GraphRec(Graph Neural Networks for Social Recommendation)

이은경

목차

1. Introduction

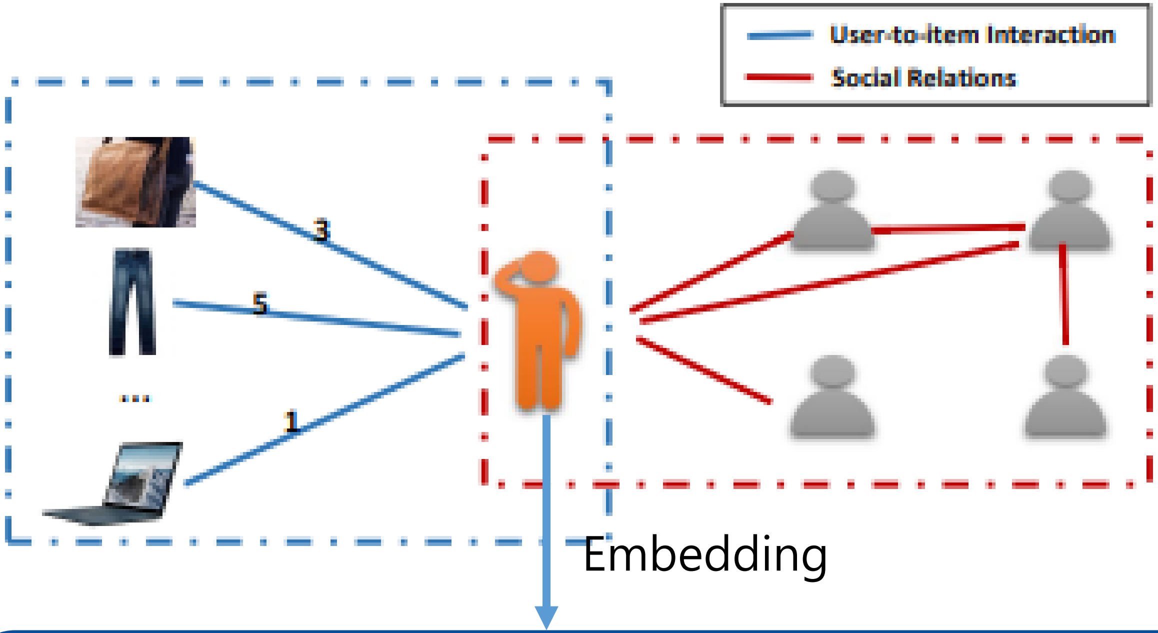


Figure 1: Graph Data in Social Recommendation. It contains two graphs including the user-item graph (left part) and the user-user social graph (right part). Note that the number on the edges of the user-item graph denotes the opinions (or rating score) of users on the items via the interactions.

핵심 : 두 graph의 정보 집계
1) 두 graph를 본직적 결합
2) User-item 간 상호작용, 의견 동시포착
3) 사회적 관계와 heterogeneous strength 구분법

기여

social recommendations 의 graph data를 일관성 있게 모델링할 수 있는 새로운 graph Neural Network GraphRec을 제안

user-item graph 에서 interactions 과 opinions을 capture하는 principled approach 제공

social relations의 heterogeneous strengths를 고려하는 방법 도입

various real-world datasets 에서 the proposed framework의 effectiveness를 입증

2. THE PROPOSED FRAMEWORK

1) Definitions and Notations

Table 1: Notation

Symbols	Definitions and Descriptions
r_{ij}	The rating value of item v_j by user u_i
$\mathbf{q}_j$	The embedding of item v_j
$\mathbf{p}_i$	The embedding of user u_i
$\mathbf{e}_r$	The opinion embedding for the rating level r, such as 5-star rating, $r \in \{1,2,3,4,5\}$
d	The length of embedding vector
$C(i)$	The set of items which user u_i interacted with
$N(i)$	The set of social friends who user u_i directly connected with
$B(j)$	The set of users who have interacted the item v_j
$\mathbf{h}_i^I$	The item-space user latent factor from item set $C(i)$ of user u_i
$\mathbf{h}_i^S$	The social-space user latent factor from the social friends $N(i)$ of user u_i
$\mathbf{h}_i$	The user latent factor of user u_i, combining from item space $\mathbf{h}_i^I$ and social space $\mathbf{h}_i^S$
$\mathbf{x}_{ia}$	The opinion-aware interaction representation of item v_a for user u_i
$\mathbf{f}_{jt}$	The opinion-aware interaction representation of user u_t for item v_j
$\mathbf{z}_j$	The item latent factor of item v_j
α_{ia}	The item attention of item v_a in contributing to $\mathbf{h}_i^I$
β_{io}	The social attention of neighboring user u_o in contributing to $\mathbf{h}_i^S$
μ_{jt}	The user attention of user u_t in contributing to $\mathbf{z}_j$
r'_{ij}	The predicted rating value of item v_j by user u_i
$\oplus$	The concatenation operator of two vectors
$\mathbf{T}$	The user-user social graph
$\mathbf{R}$	The user-item rating matrix (user-item graph)
$\mathbf{W}, \mathbf{b}$	The weight and bias in neural network

2) An Overview of the Proposed Framework

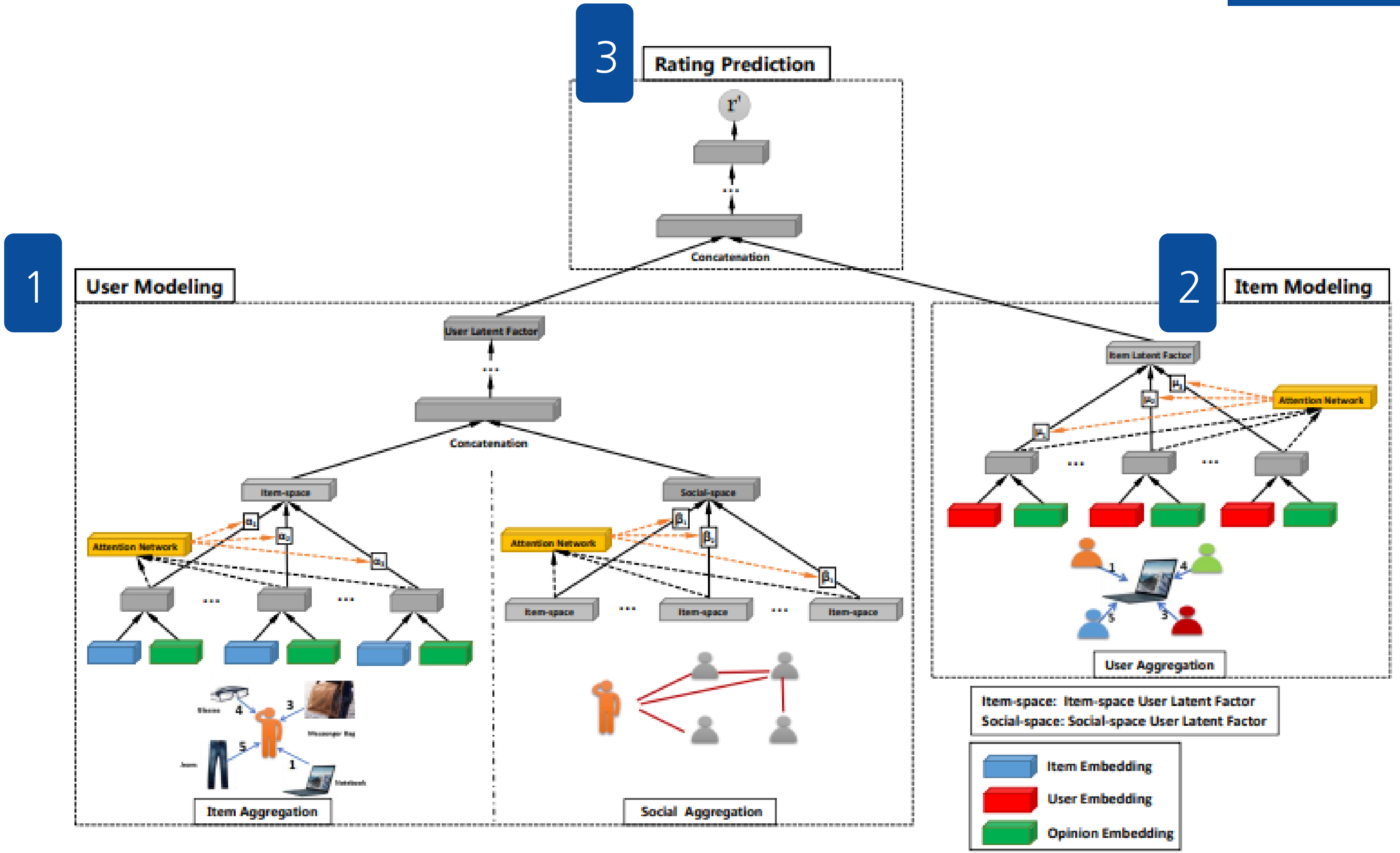
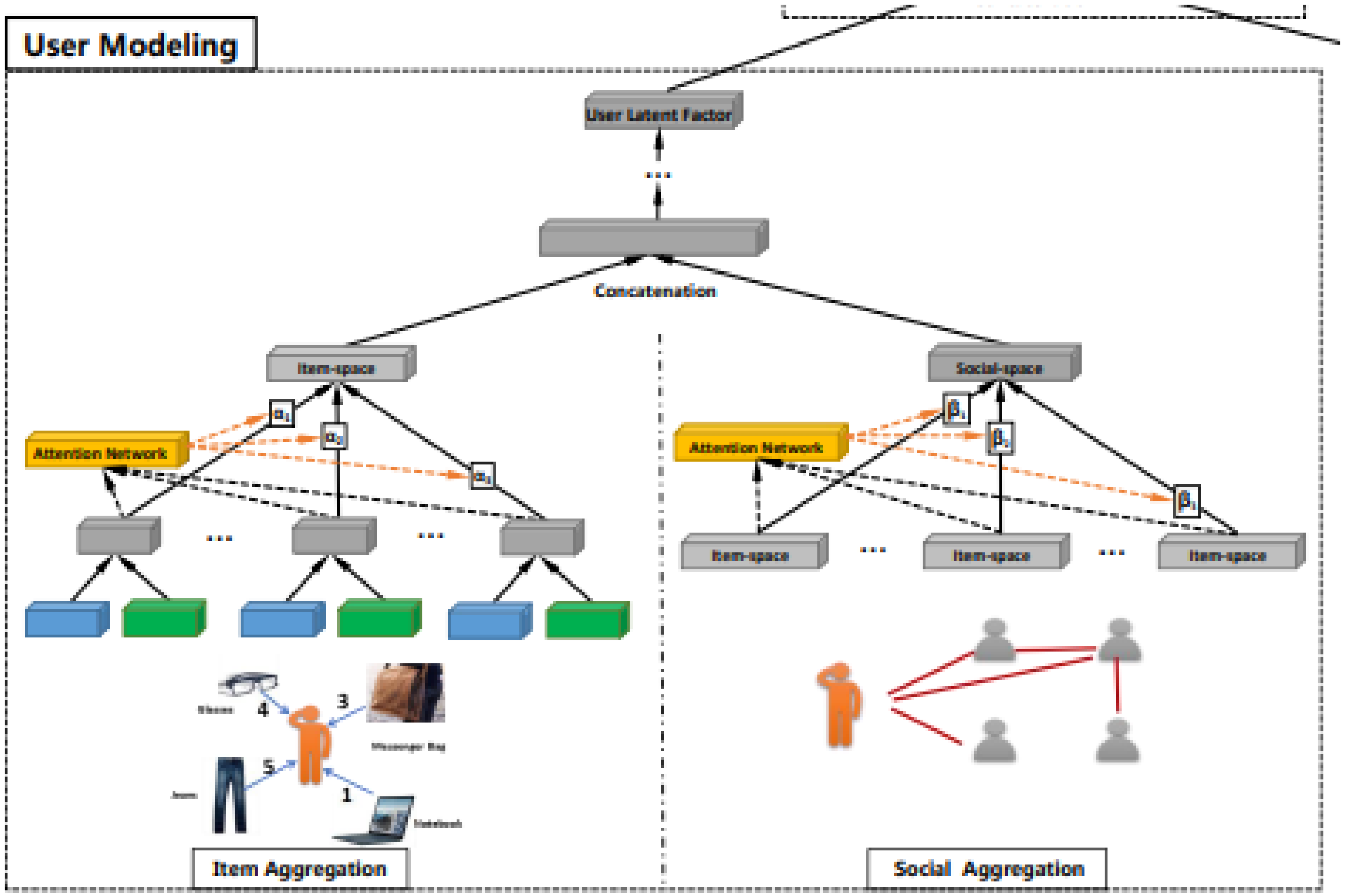


Figure 2: The overall architecture of the proposed model. It contains three major components: user modeling, item modeling, and rating prediction.

3) User Modeling

1



- User's latent factors를 학습하는 모델링
- user i 에 대한 latent factors $h_i \in \mathbb{R}^{d}$ 학습
- item graph와 social graph를 본질적으로 결합하는 방법

〈 Aggregation 〉

1) user – item graph 의 $h_i^I \in \mathbb{R}^{d}$ item-space user latent factor를 학습시 활용

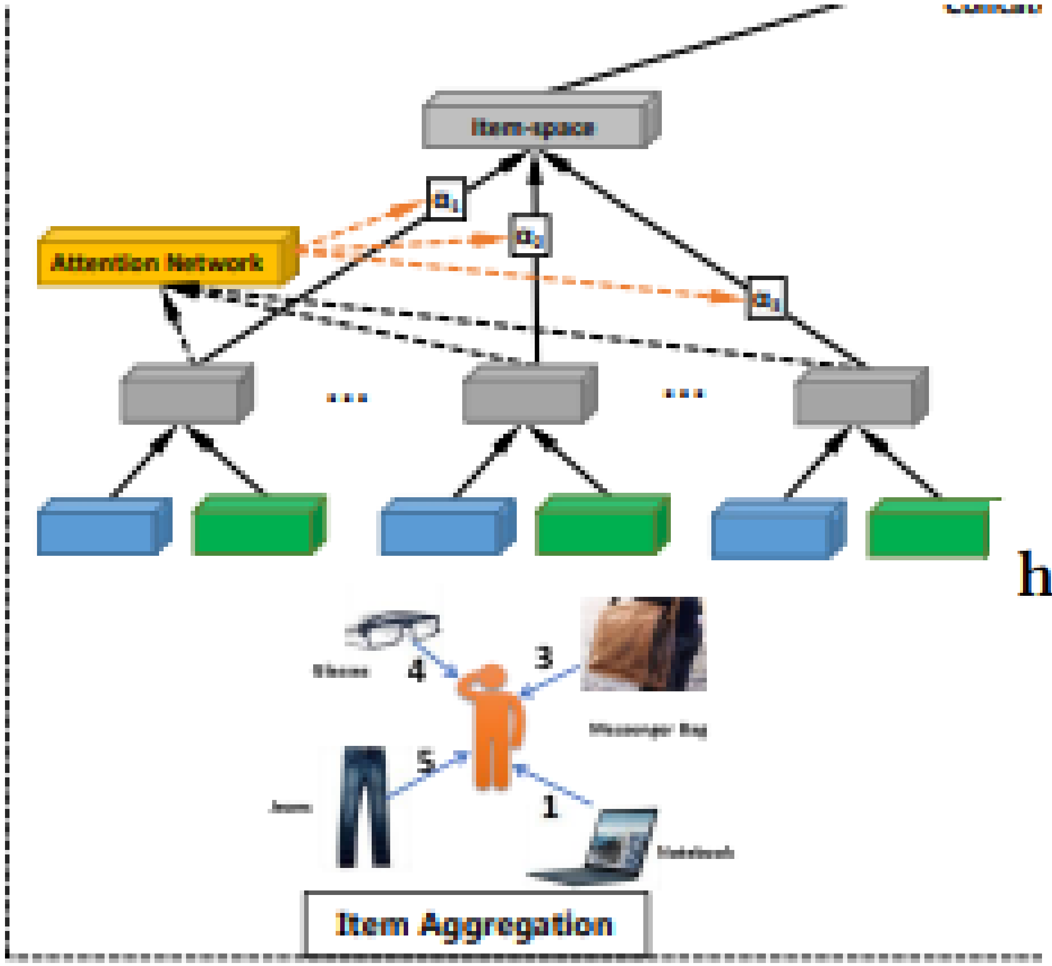
2) Social graph에서 social-space user latent factor $h_i^S \in \mathbb{R}^{d}$ 를 학습시 활용

➢ 2개의 factor 를 결합하여 final user latent factor h_i 를 만듬

3) User Modeling

1. **Item Agg**
2. Social Agg
3. Learning user Latent Factor

1



- user-item graph : users and items간 interactions + items 에 대한 users' opinions(or rating scores)
- learning item-space user latent factors h_i^I 를 위해 interactions과 opinions를 jointly capture하기 위한 방식
- 목적 : user u_i 가 interactions한 item과 item에 대한 사용자의 의견을 고려하여 item space user latent factor h_i^I 를 학습

$$\mathbf{h}_i^I = \sigma(\mathbf{W} \cdot Aggre_{items}(\{\mathbf{x}_{ia}, \forall a \in C(i)\}) + \mathbf{b}) \qquad (1)$$

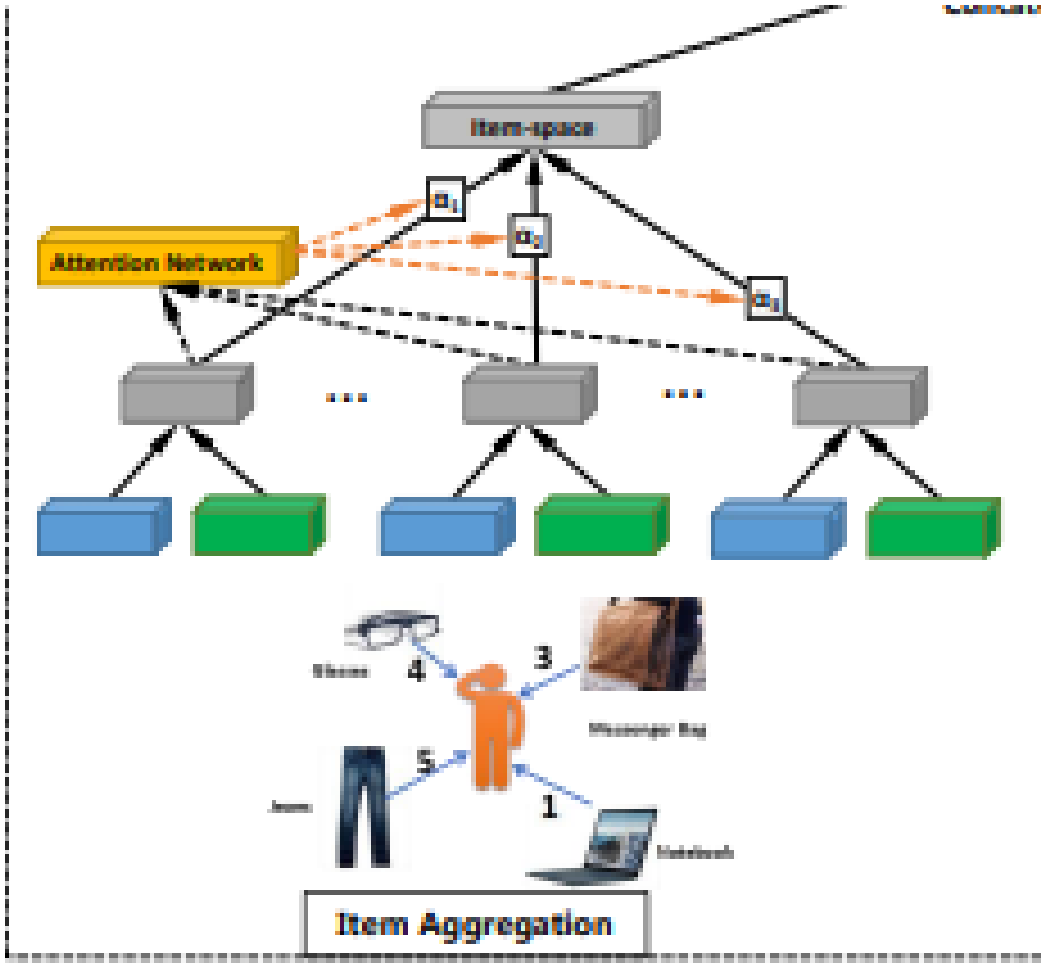
- $C(i)$: user u_i 가 interactions한 item (u_i 의 neighbor)
- x_{ia} : representation vector. user u_i 와 item v_a opinion-aware interaction
- $Aggre_{items}$: items aggregation function

3) User Modeling

1

1. **Item Agg**
2. Social Agg
3. Learning user Latent Factor

2. THE PROPOSED FRAMEWORK



- $Aggre_{items}$(items aggregation function) 의 정의
- r : user's opinions (or rating scores) > user latent factor modeling 에 기여 > dense vector representation > opinion embedding vector $e_r \in \mathbb{R}^d$ 도입
- User u_i 와 item v_a 사이의 interactions 을 위해, opinion-aware interaction representation에서 item embedding q_a 와 opinion embedding e_r 의 concatenation 를 입력으로 다층 인식 퍼셉트론(MLP) g_v 사용

$$\mathbf{x}_{ia} = g_v([\mathbf{q}_a \oplus \mathbf{e}_r]) \qquad (2)$$

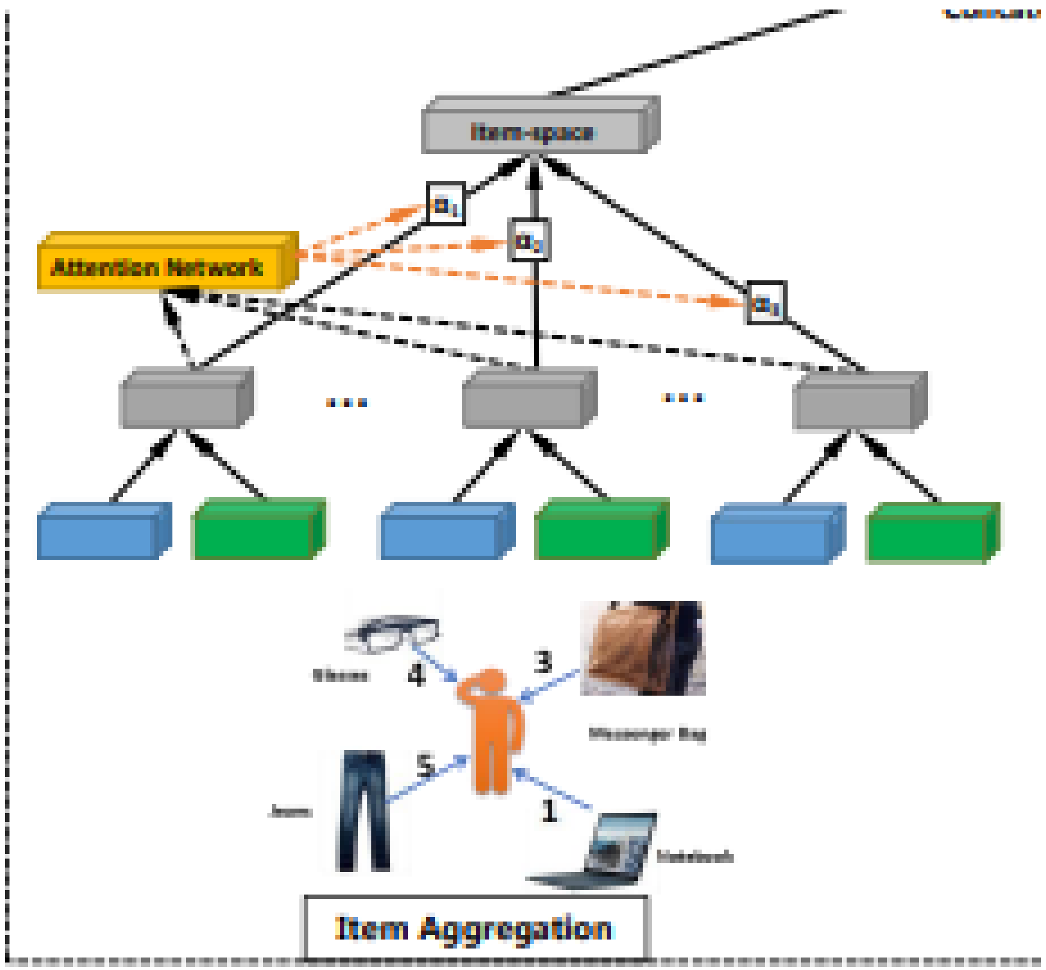
- $Aggre_{items}$: element-wise mean in $\{x_{ia}, \forall a \in C(i)\}$ vec
- 이 mean-based aggregator는 a localized spectral convolution 의 a linear approximation

3) User Modeling

1. **Item Agg**
2. Social Agg
3. Learning user Latent Factor

2. THE PROPOSED FRAMEWORK

1



$$\mathbf{h}_i^I = \sigma(\mathbf{W} \cdot \left\{ \sum_{a \in C(i)} \alpha_{ia} \mathbf{x}_{ia} \right\} + \mathbf{b}) \qquad (4)$$

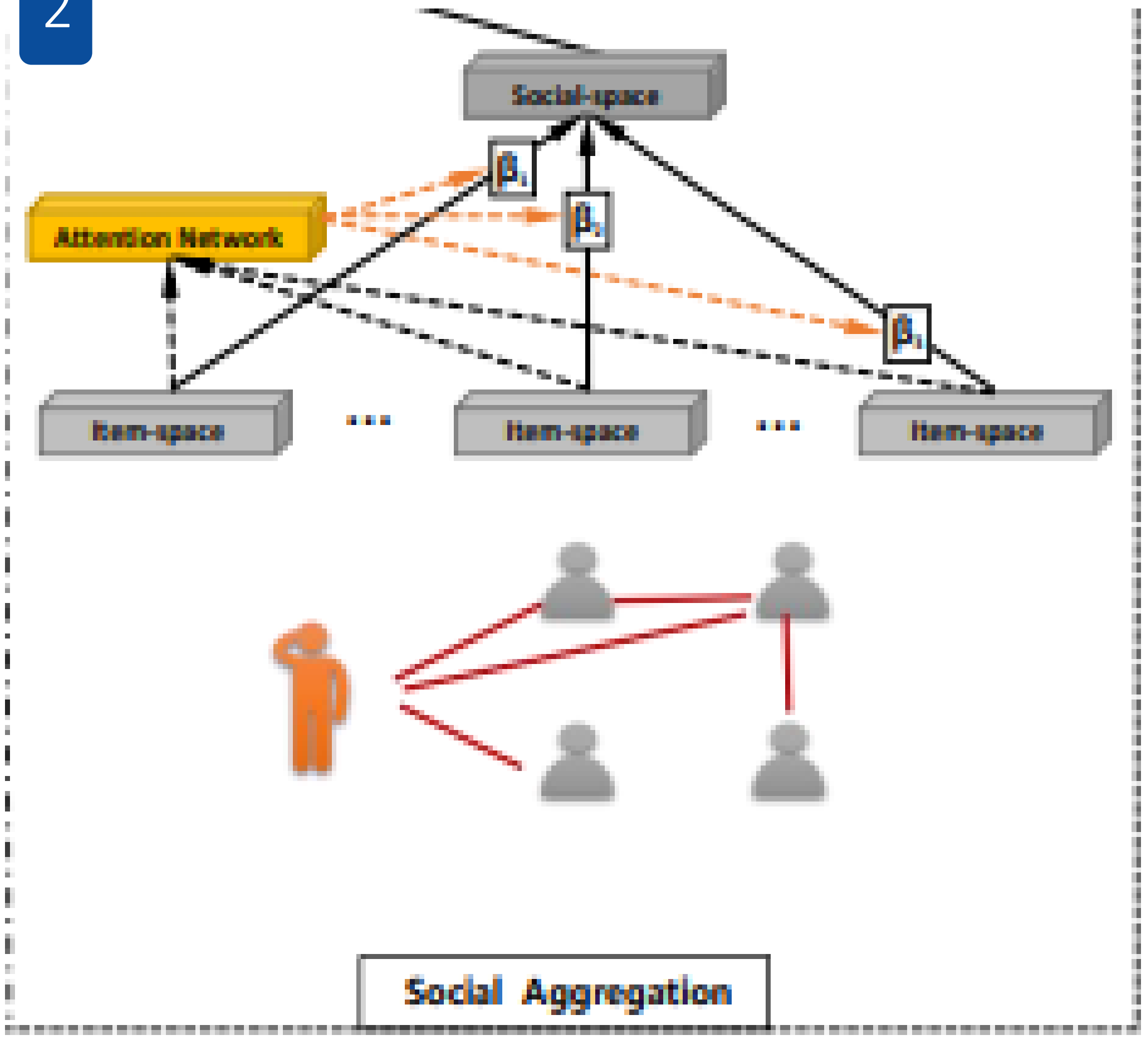
- α_{ia} : C(i)에서 user u_i 의 preference를 characterizing item-space latent factor기여하는 $v_a$interaction 가중치
- item attention α_{ia} 를 attention network(two-layer neural network)로 매개 변수화
- interaction 의 opinion-aware representation x_{ia}, user u_i의 embedding p_i

$$\alpha_{ia}^* = \mathbf{w}_2^T \cdot \sigma(\mathbf{W}_1 \cdot [\mathbf{x}_{ia} \oplus \mathbf{p}_i] + \mathbf{b}_1) + b_2 \qquad (5)$$

$$\alpha_{ia} = \frac{exp(\alpha_{ia}^*)}{\sum_{a \in C(i)} exp(\alpha_{ia}^*)} \qquad (6)$$

3) User Modeling

2



- social-space user latent factors learning 위해 social relations 의 heterogeneous strengths 고려필요
- users social information과 그들의 information을 aggregate 하기위해 attention mechanism social friends 를 선택 to select
- u_i 의 이웃 N(i) 의 item space latent factor 집계 h_i^S

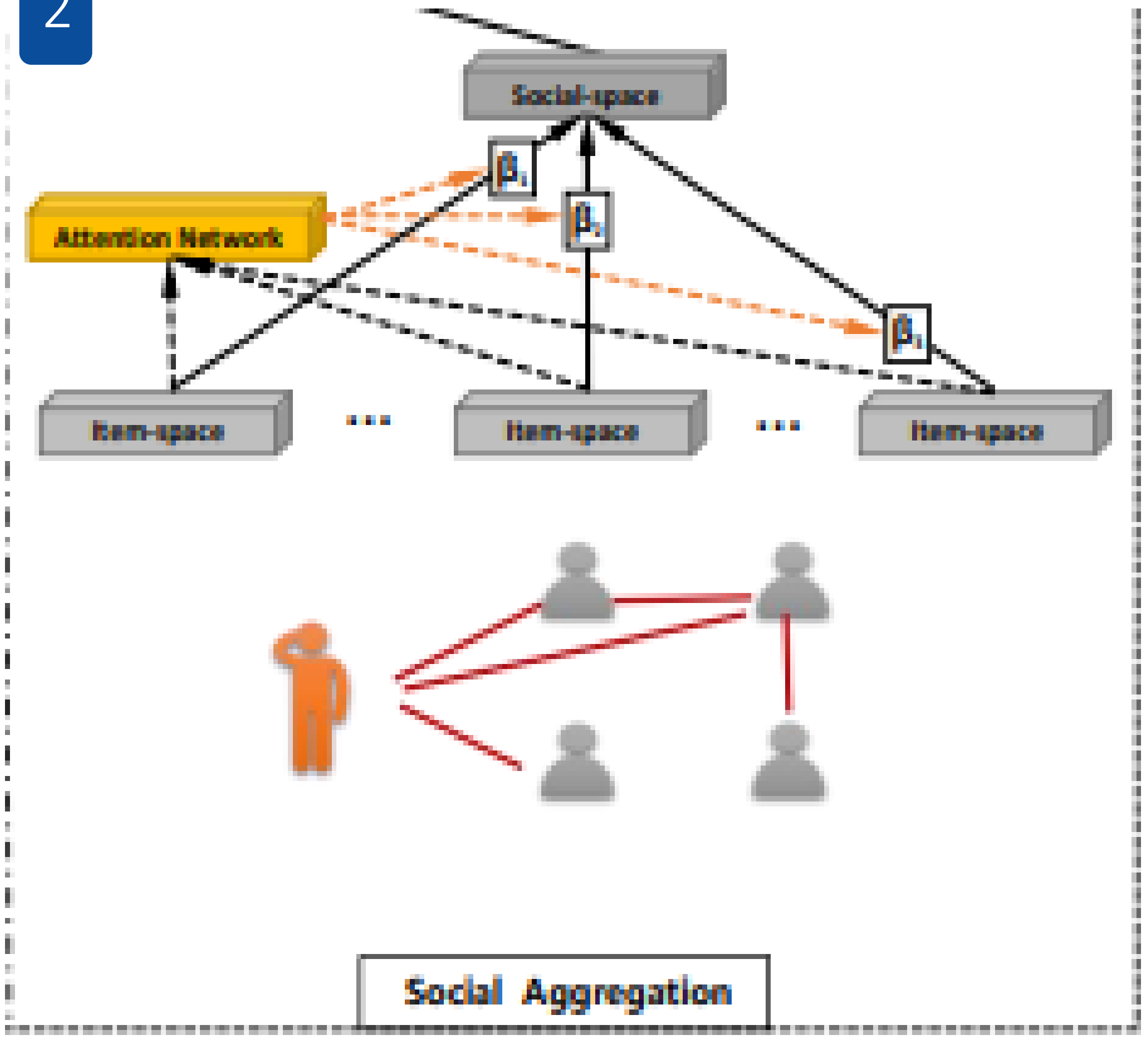
$$\mathbf{h}_i^S = \sigma(\mathbf{W} \cdot Aggre_{neigbhors}(\{\mathbf{h}_o^I, \forall o \in N(i)\}) + \mathbf{b}) \qquad (7)$$

- 여기서 $Aggre_{neigbhors}$: user's neighbors 의 aggregation function

$$\mathbf{h}_i^S = \sigma(\mathbf{W} \cdot \left\{ \sum_{o \in N(i)} \beta_i \mathbf{h}_o^I \right\} + \mathbf{b}) \qquad (8)$$

3) User Modeling

2



- $\beta_i : \frac{1}{\mathrm{N}(i)}$ mean-based aggregator에서 모든 neighbors 에 고정 > 모든 neighbors 동등한 영향
- Strong과 weak ties 가 있으므로 u_i 에 영향을 미치는 중요한 사용자를 추출하기 위해 2계층 신경망을 사용하여 attention mechanism 수행
- Social attention β_{io} 와 user embedding p_i 를 연관

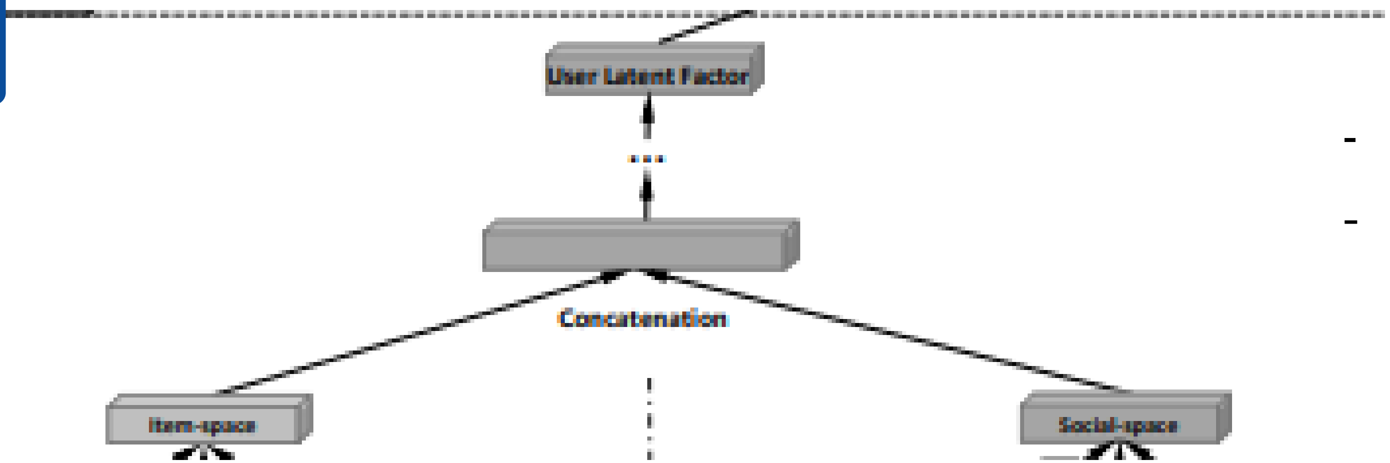
$$\mathbf{h}_i^S = \sigma(\mathbf{W} \cdot \left\{ \sum_{o \in N(i)} \beta_{io} \mathbf{h}_o^I \right\} + \mathbf{b}) \tag{9}$$

$$\beta_{io}^* = \mathbf{w}_2^T \cdot \sigma(\mathbf{W}_1 \cdot [\mathbf{h}_o^I \oplus \mathbf{p}_i] + \mathbf{b}_1) + b_2 \tag{10}$$

$$\beta_{io} = \frac{exp(\beta_{io}^*)}{\sum_{o \in N(i)} exp(\beta_{io}^*)} \tag{11}$$

3) User Modeling

3



- h_i^S 와 h_i^I 를 concat
- l 은 hidden layer 의 index

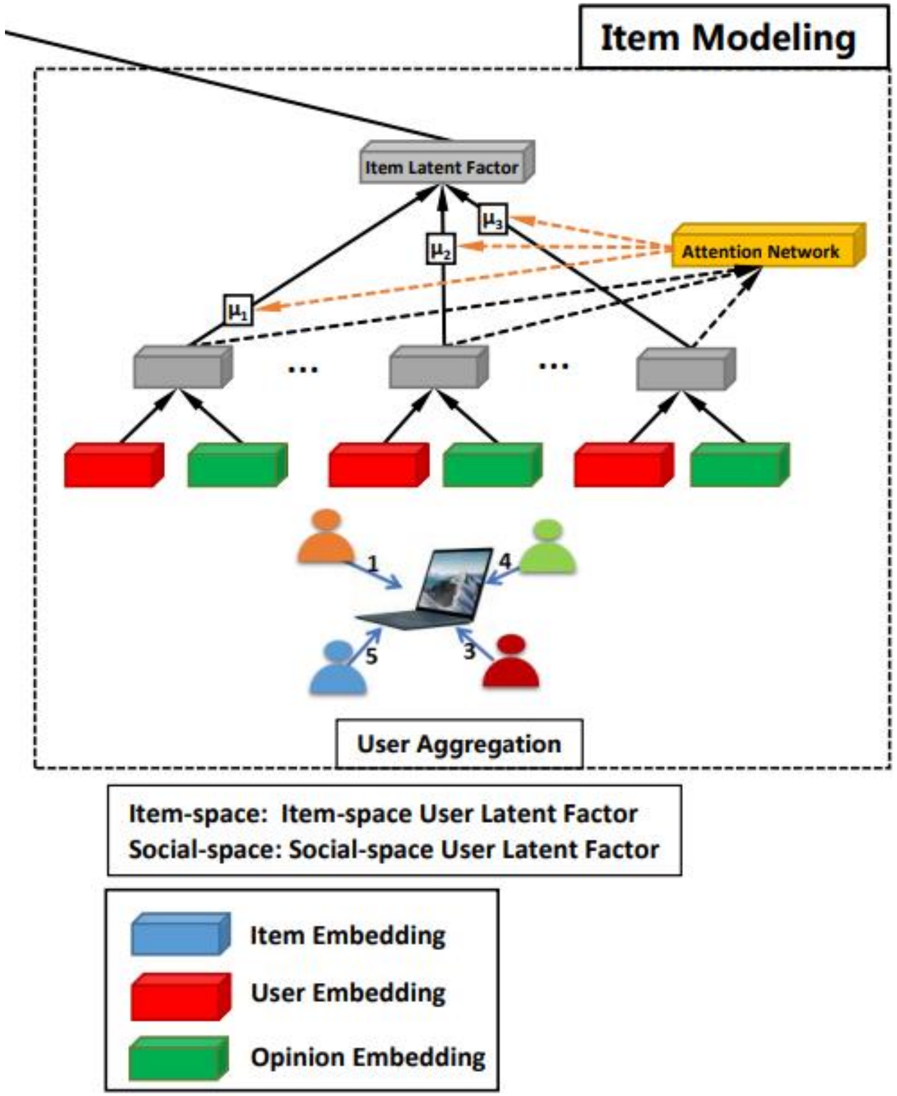
$$\mathbf{c}_1 = \left[\mathbf{h}_i^I \oplus \mathbf{h}_i^S\right] \tag{12}$$

$$\mathbf{c}_2 = \sigma(\mathbf{W}_2 \cdot \mathbf{c}_1 + \mathbf{b}_2) \tag{13}$$

$$\cdots$$

$$\mathbf{h}_i = \sigma(\mathbf{W}_l \cdot \mathbf{c}_{l-1} + \mathbf{b}_l) \tag{14}$$

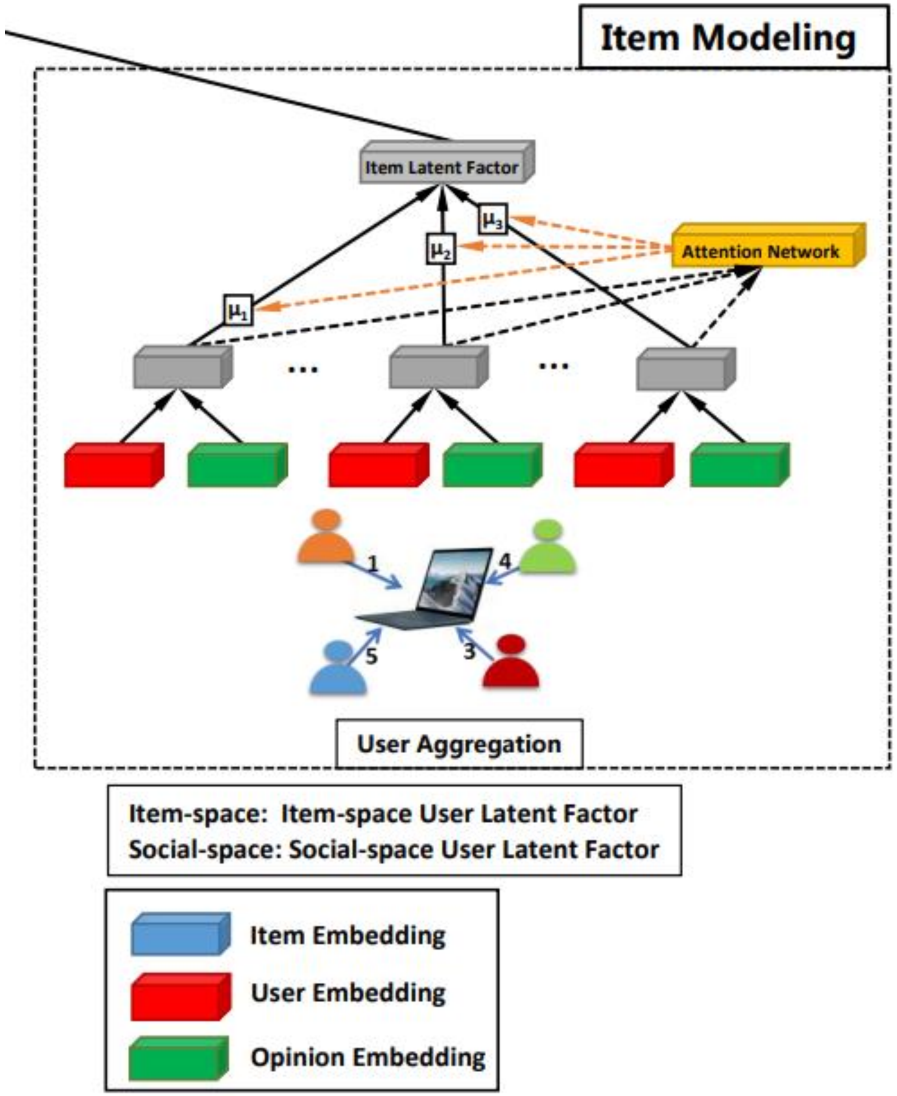
4) Item modeling



- user aggregation에 의한 item v_j에 대한 item latent factor(z_j)를 학습
- item 은 interactions 과 opinions 을 포함하는 user-item graph 와 연결 > item latent factors 학습을 위해 interactions 와 opinions 동시 포착 필요
- $B(j)$로 표시된 v_j 와 interaction한 user set에서 정보 Aggregation
- 같은 item이라도 사용자마다 interaction시 다른 의견을 표현 가능.
- 서로 다른 사용자의 이러한 의견은 사용자가 제공하는 다양한 방법으로 동일한 item의 특성을 포착할 수 있으며, 이는 item latent factors 모델링에 도움

$$\mathbf{f}_{jt} = g_u([\mathbf{p}_t \oplus \mathbf{e}_r]) \quad (15)$$

4) Item modeling



- $Aggre_{users}$: item latent factor(z_j)를 학습하기 위해 v_j 에 대한 $B(j)$ opinion-aware interaction representation 을aggregate

$$\mathbf{z}_j = \sigma(\mathbf{W} \cdot Aggre_{users}(\{\mathbf{f}_{jt}, \forall t \in B(j)\}) + \mathbf{b}) \qquad (16)$$

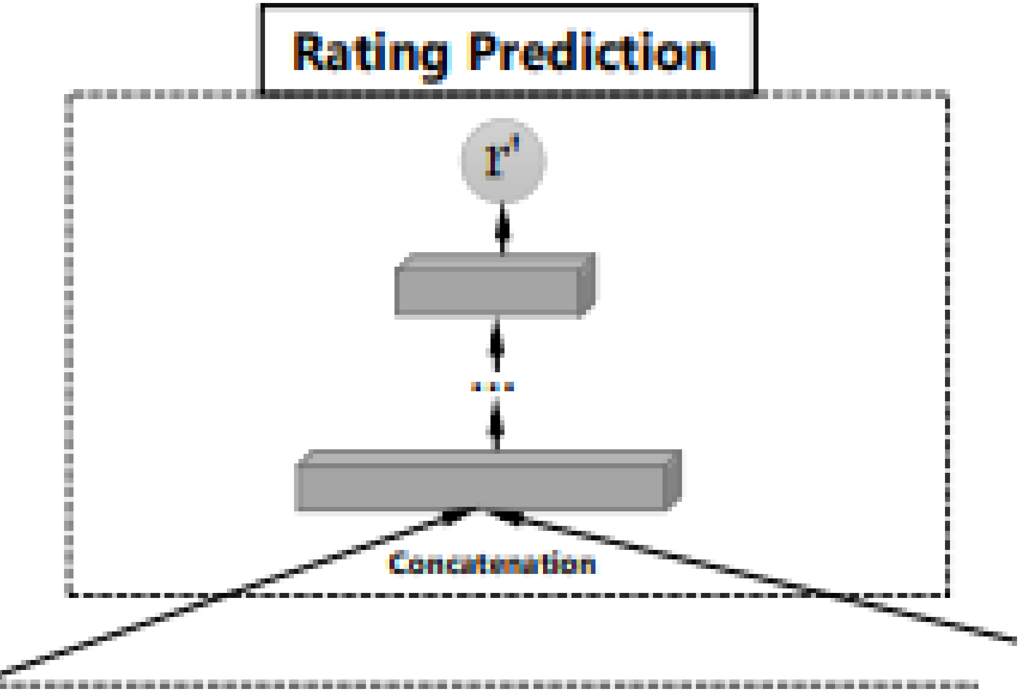
- f_{jt}와 q_j 를 입력으로 하는 2 layer two-layer neural attention
- Network 를 사용하여 사용자에 대한 differentiate
- the importance weight μ_{jt} (user-item interactions 에서 learning item latent factor시 heterogeneous influence 를 capture 하기 위해) 도입

$$\mathbf{z}_j = \sigma(\mathbf{W} \cdot \left\{ \sum_{t \in B(j)} \mu_{jt} \mathbf{f}_{jt} \right\} + \mathbf{b}) \qquad (17)$$

$$\mu_{jt}^* = \mathbf{w}_2^T \cdot \sigma(\mathbf{W}_1 \cdot [\mathbf{f}_{jt} \oplus \mathbf{q}_j] + \mathbf{b}_1) + b_2 \qquad (18)$$

$$\mu_{jt} = \frac{exp(\mu_{jt}^*)}{\sum_{t \in B_{(j)}} exp(\mu_{jt}^*)} \qquad (19)$$

5) Rating Prediction



- rating prediction task 사용
- user와 item의 잠재 요인(즉, h_i 및 z_j)을 사용하여 $h_i \oplus z_j$ 를 concat
- rating prediction 을 위해 MLP에 feed

$$\mathbf{g}_1 = [\mathbf{h}_i \oplus \mathbf{z}_j] \tag{20}$$
$$\mathbf{g}_2 = \sigma(\mathbf{W}_2 \cdot \mathbf{g}_1 + \mathbf{b}_2) \tag{21}$$
$$\cdots$$
$$\mathbf{g}_{l-1} = \sigma(\mathbf{W}_l \cdot \mathbf{g}_{l-1} + \mathbf{b}_l) \tag{22}$$
$$r'_{ij} = \mathbf{w}^T \cdot \mathbf{g}_{l-1} \tag{23}$$

- l : hidden layer 의 index
- r_{ij}' : u_i에서 v_j로의 predicted rating

6) Model Training

- rating prediction task
- |O| : observed ratings 수
- r_{ij} : ground truth rating assigned by the user i on the item j
- 최적화 : RMSprop[31]
- 3가지 임베딩 무작위 초기화 > training시 공동 학습
- Raw feature : 매우 크고 매우 희박하기 때문에 각 사용자와 item을 나타내는 데 원핫 벡터를 사용하지 않음
- 과적합 방지를 위한 Dropout 사용

$$Loss = \frac{1}{2|O|} \sum_{i,j \in O} (r'_{ij} - r_{ij})^2 \qquad (24)$$

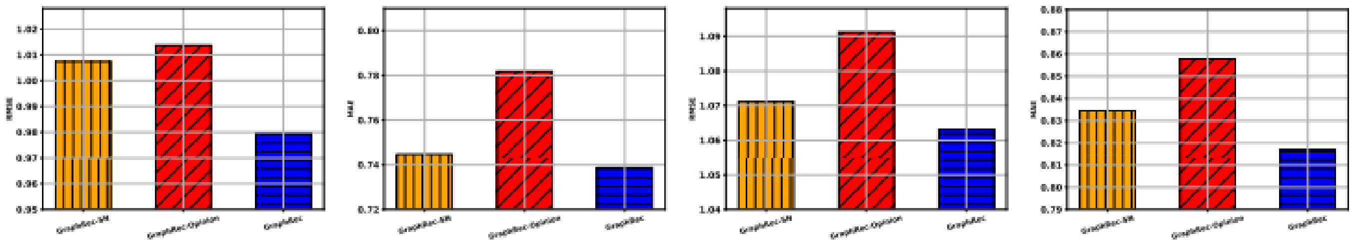
4. EXPERIMENT

Table 2: Statistics of the datasets

Dataset	Ciao	Epinions
# of Users	7,317	18,088
# of Items	10,4975	261,649
# of Ratings	283,319	764,352
# of Density (Ratings)	0.0368%	0.0161%
# of Social Connections	111,781	355,813
# of Density (Social Relations)	0.2087%	0.1087%

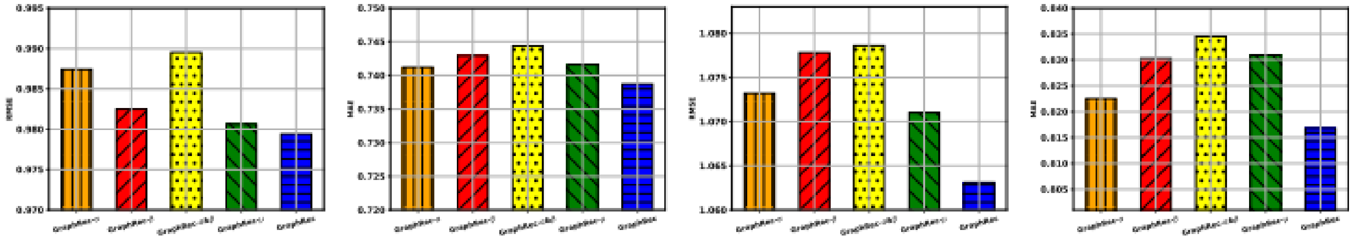
Table 3: Performance comparison of different recommender systems

Training	Metrics	Algorithms								
		PMF	SoRec	SoReg	SocialMF	TrustMF	NeuMF	DeepSoR	GCMC+SN	**GraphRec**
Ciao (60%)	MAE	0.952	0.8489	0.8987	0.8353	0.7681	0.8251	0.7813	0.7697	0.7540
	RMSE	1.1967	1.0738	1.0947	1.0592	1.0543	1.0824	1.0437	1.0221	1.0093
Ciao (80%)	MAE	0.9021	0.8410	0.8611	0.8270	0.7690	0.8062	0.7739	0.7526	0.7387
	RMSE	1.1238	1.0652	1.0848	1.0501	1.0479	1.0617	1.0316	0.9931	0.9794
Epinions (60%)	MAE	1.0211	0.9086	0.9412	0.8965	0.8550	0.9097	0.8520	0.8602	0.8441
	RMSE	1.2739	1.1563	1.1936	1.1410	1.1505	1.1645	1.1135	1.1004	1.0878
Epinions (80%)	MAE	0.9952	0.8961	0.9119	0.8837	0.8410	0.9072	0.8383	0.8590	0.8168
	RMSE	1.2128	1.1437	1.1703	1.1328	1.1395	1.1476	1.0972	1.0711	1.0631



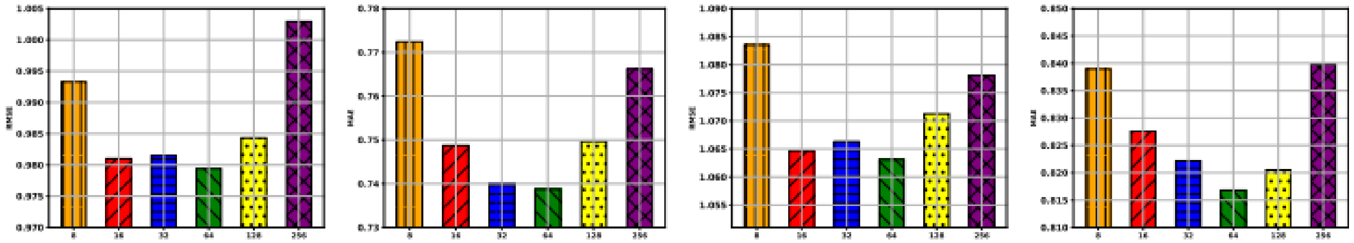
(a) Ciao-RMSE (b) Ciao-MAE (c) Epinions-RMSE (d) Epinions-MAE

Figure 3: Effect of social network and user opinions on Ciao and Epinions datasets.



(a) Ciao-RMSE (b) Ciao-MAE (c) Epinions-RMSE (d) Epinions-MAE

Figure 4: Effect of attention mechanisms on Ciao and Epinions datasets.



(a) Ciao-RMSE (b) Ciao-MAE (c) Epinions-RMSE (d) Epinions-MAE

Figure 5: Effect of embedding size on Ciao and Epinions datasets.

Q & A

감사합니다