ICDE 2023

Influential Recommender System

- Authors: Haoren Zhu, Hao Ge, Xiaodong Gu, Pengfei Zhao, Dik Lun Lee
- Main Affiliation: Hong Kong University of Science and Technology

2025.07.22

서정호

Contents

1. Proposal

▼ User-oriented Recommender System (URS)

- Traditional Recommender Systems
- User's historical interests [internal interests]
  → RS recommends to adapt the user's current interests
  : Passive Recommendation

Goal in commercial applications

- User's interests를 확장시켜서 user가 몰랐거나 관심없던 items를 Accept하게 만드는 것.
  → Customer[user] Interactions를 증가시킬 방안이 필요.
  → URS: User's interests 내에서만 Recommendation이 이루어지므로 불가능.
  → **IRS 제안.**

1. Proposal

Influential Recommender System (IRS)

▼ Concepts

- User u, item i

1. Objective Item 설정.: Target item, i_t
2. Influence path 설정.: Sequential list of carefully selected items

→ user's historical interests 만족 & user's interests를 i_t로 확장.

→ User가 given objective item을 좋아하도록 유도.: Active Recommendation

1. Proposal

Example with watching movies

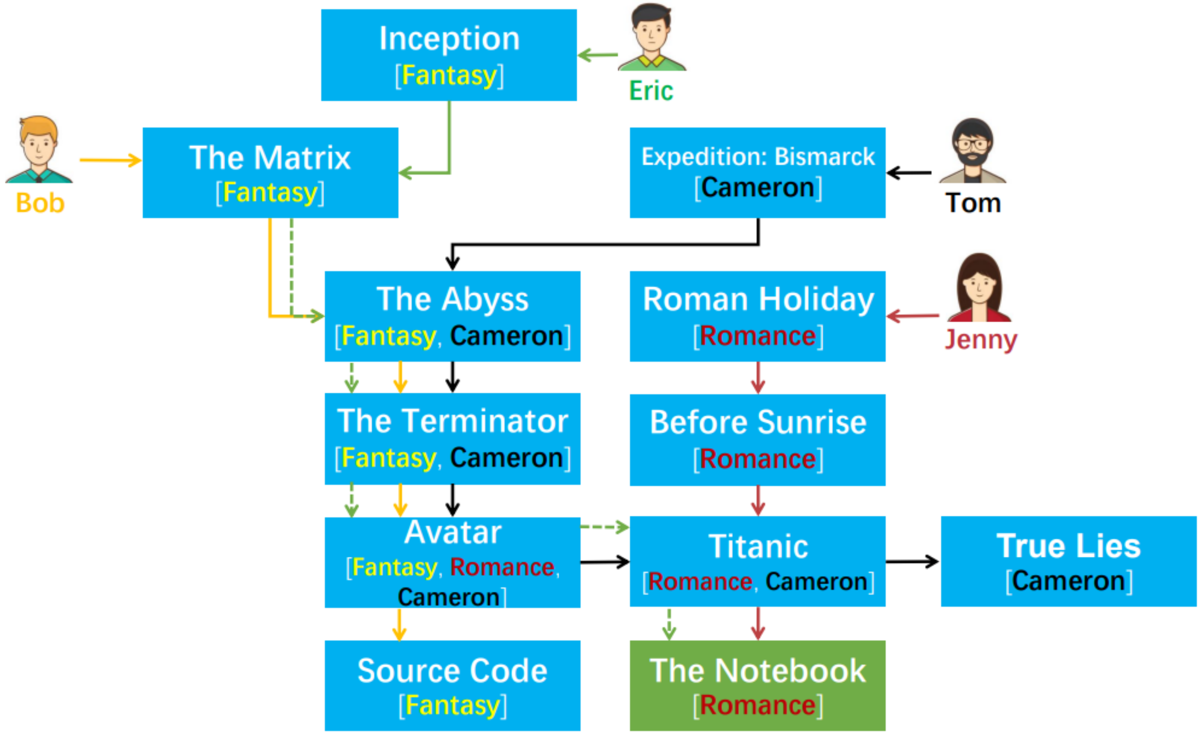


Fig. 1. Illustration of Influential Recommender System.

Fig. 1.이 잘못되었습니다.
Tom의 실선이 The Matrix에도 가야하고, The Matrix가 Cameron에 의해 directed되었어야 합니다.
따라서 Tom: Expedition: Bismarck → The Matrix → The Abyss 여야 합니다.

- Interest 대상 (3가지)
  - Genre: Fantasy, Romance
  - Director: Cameron
- Arrows
  - Solid arrows [실선]: Users' viewing histories
    → users' chronological behavioral sequences
  - Dash arrows [점선]: Users' influence path
- Objective item for user Eric: The Notebook
  - Influence Path: Inception → The Matrix → The Abyss → The Terminator → Avatar → Titanic → The Notebook

1. Only Fantasy
2. Tom과 movie 겹침 → Tom's watching history를 recommendation
   : Fantasy + Cameron
   → Fantasy + Cameron + Romance
   → Cameron + Romance
3. Jenny와 movie 겹침 → Jenny's watching history를 recommendation
   : Cameron + Romance
   → Romance: Objective item

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

DGIST DATA SCIENCE LAB

1. Proposal

▼ Challenges

1. Influence Path
    a. User's historical interest에서 너무 많이 벗어나면 안됨. for user's trust and satisfaction
    b. Objective item에 관계된 추천을 해야 함.
2. Generation of influence path
    - items간의 sequential dependencies에 부합해야 함.
    - 즉, next item/action은 user가 최근에 참여한 items/actions에 더욱 의존해야 함.
3. User's preference for external influence
    - User마다 외부 영향에 대한 선호가 다름. 즉, personal함.
        a. Extrovert한, external influences에 쉽게 시도하는 user
           → influence path가 more aggressive할 수 있음.
        b. New interests를 설득하기 어려운 user
           → influence path를 더 조심스럽게 해야 함.
    - Impressionability: External influence에 대하여 user마다 서로 다른 preference를 가짐.

해결책 → **IRN framework including PIM**

1. Proposal

Idea

- Influential Recommender Network (IRN)
  - **Challenge 2**: Item's sequential dependencies를 encoding 해야 함.
    1. User-Item interaction sequence: sequential 구조.
       & NLP sentences: sequential 구조.
    2. Items간의 dependencies: 인접하거나 멀리 있는 items 모두 고려.
       & Words간의 dependencies: 인접하거나 멀리 있는 words 모두 고려.

    → NLP의 Transformer model과 특성이 유사.

    → **Transformer-based sequential model을 구축. == IRN framework**
  - **Challenge 3**: Each user마다 개인화된 influence path 필요.

    → Item attention weights 계산할 때, user embedding을 넣어야 함.

    → **Personalized Impressionability Mask (PIM)**

2. Influential Recommender System

A. Problem Definition

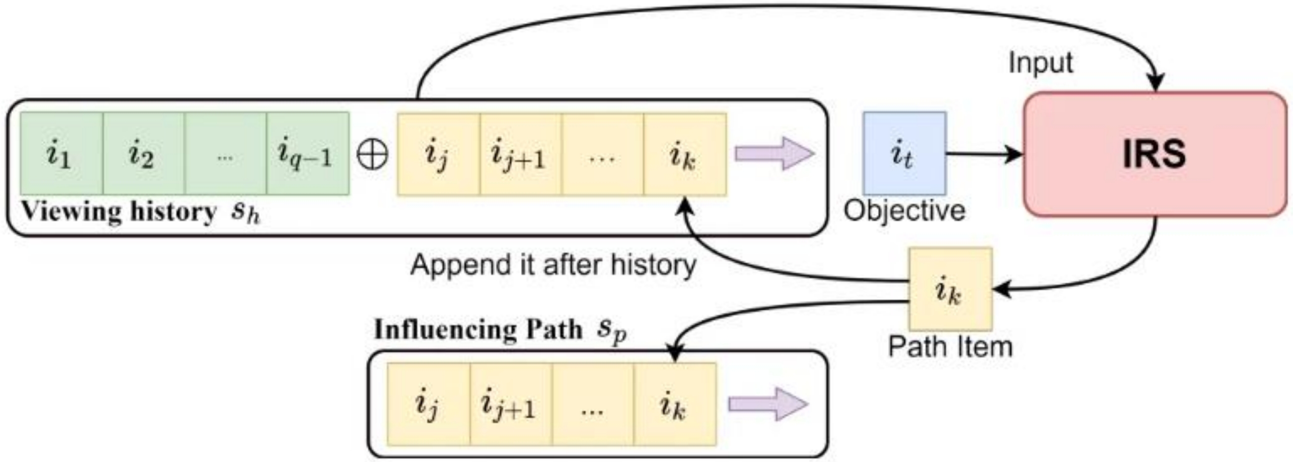


Fig. 2. Illustration of the Influential Recommendation Process. *IRS* recursively recommends a path item i_k until reaching the objective item i_t or exceeding the maximum length allowed. All the path items constitute the influence path s_p.

- IRS만을 기준으로 하면,
  - Input: $s_h \oplus s_p$
  - output: 최종 s_p

2. Influential Recommender System

Notation

- Item set: $I = \{i_1, i_2, \dots, i_{|I|}\}$
- User set: $U = \{u_1, u_2, \dots, u_{|U|}\}$

- 특정 user u에 대한, User-Item interaction sequence: $s = \{i_1, i_2, \dots, i_m\} \in S_u$
  - Watching Movie라고 하면, 시간의 흐름 $t_1, t_2, \dots, t_m$에 대한 movies 순서 sequence
  - Movie == item, watching = interaction
- 특정 user u에 대한, set of s: S_u
- 모든 (user,item)에 대한, interaction sequence set: $S = \cup_{u=1}^{|U|} S_u = \{s_1, s_2, \dots, s_{|S|}\}$

- 특정 user u에 대한,
  - user's viewing history: $s_h = \{i_1, i_2, \dots, i_{q-1}\}$ means user's original interests
  - Influence path: $s_p = \{i_j, i_{j+1}, \dots, i_k\}$
  - Objective item: i_t means target item

Algorithm

- Assumption: Simplicity를 위해, user가 모든 recommendations을 accept한다고 가정
  - 원래는 user가 recommendation을 보고 item accept or reject를 결정
- $\mathcal{F}$: Recommender function → learning 필요.
- s_p: 초기엔 empty set → 점차 채워짐.

Algorithm 1 Generate influence path s_p for user u

Input: Interaction history s_h, the objective item i_t, the maximum path length M
Output: influence path s_p
s_p = [ ]
while $|s_p| \leq M$ **do**
  $i = \mathcal{F}(s_h, i_t, s_p)$
  $s_p = s_p + i$
  if i is i_t **then**
    break
  end if
end while

학습이 완료된 후, 최종 $\mathcal{F}$를 algorithm 1에 사용함.

DGIST DATA SCIENCE LAB

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

2. Influential Recommender System

Previous Trials (1)

B. Path-finding Algorithms as IRS

Influence path generation == path-finding problem in Graph

- S_u로부터 graph를 그려야 함.
- s_h의 last item을 i_h라고 하면, i_h는 user의 recent interest임.
- i_h와 i_t 사이의 path를 찾아야 함. → path-finding algorithm 사용.
- 가장 먼저 소개된 path-finding algorithm: Pf2Inf
- Pf2Inf algorithm example

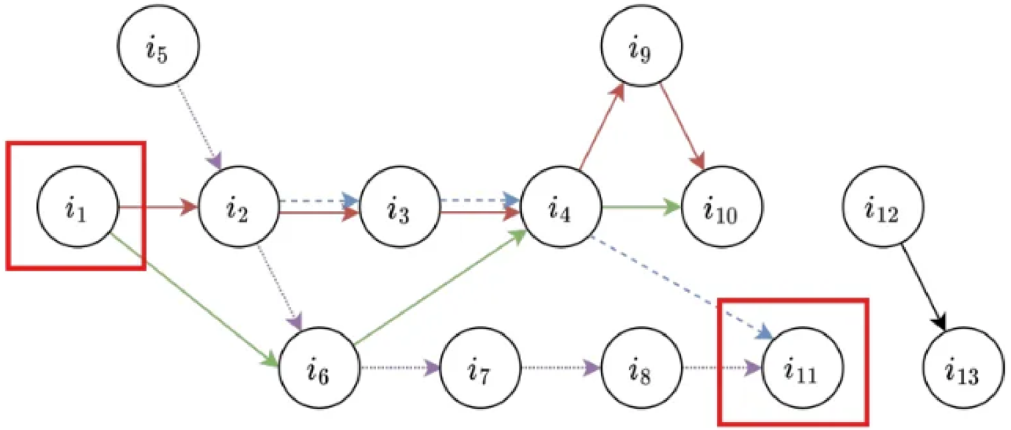


Fig. 3. An example of *Pf2Inf*. Here we construct an undirected graph from 5 user-item interaction sequences (one color per user). The arrow indicates the item order within a sequence.

- $i_h = i_1, i_t = i_{11}$일 때, shortest path를 influence path로 찾음.
  - $s_p = \{i_1, i_6, i_4, i_{11}\}$

DGIST DATA SCIENCE LAB

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

Previous Trials (1)

C. Adapting Existing RS with Greedy Search

Pf2Inf 단점.

1. s_h에서 items의 sequential patterns를 구할 수 없다.
2. Sparse recommendation dataset에서, disjoint graphs가 발생한다.

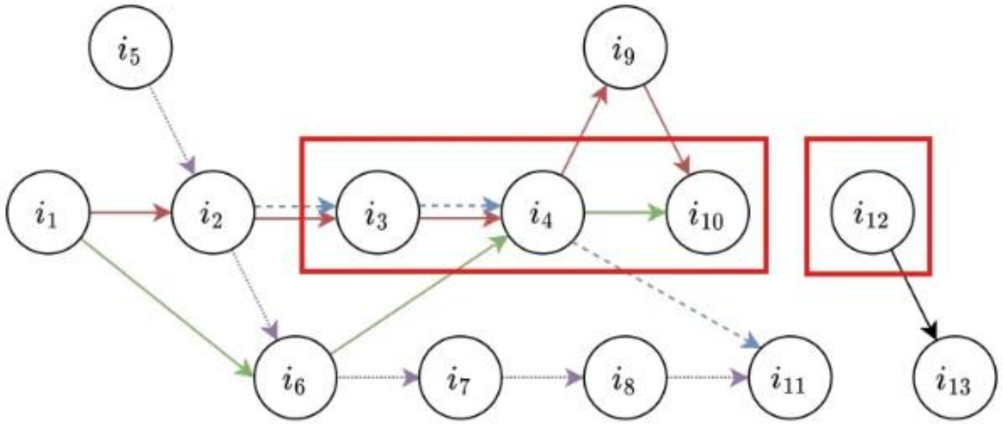


Fig. 3. An example of *Pf2Inf*. Here we construct an undirected graph from 5 user-item interaction sequences (one color per user). The arrow indicates the item order within a sequence.

이 경우 $s_h = \{i_3, i_4, i_{10}\}, i_t = i_{12}$일 때, 적절한 s_p를 찾을 수 없다.

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

2. Influential Recommender System

Previous Trials (2)

▼ Pf2Inf's Alternative = Rec2Inf

- Greedy search strategy

1. Given s_h, use existing RS $\mathcal{R}$ → generate top-k recommendations set $\mathcal{R}_k = \mathcal{R}\left(s_h \oplus s_p\right)$
2. Item embedding ← e.g. item2vec
3. Calculate distance betw. item $i \in \mathcal{R}_k$ and objective item i_t
4. Choose the closest item i' and then Add i' into the s_p
5. Repeat 1 - 4

   → **End if** i_t is added into the s_p

▼ Rec2Inf's Limitations

1. 각 iteration에서 distance가 minimum인 item 하나씩 s_p로 더해짐.
   → local optimal selection
   → global optimal influence path라고 할 수 없음.
2. Objective item i_t가 training process에는 포함되지 않음.
   → RS가 이른 시기에 계획될 수 없음.

2. Influential Recommender System

💡 [목표] Influence Path s_p 찾기.

- B & C에서 시도했던 graph-based model

  → path-finding problem이 잘 해결되지 않음.

  → graph-based model 포기.

- 다른 방법? → N.N-based IRN.

2. Influential Recommender System

Influential Recommender Network (IRN)

From the idea,

- **Transformer structure**
- **PIM for personalized**

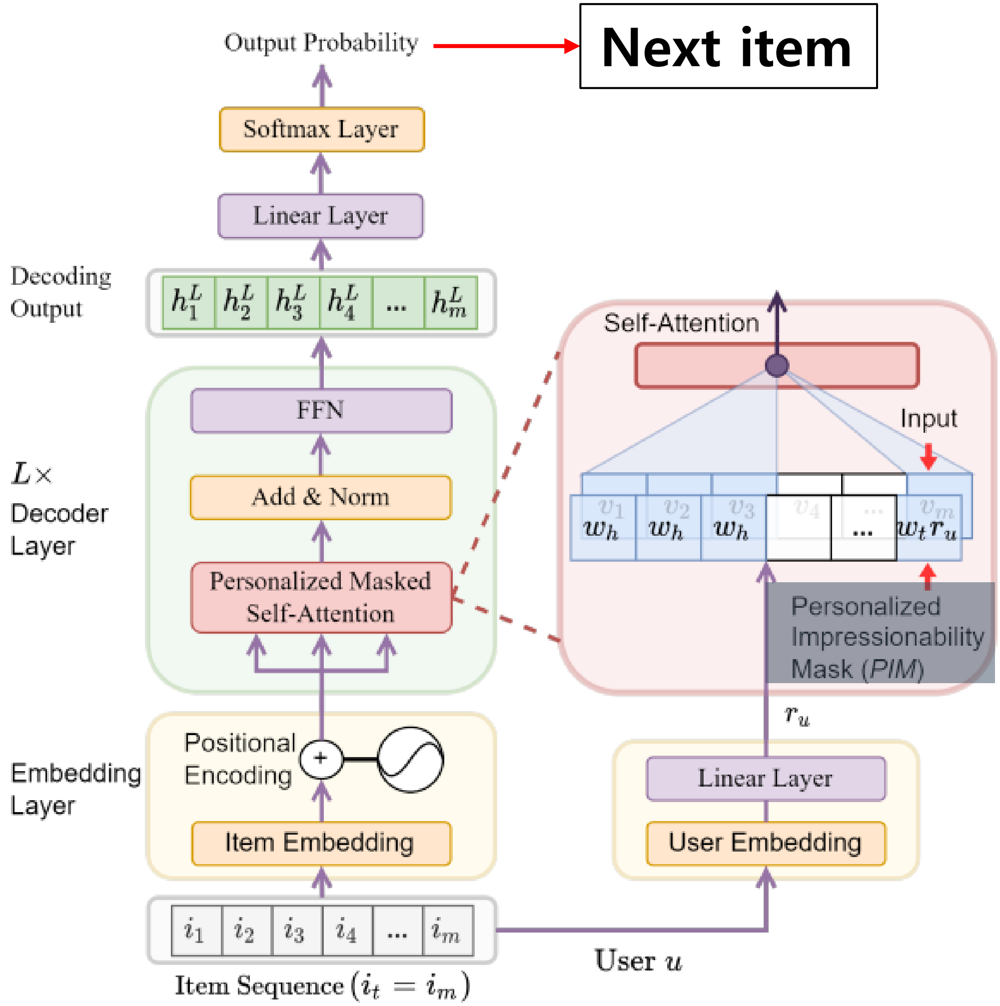
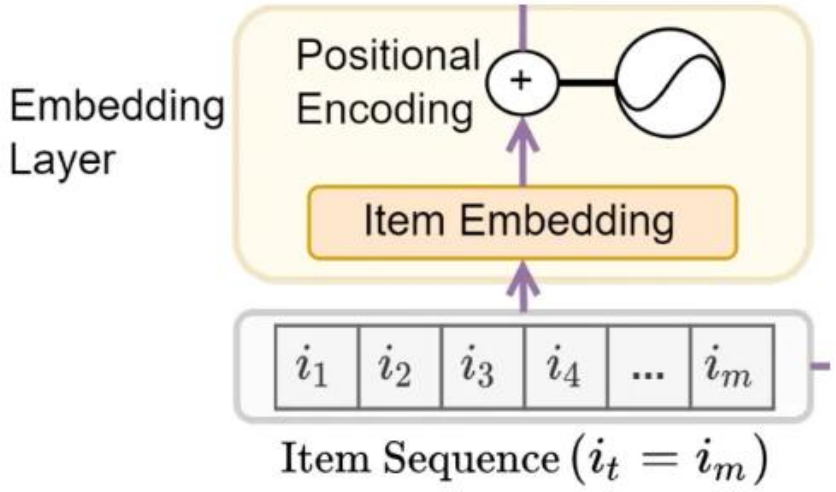


Fig. 4. Overall framework of *IRN*. Given a sequence $s = \{i_1, ..., i_m\}$ where the last item is the objective item ($i_t = i_m$), *IRN* first embeds s into vectors. Then it learns the hidden representation of the input sequence with a stack of L decoder layers. The self-attention layers in the decoders incorporate a specially designed *Personalized Impressionability Mask* to cater to users with different acceptance of external influence.

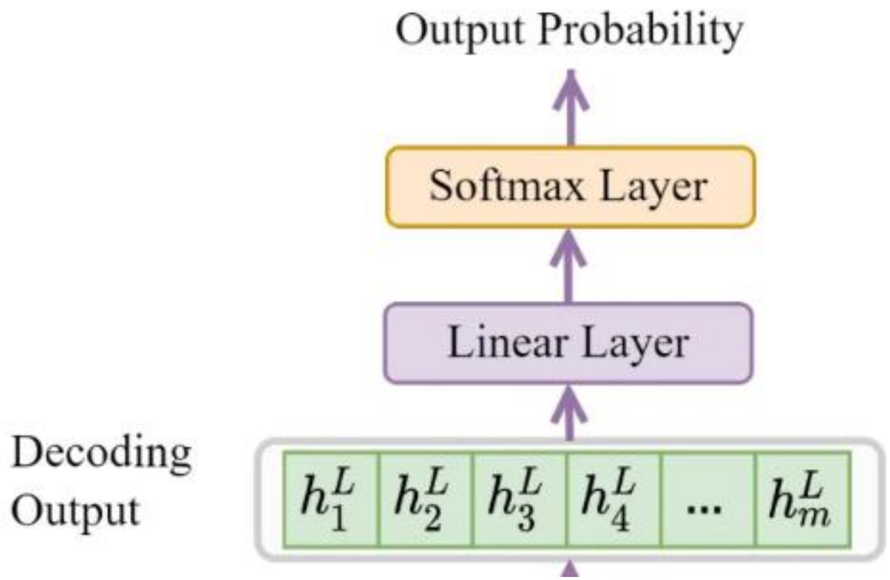
2. Influential Recommender System

Influential Recommender Network (IRN)

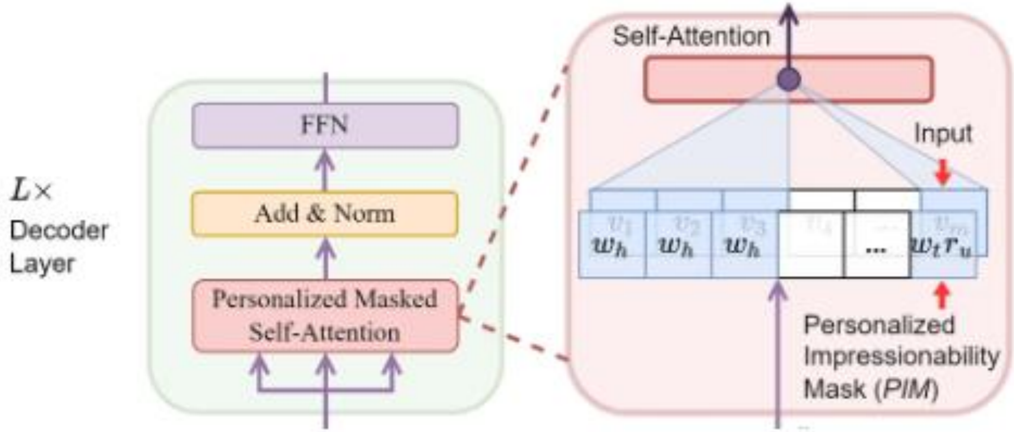
1. Embedding layer: input sequence → input vector



3. Output layer: hidden states → 확률분포



2. L decoder layers: 각각 self-attention layer 존재, with Personalized Masking

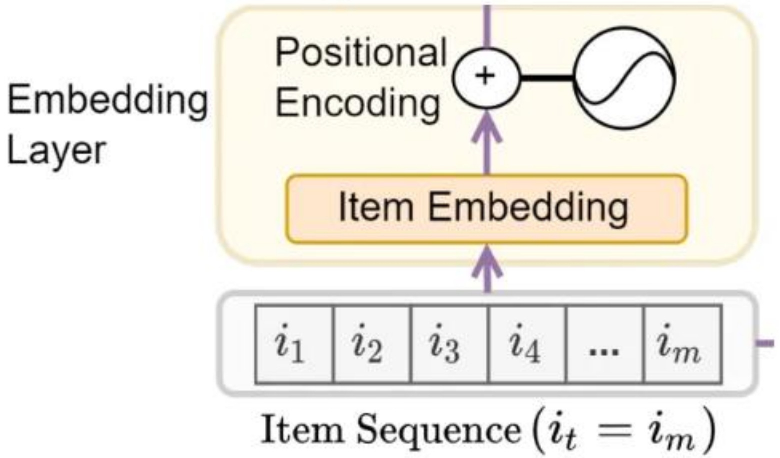


DGIST DATA SCIENCE LAB

2. Influential Recommender System

Influential Recommender Network (IRN)

▼ 1) Item Embedding



- Embedding: item[discrete token] → vector [output]
- 이 때 transformer 구조에서 ordering property를 살리기 위해, Positional Encoding 필요.

$$\mathbf{e}(i_j) = \mathrm{TE}[i_j] + \mathrm{PE}[j] \in \mathbb{R}^d \quad \text{for } j = 1, \ldots, m \quad (1)$$

  - $\mathrm{TE} \in \mathbb{R}^{|I| \times d}$: Token Embedding matrix [Item embedding]
  - $\mathrm{PE} \in \mathbb{R}^{m \times d}$: Positional Embedding matrix [Positional Encoding]
- Pre-trained embedding by item2vec model → better initial weights → better model performance

2. Influential Recommender System

Influential Recommender Network (IRN)

- 2) Decoder
  - L개의 layers, input → 1st layer → 2nd layer → ... → L-th layer → output
    - First decoder layer's input: $\mathbf{H}^0 = \{\mathbf{e}(i_1), \ldots, \mathbf{e}(i_m)\}$
    - Embedding sequence $\mathbf{H}^0$ → Sequence of hidden states $\mathbf{H}^L = \{\mathbf{h}_1^L, \ldots, \mathbf{h}_m^L\}$
  - PIM을 고려하지 않았을 경우, [Not personalized]
    - $\mathbf{h}_j$: hidden state of i_j ← [history, target] $i_1, i_2, \ldots, i_{j-1}, i_t$

$$\mathbf{h}_j = \mathrm{Dec}(\mathbf{h}_{<j}, \mathbf{h}_t) \quad (2)$$

    - Each decoder layer: $(l-1)^{th}$ layer가 l^{th} layer를 update.

$$\mathbf{H}^l = \text{Self-Attn}(\mathbf{H}^{l-1}) \quad (3)$$

    - But, 식 (3)은 틀린 식인 것 같음. Add&Norm, FFN이 고려되지 않음. in Fig. 4.

- 수정한 식.

$$\mathbf{H}^l = \mathrm{FFN}(\mathrm{AddNorm}(\text{Self-Attn}(\mathbf{H}^{l-1}))) \quad (3')$$

- 또한 위 구조에서 Query, Key, Values는 모두 같은 값을 사용.

$$\text{Self-Attn}(\mathbf{H}^l) = \mathrm{Attention}(\mathbf{H}^l, \mathbf{H}^l, \mathbf{H}^l)$$
$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \left[\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\right]^T \mathbf{V} \quad (4)$$

  - $\mathbf{Q}$: Query vector
  - $\mathbf{K}$: Key vector
  - $\mathbf{V}$: Value vector

- Personalized Impressionability Mask (PIM)
  - self-attention layer에 삽입
  1. Which items can be seen by the attention layer
  2. to what extent the item will be aggregated to produce the hidden states

2. Influential Recommender System

Influential Recommender Network (IRN)

- w_t: mask weight for i_t
- w_h: mask weight for preceding history.
  - j-th row: Only $i_1, \ldots, i_j$ predicts i_{j+1}.
- $w_t > w_h$: i_t에 더 많은 가중치를 부여. → w_t값을 조정함으로서 IRN의 공격적인 정도를 조정할 수 있음. [사람에 따라 공격적으로 유도하는 등.]

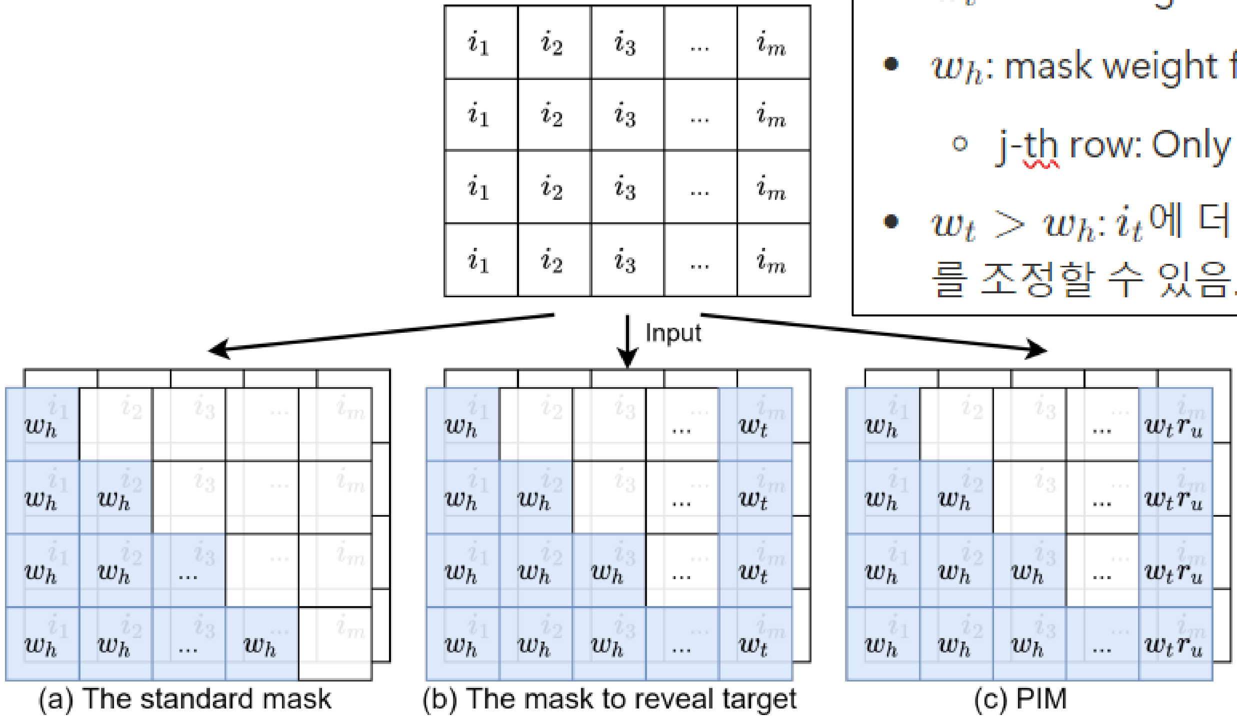
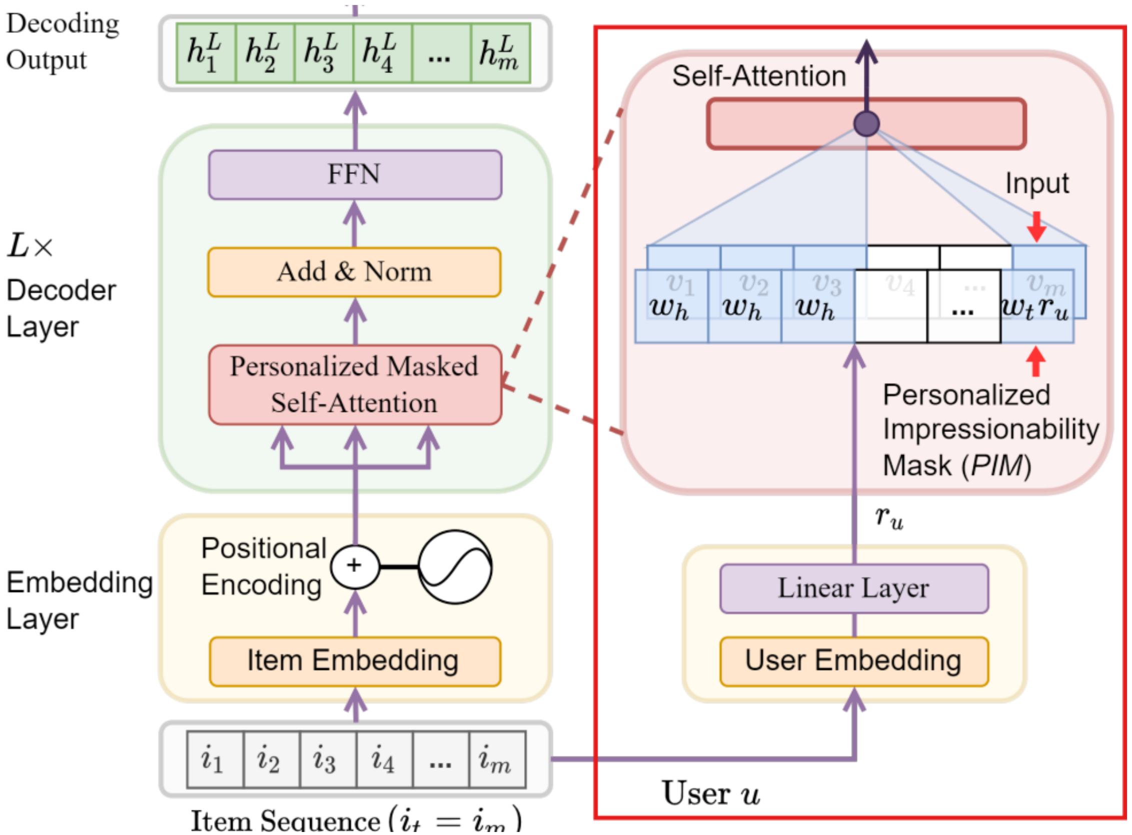


Fig. 5. Personalized Impressionability Mask (PIM). $\{i_1, \ldots, i_m\}$ is the input sequence, where the last item i_m is the objective item. (a) displays the standard mask of the decoder. For example, at the second line where the model predicts i_3, only i_1 and i_2 (in blue squares) are perceived by the attention layer while the follow-up items are masked. (b) displays the special mask to reveal the objective, and (c) displays the *PIM* decided by the personalized impressionability factor r_u.

2. Influential Recommender System

Influential Recommender Network (IRN)

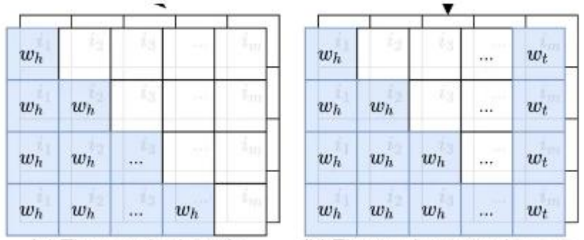


- Users마다 external influence에 대한 acceptance가 다름. [Personalized Curiousness of exploration]
  - → IRN의 aggressive 정도 조절 ↔ w_t 조정.

$$\mathbf{e}(u) = \mathcal{U}[u]$$
$$r_u = \mathbf{W}^U \mathbf{e}(u) \quad (5)$$

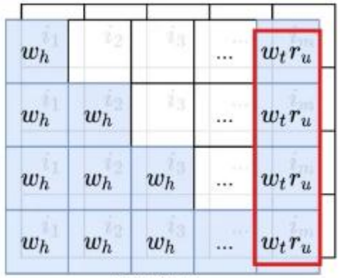
  - $\mathbf{e}(u) \in \mathbb{R}^{d'}$: embedding of user u
  - $\mathcal{U}$: user embedding matrix
  - $\mathbf{W}^U$: parameters of the linear transformation

- 기존: history & target



(a) The standard mask (b) The mask to reveal target

- paper: Personalized term 넣어서 PIM 삽입. user-term인 r_u를 곱하여, $w_t \times r_u$사용.



(c) PIM

▼ PIM 고려한 최종 식.

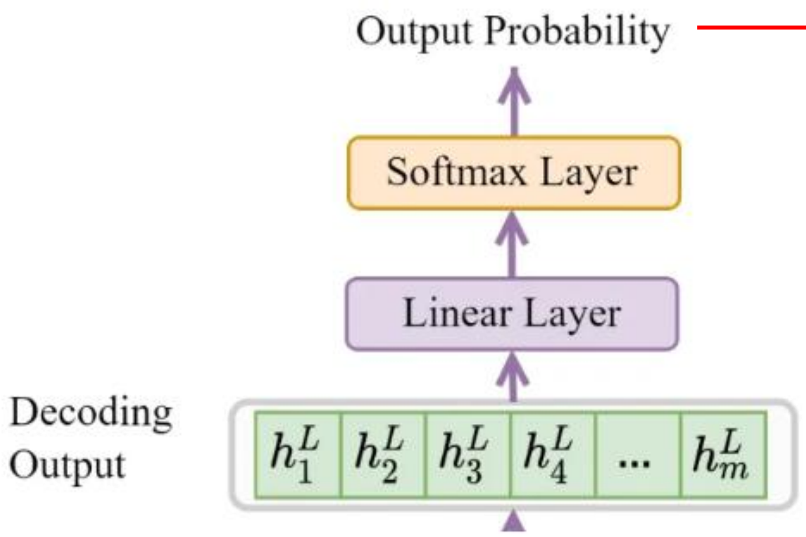
$$\mathbf{h}_j = \mathrm{Dec}\left(\mathbf{h}_{<j}, \mathbf{h}_t, r_u\right)$$
$$\mathbf{H}^l = \text{Self-Attn}\left(\mathbf{H}^{l-1}, r_u\right) \quad (*)$$
$$\mathbf{H}^l = \mathrm{FFN}\left(\mathrm{AddNorm}\left(\text{Self-Attn}\left(\mathbf{H}^{l-1}, r_u\right)\right)\right)$$
$$\text{for } l = 1, \ldots, L \quad (6)$$

- 마찬가지로, (*)은 틀린 식 같음. (6)으로 수정.

2. Influential Recommender System

Influential Recommender Network (IRN)

▼ 6) Output Probability



Next item

$$p\left(i_j \mid i_{<j}, i_t, u\right) = \text{softmax}\left(\mathbf{W}^p \mathbf{h}_{j-1}^L\right) \quad (7)$$

- $\mathbf{W}^p \in \mathbb{R}^{|I| \times d}$: projection matrix
- IRN's Recommendation: probability dist. $p\left(i_j \mid i_{<j}, i_t, u\right)$ → item generating → influence path. at each step.

2. Influential Recommender System

Influential Recommender Network (IRN)

- Learning: Loss를 minimize하는 parameter
- Inference: below algorithm iteration

Algorithm

- Assumption: Simplicity를 위해, user가 모든 recommendations을 accept한다고 가정
  - 원래는 user가 recommendation을 보고 item accept or reject를 결정
- $\mathcal{F}$: Recommender function → learning 필요.
- s_p: 초기엔 empty set → 점차 채워짐.

Algorithm 1 Generate influence path s_p for user u

Input: Interaction history s_h, the objective item i_t, the maximum path length M
Output: influence path s_p
s_p = [ ]
while $|s_p| \leq M$ **do**
    $i = \mathcal{F}(s_h, i_t, s_p)$
    $s_p = s_p + i$
    if i is i_t **then**
        break
    end if
end while

학습이 완료된 후, 최종 $\mathcal{F}$를 algorithm 1에 사용함.

7) Objective Function

for user u,

- sequence of items $s = \{i_1, \ldots, i_m\} \in S_u$
- If actual path $s \rightarrow P(s|i_t, u)$: maximized
- $\mathrm{PPL}(s|i_t, u) := P(s|i_t, u)^{-\frac{1}{m}}$: minimized

$$\mathrm{PPL}(i_1, \ldots, i_m|i_t, u) = \left(\prod_{j=1}^{m} P(i_j|i_{<j}, i_t, u)\right)^{-\frac{1}{m}} \qquad (8)$$

- Computing을 위해 $\mathrm{PPL}(s|i_t, u) \downarrow \Rightarrow \log \mathrm{PPL}(s|i_t, u) \downarrow$를 사용하여,

$$\log\mathrm{PPL}(i_1, \ldots, i_m|i_t, u) = -\frac{1}{m}\sum_{j=1}^{m} \log P(i_j|i_{<j}, i_t, u) \text{ : minized}$$

- 따라서 Loss function으로 Cross-Entropy loss 사용.

$$\mathcal{L} = \frac{1}{|S|}\sum_{u \in U}\sum_{s \in S_u}\left(\sum_{j=1}^{m} \log P(i_j|i_{<j}, i_t, u)\right) \qquad (9)$$

  - $S = \cup_{u=1}^{|U|} S_u$이므로, 모든 user가 고려됨.
  - Each user: multiple sequences 가능.

3. Evaluation Metrics

Evaluation Metrics

- **Success Rate** (SR_M) measures the ratio of generated s_p's that can successfully reach the objective item i_t within the maximum length M (as denoted in Algorithm 1):

$$SR_M = \frac{1}{|U|} \sum_{u=1}^{|U|} \mathbb{1}(i_t^u \in s_p^u) \qquad (11)$$

- **Increase of Interest** (IoI_M) measures the change of user's interest in i_t after being persuaded through s_p:

$$IoI_M = \frac{1}{|U|} \sum_{u=1}^{|U|} \Big(\log P(i_t^u | s_h^u \oplus s_p^u) - \log P(i_t^u | s_h^u) \Big) \qquad (12)$$

where "$\oplus$" refers to the concatenation of two sequences;

Target item이 influence path에 포함된 user의 성공비율

Influence path로 인한 target item의 user interest 변화량. [prob. ~ interest]

DGIST DATA SCIENCE LAB

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

3. Evaluation Metrics

Evaluation Metrics

- **Increment of Rank** (IoR_M) measures the increase of the ranking for i_t after being persuaded through s_p:

$$IoR_M = \frac{1}{|U|}\sum_{u=1}^{|U|} -\Big(R(i_t^u|s_h^u \oplus s_p^u) - R(i_t^u|s_h^u)\Big) \quad (13)$$

where $R(i_t^u|s_h^u)$ denotes the ranking of the objective item i_t^u based on $P(i_t^u|s_h^u)$.

- Target item이 influence path에 포함될 때, target item의 ranking 증가량.
- R값이 작아진 것 == ranking 증가

3. Evaluation Metrics

Evaluation Metrics

- **Perplexity (PPL)** measures the naturalness and smoothness of the influence path s_p, that is, how likely the path can appear in the viewing history. In our experiment, PPL is defined as the conditional probability of that s_p follows s_h:

$$\mathrm{PPL}(s_p|s_h) = \left(\prod_{k=1}^{|s_p|} P(i_k|s_h \oplus i_{<k})\right)^{-\frac{1}{|s_p|}}$$

$$\log(\mathrm{PPL}) = \frac{1}{|U|}\sum_{u=1}^{|U|} \log \mathrm{PPL}(s_p^u|s_h^u) = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{k=1}^{|s_p^u|} \log P(i_k^u|s_h^u \oplus i_{<k}^u) \quad (14)$$

- Influence path의 smothness

3. Evaluation Metrics

Evaluation Metrics

performance of the evaluator using **HR@20** (Hit Ratio) and **MRR** (Mean Reciprocal Rank):

$$HR@20 = \frac{1}{|U|}\sum_{u=1}^{|U|} \mathbb{1}\big(R(i_q^u|s_h^u) \leq 20\big)$$
$$MRR = \frac{1}{|U|}\sum_{u=1}^{|U|} \frac{1}{R(i_q^u|s_h^u)} \quad (18)$$

4. Results

TABLE IV
PERFORMANCE OF IRS IN TERMS OF NEXT-ITEM RECOMMENDATION.

Method \ Dataset		Lastfm		Movielens-1M	
		$HR@20$	MRR	$HR@20$	MRR
Next-item RS	GRU4Rec	0.0392	0.0081	0.243	0.063
	Caser	0.0460	0.0106	0.252	0.062
	SASRec	0.0445	0.0097	0.257	0.062
	Bert4Rec	0.0489	0.0199	0.264	0.081
IRS	GRU4Rec	0.0387	0.0079	0.198	0.047
	Caser	0.0430	0.0102	0.250	0.061
	SASRec	0.0437	0.0092	0.207	0.041
	IRN	0.0458	0.0190	0.242	0.073