 걸 만들었다고 표현하고 있는 문장)

Factorization Machine은 모든 Feature들의 factorize된 interaction을 사용함으로써,

- **Sparse**한 환경에서도 Interaction을 잘 추정할 수 있고
- 학습과 예측이 선형시간 안에 이루어 짐
- 그렇기에, Parameter 수도 선형적이라 SGD를 이용한 다양한 Optimization이 가능

Summary

개인적인 생각

최근 개인화 추천이 뜨면서, 2010년 논문이지만 재조명 되고 있음

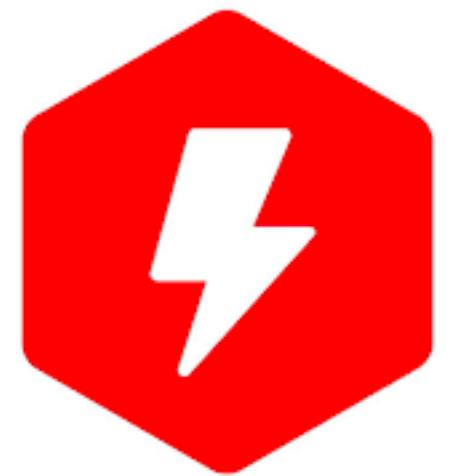
간단하면서, 작고, 빠른 모델 + Side Feature를 Input으로 넣을 수 있다는 점

**속도가 빠른건 확실하지만,
경우에 따라서, 많은 데이터 IN/OUT과 컴퓨팅 리소스를 필요로 함**

하루 이용자 수(*DAU) 50만, 아이템 수 500만 이라면?

*DAU(Daily Active User): 하루 활동 이용자 수

번개장터에서는...



Next...



Amazon Personalize

2017년 논문

Personalizing Session-based Recommendations with
Hierarchical Recurrent Neural Networks

<https://arxiv.org/pdf/1706.04148.pdf>

끝

