Neural Graph Collaborative Filtering

- SIGIR 2019 -

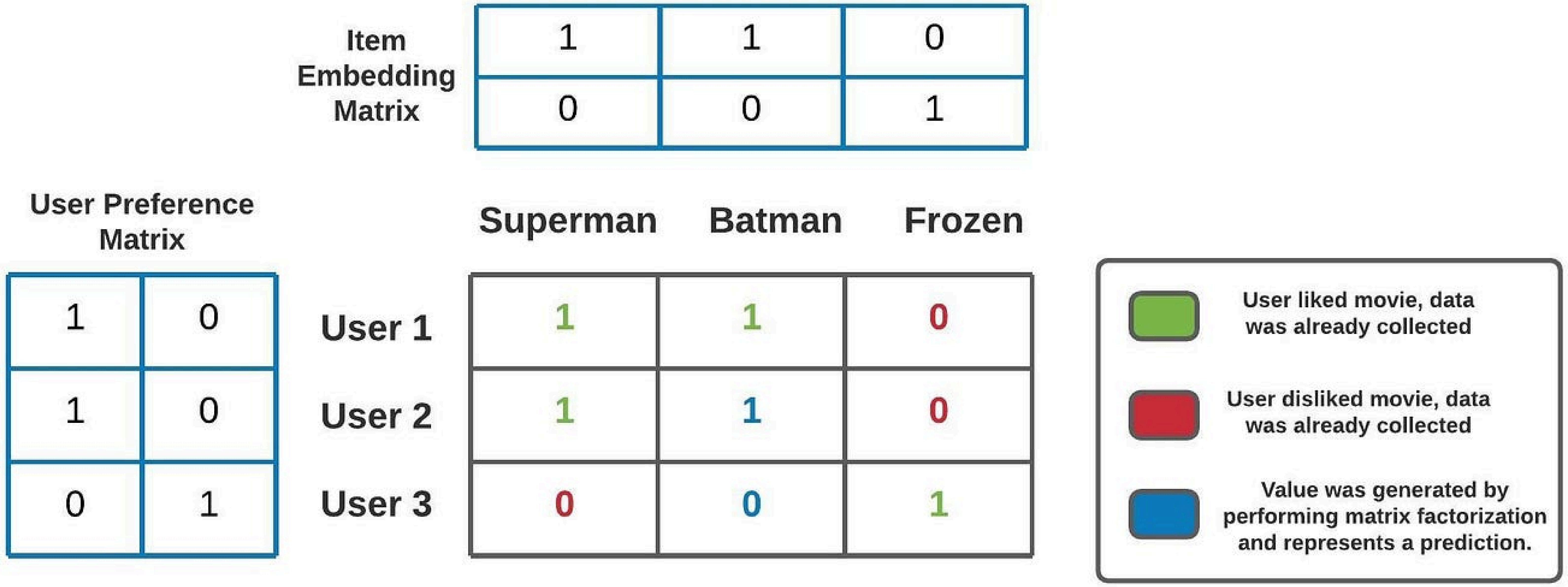
Kookmin Univ.
Se Won Hong

Index

Introduction

previous CF

초기화환 User,Item 임베딩을 연관성은 고려하지 않고 점수를 맞추기 위한 학습만 진행하여 예측



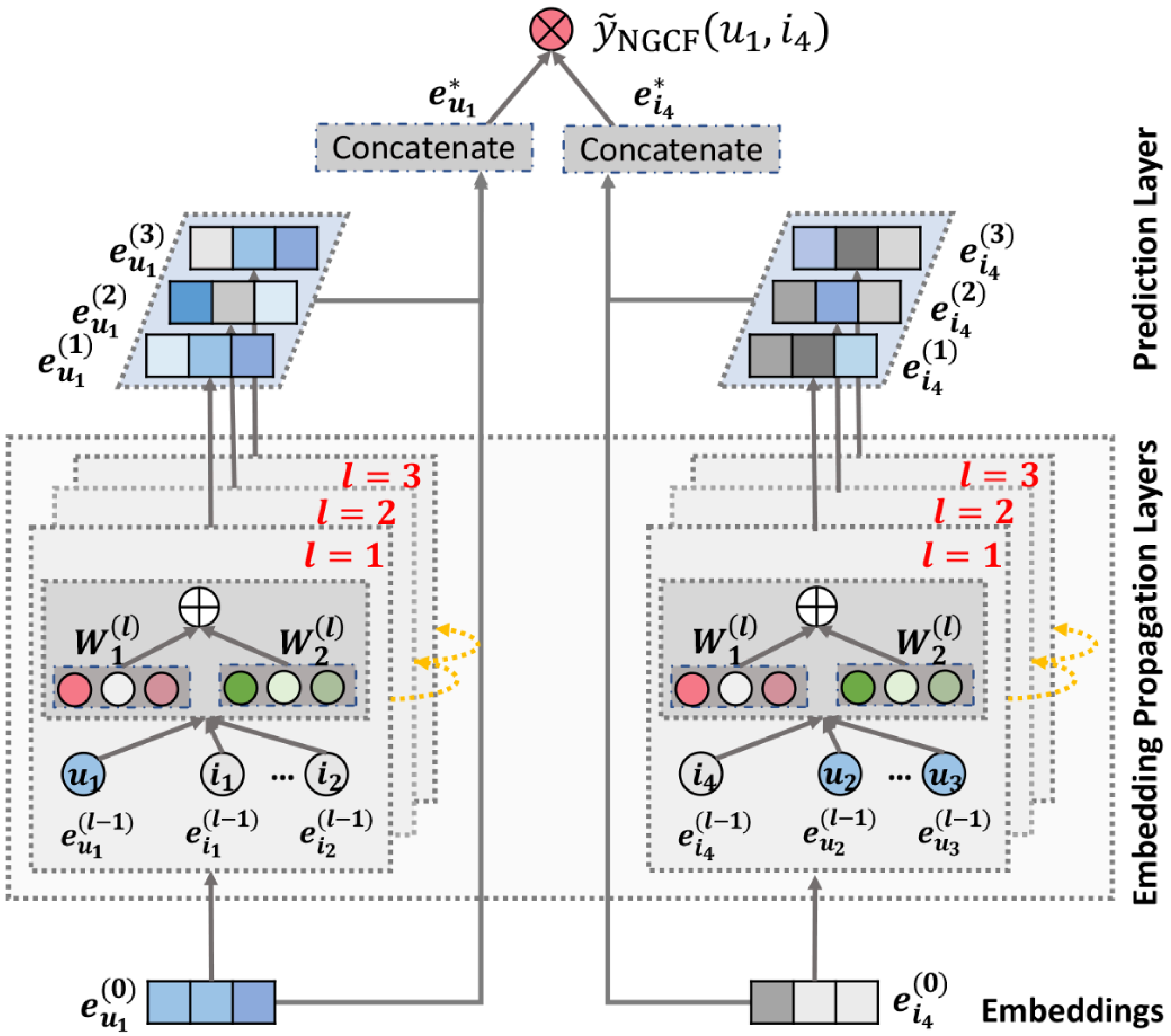
Introduction

previous CF

기존 CF는 협업 관계를 반영하지 않은
ID 기반 임베딩에 의존

Model

Model Architecture



Model

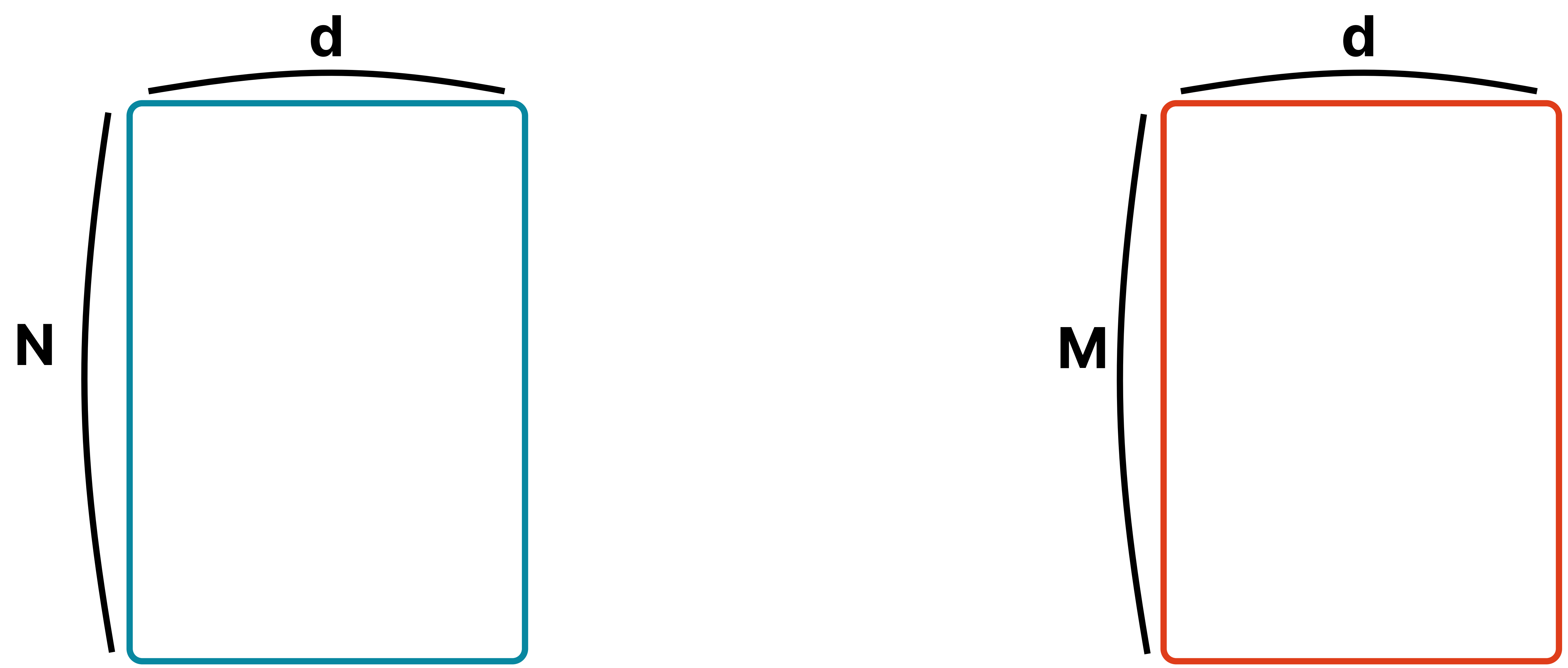
NGCF

초기 임베딩 look-up 테이블을 생성

N : User 수

M : Item 수

d : 임베딩차원



User Lookup Emb Table

Item Lookup Emb Table

Model

First-Order Propagation

- **Message Construction(메세지 생성)**

f() : Message 전달 함수

e_i : Item 임베딩

e_u : User 임베딩

N_u,N_i : 자기자신 이웃 수

p_ui : 정규화(사용자의 상호작용 얼마나 사용자의 선호에 영향을 미치는지 반영)

$$\mathbf{m}_{u\leftarrow i} = f(\mathbf{e}_i, \mathbf{e}_u, p_{ui}),$$

$$\mathbf{m}_{u\leftarrow i} = \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}\Big(\mathbf{W}_1\mathbf{e}_i + \mathbf{W}_2(\mathbf{e}_i \odot \mathbf{e}_u)\Big).$$

Model

First-Order Propagation

- **Message Aggregation(메세지 전파)**

LeakyReLU를 사용하여
Positive signal과 조금의 negative signal을 반영

m_(u<-u) : 자기자신 메세지로 W_1*e_u (W_1은 msg construction에 나온 W_1와 동일)

$$\mathbf{e}_u^{(1)} = \text{LeakyReLU}\Big(\mathbf{m}_{u \leftarrow u} + \sum_{i \in \mathcal{N}_u} \mathbf{m}_{u \leftarrow i}\Big),$$

Model

High-Order Propagation

$$\mathbf{e}_u^{(l)} = \mathrm{LeakyReLU}\Big(\mathbf{m}_{u\leftarrow u}^{(l)} + \sum_{i\in\mathcal{N}_u} \mathbf{m}_{u\leftarrow i}^{(l)}\Big),$$

$$\begin{cases} \mathbf{m}_{u\leftarrow i}^{(l)} = p_{ui}\Big(\mathbf{W}_1^{(l)}\mathbf{e}_i^{(l-1)} + \mathbf{W}_2^{(l)}(\mathbf{e}_i^{(l-1)} \odot \mathbf{e}_u^{(l-1)})\Big), \\ \mathbf{m}_{u\leftarrow u}^{(l)} = \mathbf{W}_1^{(l)}\mathbf{e}_u^{(l-1)}, \end{cases}$$

Model

High-Order Propagation

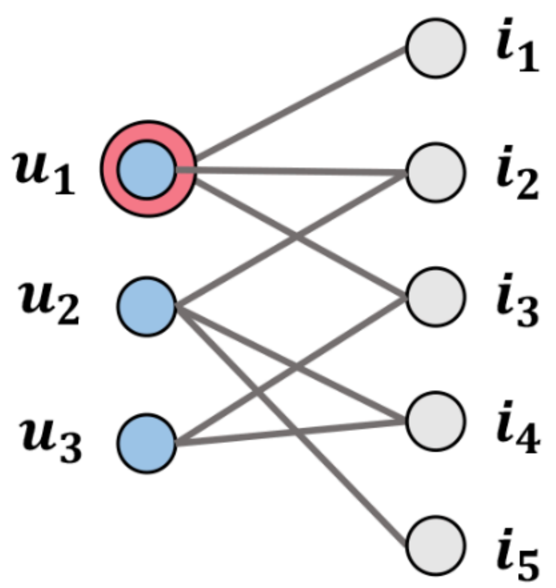
- 전체 임베딩에 대해 아래와 같은 함수로 표현 가능
- 행렬 계산을 통해 전체 임베딩을 효율적으로 계산 가능

$$\mathbf{E}^{(l)} = \mathrm{LeakyReLU}\left((\mathcal{L} + \mathbf{I})\mathbf{E}^{(l-1)}\mathbf{W}_1^{(l)} + \mathcal{L}\mathbf{E}^{(l-1)} \odot \mathbf{E}^{(l-1)}\mathbf{W}_2^{(l)} \right)$$

Model

High-Order Propagation

- 그래프로 예시 들기



$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \text{ and } \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^{\top} & \mathbf{0} \end{bmatrix}$$

$$A = \begin{array}{cc} & \begin{array}{cc} \overbrace{\qquad\quad}^{N} & \overbrace{\qquad\qquad\qquad}^{M} \end{array} \\ \begin{array}{c} N \\ \\ M \end{array} & \begin{array}{cc} \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & \boxed{\begin{matrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{matrix}}\ R \\ \underbrace{\boxed{\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{matrix}}}_{R^T} & \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \end{array} \end{array}$$

$$D = \begin{matrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{matrix}$$

A : Adjacency matrix

D : diagonal matrix

Model

High-Order Propagation

- 그래프로 예시 들기
- L로 아래의 정규화를 행렬로 만들 수 있음

$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \text{ and } \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^{\top} & \mathbf{0} \end{bmatrix}$$

$$\begin{matrix}
0 & 0 & 0 & \frac{1}{\sqrt{3}}\frac{1}{\sqrt{1}} & \frac{1}{\sqrt{3}}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}}\frac{1}{\sqrt{2}} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{\sqrt{3}}\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{3}}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}}\frac{1}{\sqrt{1}} \\
0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} & 0 \\
0 & 0 & \frac{1}{\sqrt{1}}\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{\sqrt{2}}\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{\sqrt{2}}\frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{\sqrt{2}}\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}}\frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{\sqrt{1}}\frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}$$

$$\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}$$

Prediction

Embedding Concat

- 여러 hop(layer)에서 나온 임베딩들을 concat하여 결과 예측에 사용
- LSTM, max Pooling 등 여러 집계함수를 사용해도 되지만, Concat의 성능이 제일 우수함

$$\mathbf{e}_u^* = \mathbf{e}_u^{(0)} \| \cdots \| \mathbf{e}_u^{(L)}, \quad \mathbf{e}_i^* = \mathbf{e}_i^{(0)} \| \cdots \| \mathbf{e}_i^{(L)},$$

Prediction

Score Prediction

- 앞서 concat한 User, Item임베딩들을 내적하여 Score 예측값 도출

$$\hat{y}_{\mathrm{NGCF}}(u, i) = \mathbf{e}_u^{* \top} \mathbf{e}_i^{*}.$$

Prediction

Object Loss

- **BPR Loss 사용**
- **BPR Loss란?**
  - **상호작용한 아이템(Positive)은 상호작용하지 않은 아이템(Negative)보다 예측점수가 높아야한다**

u : User

i : u User가 관측한 Item(pos)

j : u User가 관측하지 않은 Item(neg)

σ : 시그모이드 함수(0~1 확률로 변환)

Θ : 러닝 파라미터 (W, item/user Embeddings)

λ : L2 정규화

$$Loss = \sum_{(u,i,j)\in O} -\ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda\,\|\Theta\|_2^2$$

Experiments

Table 2: Overall Performance Comparison.

	Gowalla		Yelp2018*		Amazon-Book	
	recall	ndcg	recall	ndcg	recall	ndcg
MF	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
NeuMF	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
CMN	<u>0.1405</u>	<u>0.1221</u>	0.0457	0.0369	0.0267	0.0218
HOP-Rec	0.1399	0.1214	<u>0.0517</u>	<u>0.0428</u>	<u>0.0309</u>	<u>0.0232</u>
GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219
NGCF-3	**0.1569***	**0.1327***	**0.0579***	**0.0477***	**0.0337***	**0.0261***
%Improv.	11.68%	8.64%	11.97%	11.29%	9.61%	12.50%
p-value	2.01e-7	3.03e-3	5.34e-3	4.62e-4	3.48e-5	1.26e-4