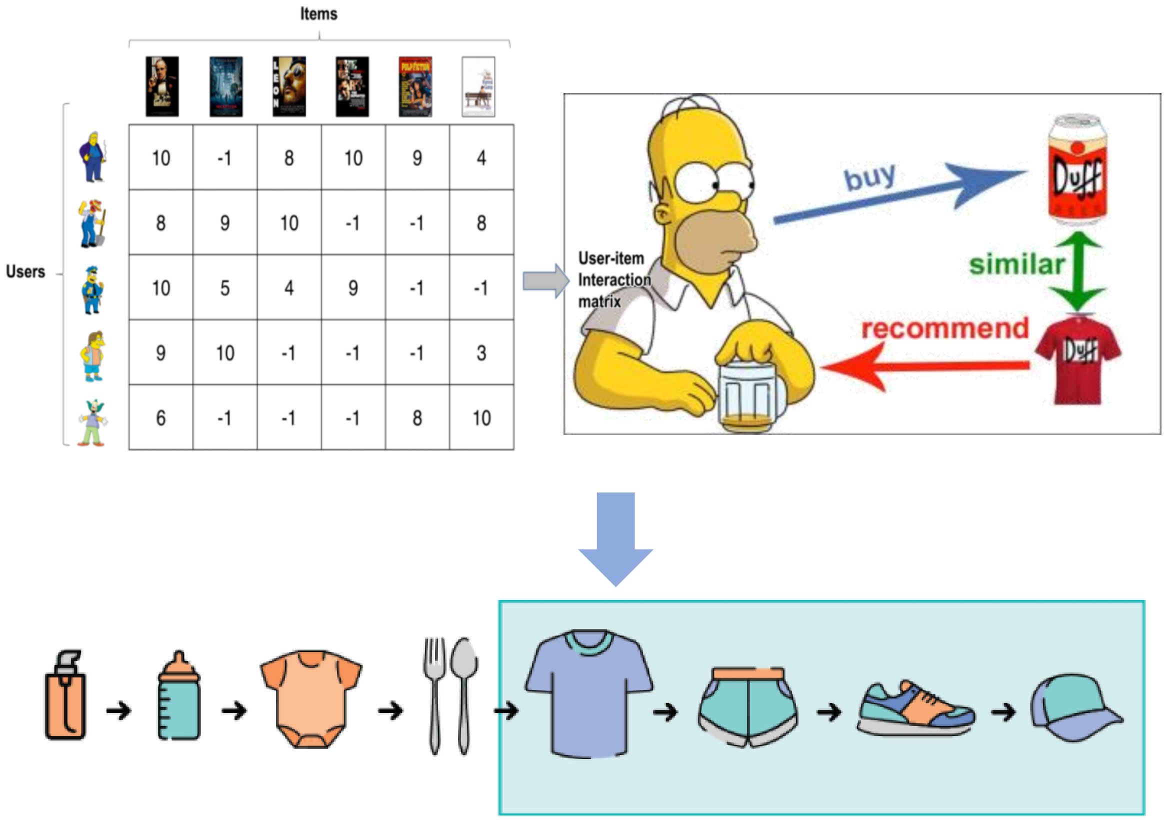
Session-based Recommender Systems

Jihwan Lee 2022-04-01

Introduction
Session-based recommendations



[fig 01] Difference between CFRS and SBRS

- Content-based and CF Underly assumption that all of the historical interactions of a user are equally important to her current preference.
- Limitation

  [1] time-sensitive context (e.g., the recently viewed or purchased items)

  [2] Dynamic user's preference for items

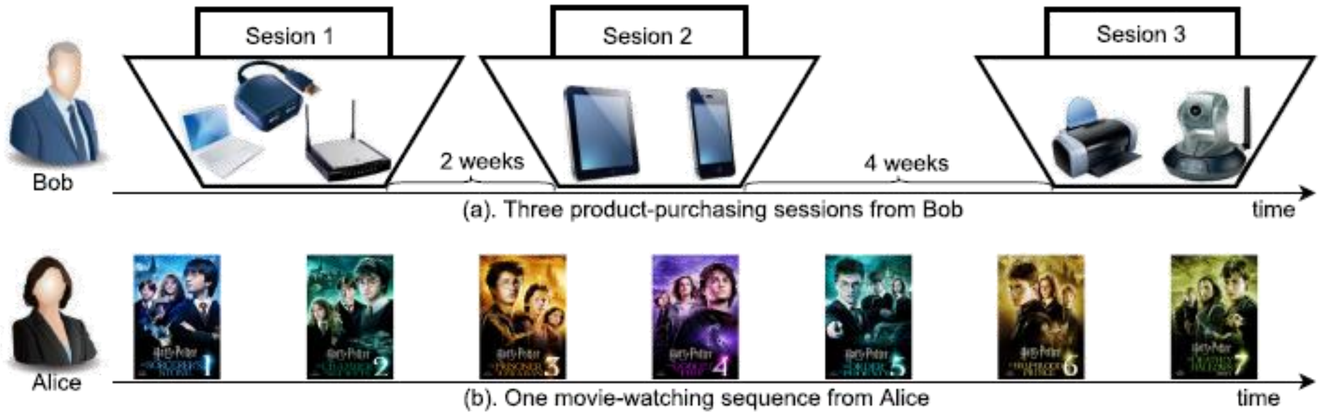
  → Session-based Recommender Systems(SBRSs) have emerged with increasing attention in recent years.

Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

Introduction
SRS vs SBRS

Session Example

Sequence Example

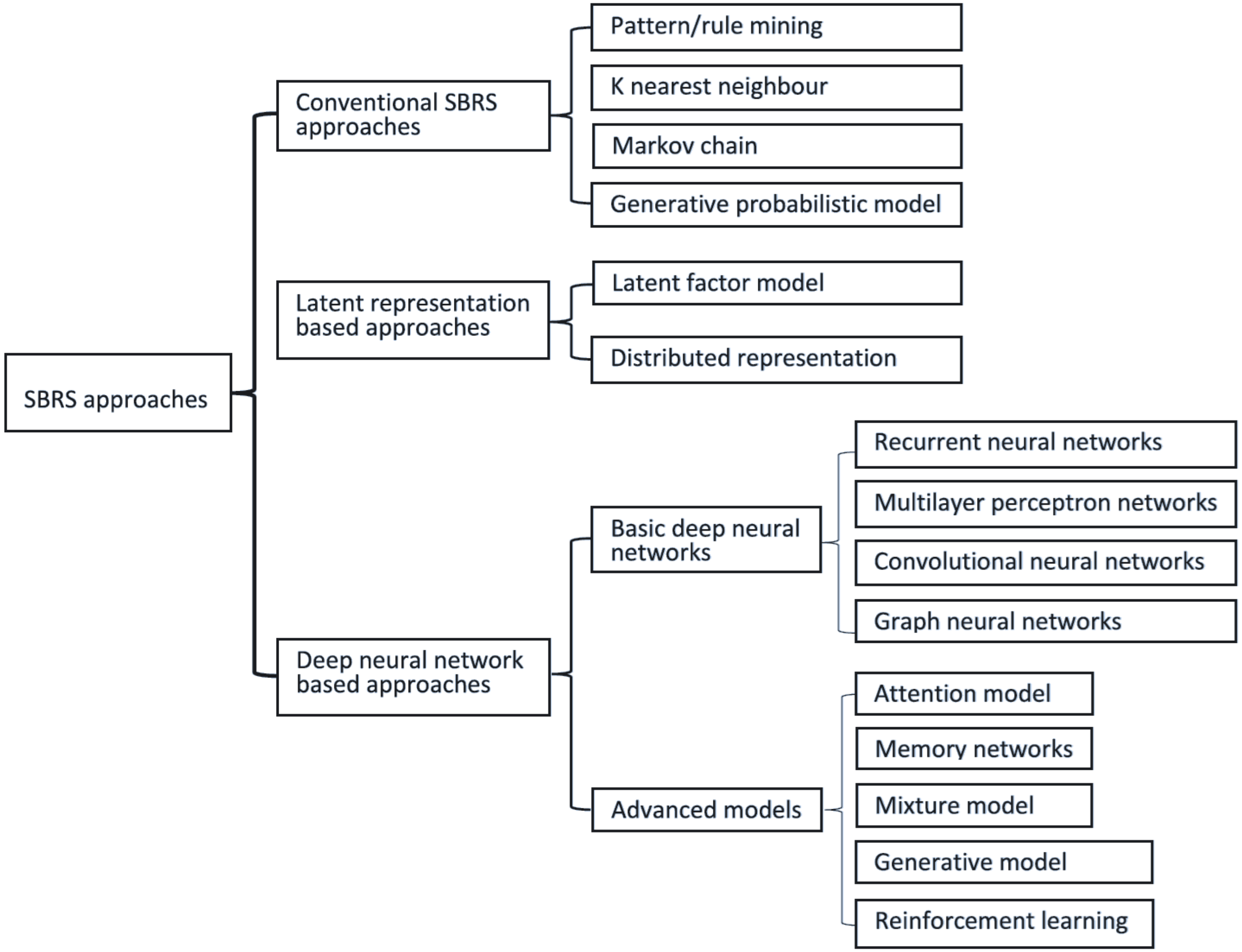


Data Type		Boundary	Order	Time Interval	Main relations embedded	Prediction
Session Data	Unordered session	Multiple	No	Non-identical	Co-occurrence-based dependencies	Unknown part(e.g.,an item or batch of items), Future session(e.g.,the next-basket)
	Ordered session	Multiple	Yes	Non-identical	Co-occurrence-based dependencies and sequential dependencies	
Sqeuence data		Single	Yes	Not included	Seqeuntial dependencies	Future item

[fig 02] Session data vs Sequence data

Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

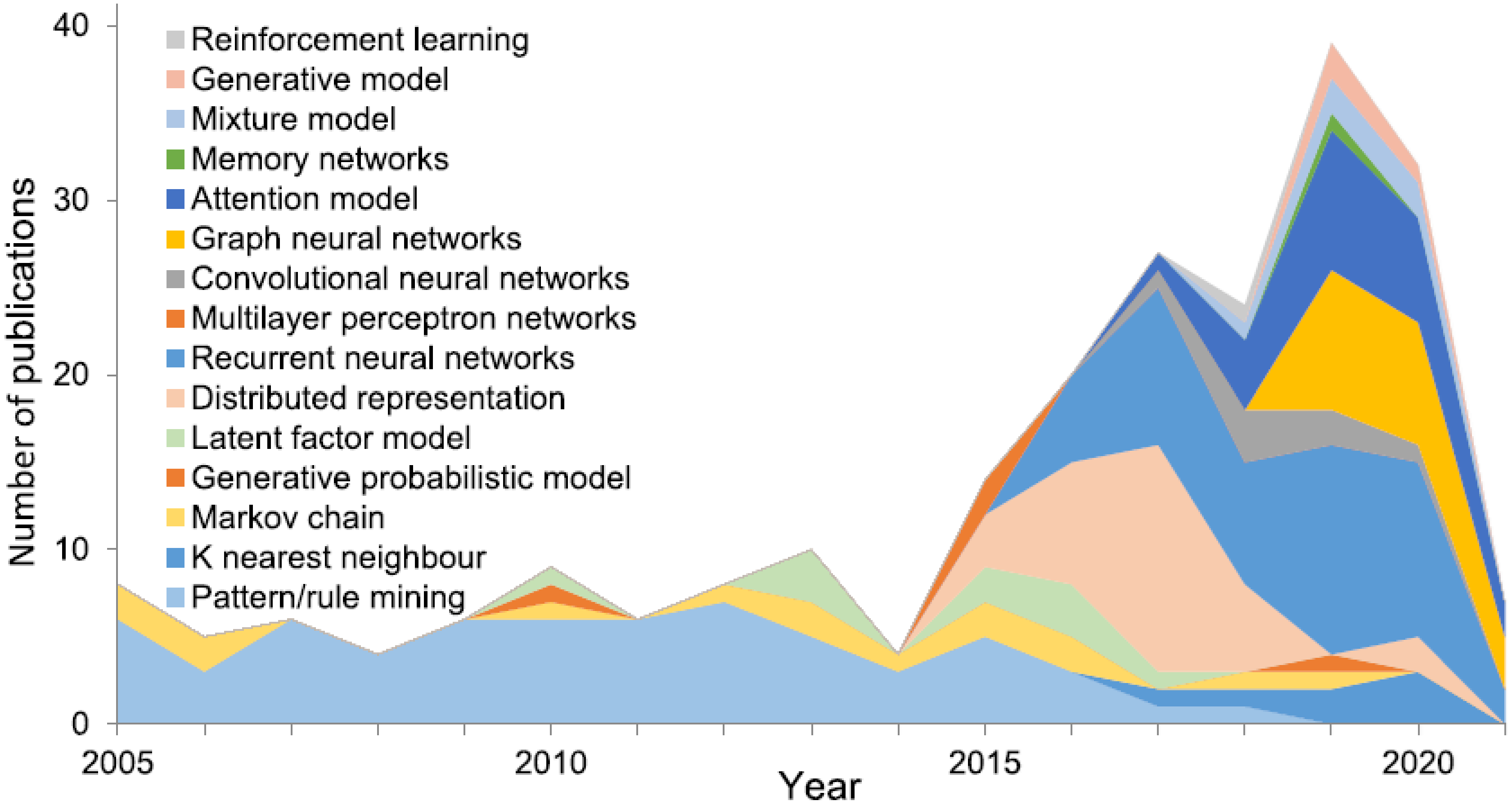
Introduction
Taxonomy



Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

Introduction

Number of the publications on each class of SBRS



Date : 20 March, 2021 (google scholar)

Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

Conventional SBRS

K nearest Neighbor-based SBRSs

[1] Item-KNN (IKNN)

[2] Session-KNN (SKNN)

Item-KNN Example (IKNN)

Session #	Session content (ordered list of item IDs)
Session 1	[Item 12, Item 23, Item 7, Item 562, Item 346, Item 85]
Session 2	[Item 23, Item 65, Item 12, Item 3, Item 9, Item 248]
Session 3	[Item 248, Item 12, Item 7, Item 9, Item 346]
Session 4	[Item 85, Item 65, Item 248, Item 12, Item 346, Item 9]
Session 5	[Item 346, Item 7, Item 9, Item 3, Item 12]

Current Session

Session #	3	7	9	12	23	65	85	248	346	562
Session 1	0	1	0	1	1	0	1	0	1	1
Session 2	1	0	1	1	1	1	0	1	0	0
Session 3	0	1	1	1	0	0	0	1	1	0
Session 4	0	0	1	1	0	1	1	1	1	0
Session 5	1	1	1	1	0	0	0	0	1	0

- Each item is encoded into a binary vector where each element in-dicates whether the item occurs (set to “1”) in a specific session or not(set to “0”).

https://livebook.manning.com/book/graph-powered-machine-learning/chapter-6/v-8/90

Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

Conventional SBRS

K nearest Neighbor-based SBRSs

Item-KNN Example (IKNN)

Session 5	[Item 346, Item 7, Item 9, Item 3, $\boxed{\text{Item 12}}$

Session #	3	7	9	12	23	65	85	248	346	562
Session 1	0	1	0	1	1	0	1	0	1	1
Session 2	1	0	1	1	1	1	0	1	0	0
Session 3	0	1	1	1	0	0	0	1	1	0
Session 4	0	0	1	1	0	1	1	1	1	0
Session 5	1	1	1	1	0	0	0	0	1	0

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Item 12: [1,1,1,1,1]

Item 23: [1,1,0,0,0] 0.6324

Item 65: [0,1,0,1,0] 0.6324

Item 85: [1,0,0,1,0] 0.6324

$\boxed{\text{Item 248: [0,1,1,1,0]} \quad 0.7745}$

Item 562: [0,1,0,1,0] 0.447

→ Next Recommendation

Item 248

- During the recommendation, only the last item(Item 12) is considered for the current session(Session 5).
- The similarity between items can be calculated on their vectors with a certain similarity measure, like cosine similarity.

https://livebook.manning.com/book/graph-powered-machine-learning/chapter-6/v-8/90

Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

Conventional SBRS

K nearest Neighbor-based SBRSs

Session-KNN Example (SKNN)

Session #	Session content (ordered list of item IDs)
Session 1	[Item 12, Item 23, Item 7, Item 562, Item 346, Item 85]
Session 2	[Item 23, Item 65, Item 12, Item 3, Item 9, Item 248]
Session 3	[Item 248, Item 12, Item 7, Item 9, Item 346]
Session 4	[Item 85, Item 65, Item 248, Item 12, Item 346, Item 9]
Session 5	[Item 346, Item 7, Item 9, Item 3, Item 12]

Session #	Session content (ordered list of item IDs)
Session 1	[Item 12, Item 23, Item 7, Item 562, Item 346, Item 85]
Session 2	[Item 23, Item 65, Item 12, Item 3, Item 9, Item 248]
Session 3	[Item 248, Item 12, Item 7, Item 9, Item 346]
Session 4	[Item 85, Item 65, Item 248, Item 12, Item 346, Item 9]
Session 5	[Item 346, Item 7, Item 9, Item 3, Item 12]

[fig 03] Left: IKNN vs Right: SKNN

- IKNN considers only the last event(e.g., ordered item 12) in current Session, the SKNN method compares the entire current session with the past sessions in the training data to determine the items to be recommended.

Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

Conventional SBRS

K nearest Neighbor-based SBRSs

Session-KNN Example (SKNN)

Session #	3	7	9	12	23	65	85	248	346	562
Session 1	0	1	0	1	1	0	1	0	1	1
Session 2	1	0	1	1	1	1	0	1	0	0
Session 3	0	1	1	1	0	0	0	1	1	0
Session 4	0	0	1	1	0	1	1	1	1	0
Session 5	1	1	1	1	0	0	0	0	1	0

1. Suppose that the number of sessions is five and K=4

Session #	3	7	9	12	23	65	85	248	346	562
Session 1	0	1	0	1	1	0	1	0	1	1
Session 2	1	0	1	1	1	1	0	1	0	0
Session 3	0	1	1	1	0	0	0	1	1	0
Session 4	0	0	1	1	0	1	1	1	1	0
Session 5	1	1	1	1	0	0	0	0	1	0

0.8

0.5477

0.5477

0.5477

2. Calculate the similarity between each session binary vector by Cosine similarity

(If k=1, Only session 3 is considered when calculating item scores)

Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

Conventional SBRS

K nearest Neighbour-based SBRSs

Session-KNN Example (SKNN)

Session #	3	7	9	12	23	65	85	248	346	562
Session 1	0	1	0	1	1	0	1	0	1	1
Session 2	1	0	1	1	1	1	0	1	0	0
Session 3	0	1	1	1	0	0	0	1	1	0
Session 4	0	0	1	1	0	1	1	1	1	0
Session 5	1	1	1	1	0	0	0	0	1	0

0.8

0.5477

0.5477

0.5477

Session 5	[Item 346, Item 7, Item 9, Item 3, Item 12]

3. Calculate the recommendation score for each item i

Recommendation Score func: $score_{\mathrm{SKNN}}(i, s) = \Sigma_{n \in N_s} sim(s, n) \cdot 1_n(i)$

Ns : Neighbor session, s: current session, 1n(i): returns 1 if session n contains item I and 0 otherwise

score(item 23,session 5)=0.547*0+0.547*1+0.8*1+0.547*1=1.894

score(item 65,session 5)=0.547*1+0.547*1+0.8*0+0.547*0=1.094

score(item 85,session 5)=0.547*0+0.547*1+0.8*0+0.547*1=1.094

score(item 248,session 5)=0.547*0+0.547*1+0.8*1+0.547*1=1.894

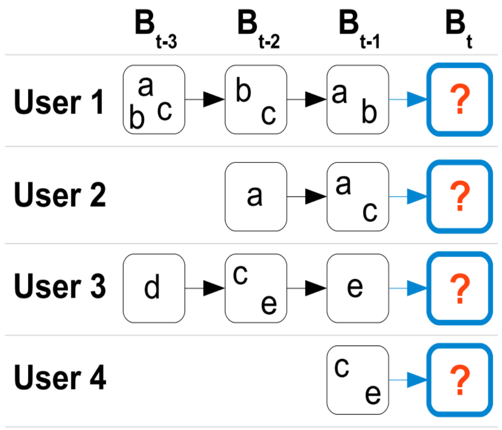
score(item 562,session 5)=0.547*1+0.547*0+0.8*0+0.547*0=0.547

Next Recommendation: Item 23,Item 248

Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

Conventional SBRS

Markov Chain



- $U = \{u_1, \dots, u_{|U|}\}$: users
- $I = \{i_1, \dots, i_{|I|}\}$: items
- $\mathcal{B}^u = (B_1^u, \dots, B_{t_u-1}^u)$: historical baskets of user u with $B_t^u \subseteq I$
- $\mathcal{B} = \{\mathcal{B}^{u_1}, \dots, \mathcal{B}^{u_{|U|}}\}$: purchase history of all users

$$p(i \in B_t | B_{t-1}) := \frac{1}{|B_{t-1}|} \sum_{l \in B_{t-1}} p(i \in B_t | l \in B_{t-1})$$

Transition Matrix

from \ to	a	b	c	d	e	#
a	0.5	$\boxed{0.5}$	1	0	0	2
b	0.5	1	$\boxed{0.5}$	0	0	2
c	0.3	0.7	0.3	0	0.3	3
d	0	0	1	0	1	1
e	0	0	0	0	1	1

Example

$$P(o_1 \to o_2 \to o_3) = P(o_1) * P(o_2|o_1) * P(o_3|o_2). \longrightarrow P(A \to B \to C) = P(A) * P(B|A) * P(C|B)$$

$$= \frac{4}{16} * 0.5 * 0.5 = 0.0625$$

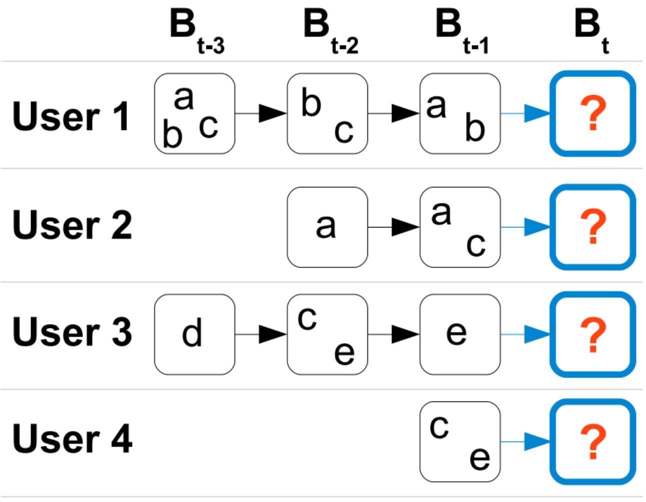
Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

Latent Representation Approaches for SBRSs

Latent Representation Approach for SBRs

Factorized Personalized Markov Chain (FPMC)

Problem of Markov Chain

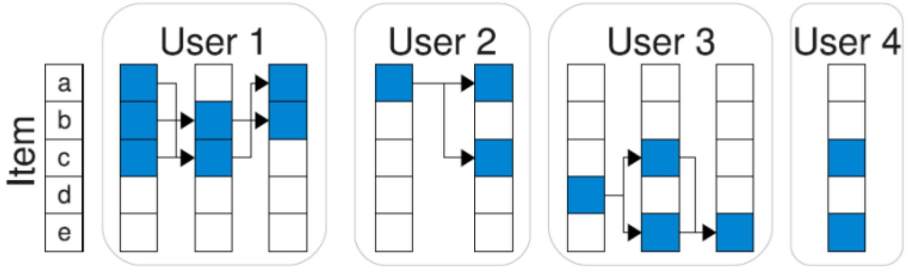


- $U = \{u_1, \ldots, u_{|U|}\}$: users
- $I = \{i_1, \ldots, i_{|I|}\}$: items
- $\mathcal{B}^u = (B_1^u, \ldots, B_{t_u-1}^u)$: historical baskets of user u with $B_t^u \subseteq I$
- $\mathcal{B} = \{\mathcal{B}^{u_1}, \ldots, \mathcal{B}^{u_{|U|}}\}$: purchase history of all users

$$p(i \in B_t | B_{t-1}) := \frac{1}{|B_{t-1}|} \sum_{l \in B_{t-1}} p(i \in B_t | l \in B_{t-1}) = \frac{\hat{p}(i \in B_t \wedge l \in B_{t-1})}{\hat{p}(l \in B_t)}$$

$$= \frac{|(B_t, B_{t-1}) : i \in B_t \wedge l \in B_{t-1}|}{|(B_t, B_{t-1}) : l \in B_{t-1}|}$$

Markov Chain Predict Formula



Transition Matrix

from \ to	a	b	c	d	e	#
a	0.5	0.5	1	0	0	2
b	0.5	1	0.5	0	0	2
c	0.3	0.7	0.3	0	0.3	3
d	0	0	1	0	1	1
e	0	0	0	0	1	1

[fig 04] Non-Personalized Problem in Markov Chain

Rendle, Steffen, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing personalized markov chains for next-basket recommendation." *Proceedings of the 19th international conference on World wide web*. 2010.

Latent Representation Approaches for SBRSs

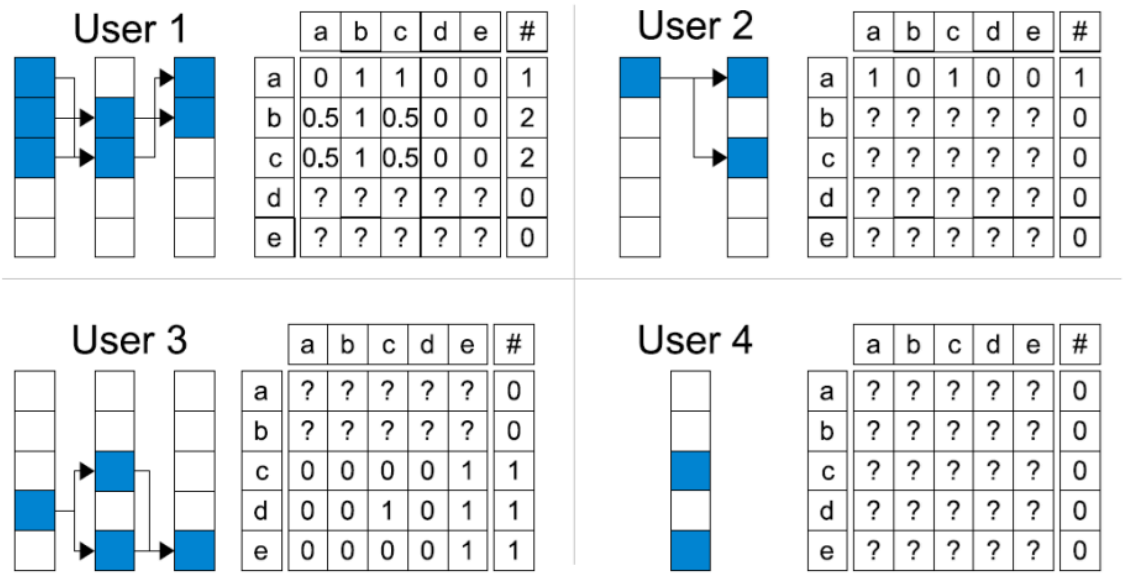
Latent Representation Approach for SBRs

Factorized Personalized Markov Chain (FPMC)

Problem of Markov Chain

$$p(i \in B_t | B_{t-1}) := \frac{1}{|B_{t-1}|} \sum_{l \in B_{t-1}} p(i \in B_t | l \in B_{t-1}) \longrightarrow p(i \in B_t^u | B_{t-1}^u) := \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} p(i \in B_t^u | l \in B_{t-1}^u)$$

change the non-personalized formula to user-specific formula, so predicition depends only on the users transitions.



User 1

	a	b	c	d	e	#
a	0	1	1	0	0	1
b	0.5	1	0.5	0	0	2
c	0.5	1	0.5	0	0	2
d	?	?	?	?	?	0
e	?	?	?	?	?	0

User 2

	a	b	c	d	e	#
a	1	0	1	0	0	1
b	?	?	?	?	?	0
c	?	?	?	?	?	0
d	?	?	?	?	?	0
e	?	?	?	?	?	0

User 3

	a	b	c	d	e	#
a	?	?	?	?	?	0
b	?	?	?	?	?	0
c	0	0	0	0	1	1
d	0	0	1	0	1	1
e	0	0	0	0	1	1

User 4

	a	b	c	d	e	#
a	?	?	?	?	?	0
b	?	?	?	?	?	0
c	?	?	?	?	?	0
d	?	?	?	?	?	0
e	?	?	?	?	?	0

[fig 05] **Personalized Markov Chains**

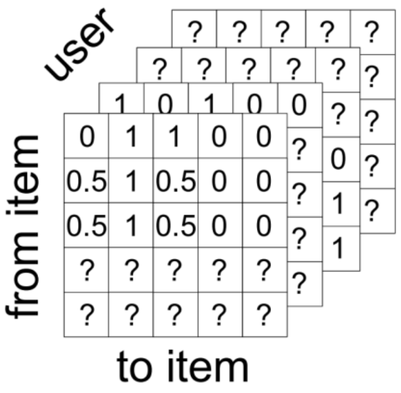
- **Problems of personalized ML-Estimation:**
  [1] Many of the parmameters cannot be estimated.
  [2] Theoretical properties are easily fail since the data is extremely sparse.

Rendle, Steffen, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing personalized markov chains for next-basket recommendation." *Proceedings of the 19th international conference on World wide web*. 2010.

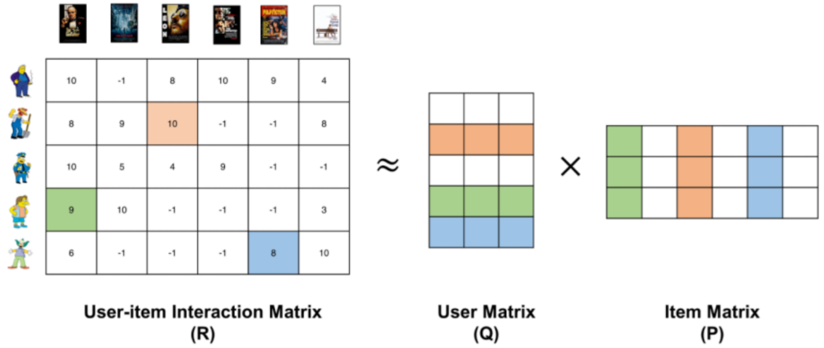
Latent Representation Approaches for SBRSs

Latent Representation Approach for SBRs

Factorized Personalized Markov Chain (FPMC)



“Personalized trainsition cube”



Idea (MF)

$$p(i \in B_t^u | B_{t-1}^u) := \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} p(i \in B_t^u | l \in B_{t-1}^u)$$

$$\boxed{\langle v_u^{U,I}, v_i^{I,U} \rangle + \langle v_i^{I,L}, v_l^{L,I} \rangle + \langle v_u^{U,L}, v_l^{L,U} \rangle}$$

$$\hat{p}(i \in B_t^u | B_{t-1}^u) = \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} \left(\langle v_u^{U,I}, v_i^{I,U} \rangle + \langle v_i^{I,L}, v_l^{L,I} \rangle + \langle v_u^{U,L}, v_l^{L,U} \rangle \right)$$

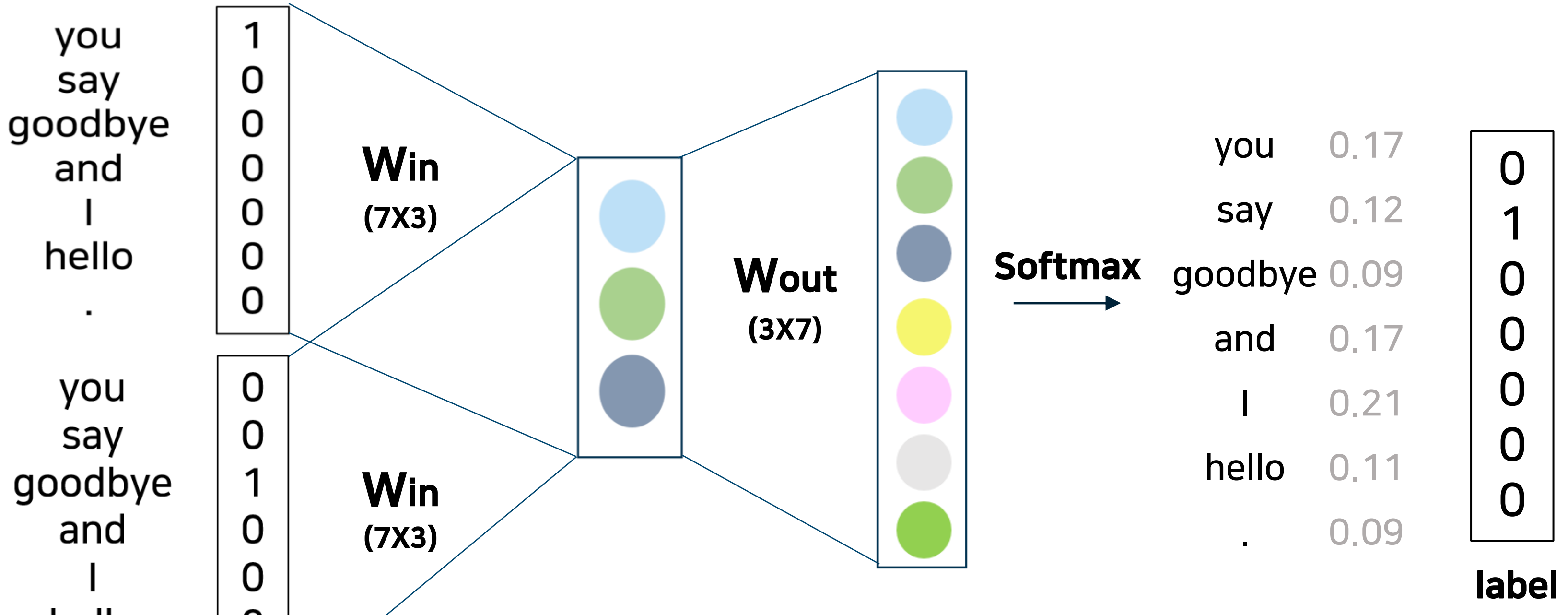
FPMC= MF + MC

Rendle, Steffen, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing personalized markov chains for next-basket recommendation." *Proceedings of the 19th international conference on World wide web*. 2010.

Latent Representation Approaches for SBRSs

Distributed Representation-based SBRSs

Session-based Wide-In-Wide-Out (SWIWO)



[fig 06] CBOW Model

CBOW

You $\boxed{?}$ goodbye and I say hello

$$\text{Loss} = -\sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i$$

- The SWIWO model is inspired by Skip-gram and CBOW model.
- Skip-gram and CBOW is the representative methods of Word2Vec in Natural Language Processing.

Hu, Liang, et al. "Diversifying Personalized Recommendation with User-session Context." IJCAI. 2017.

Latent Representation Approaches for SBRSs

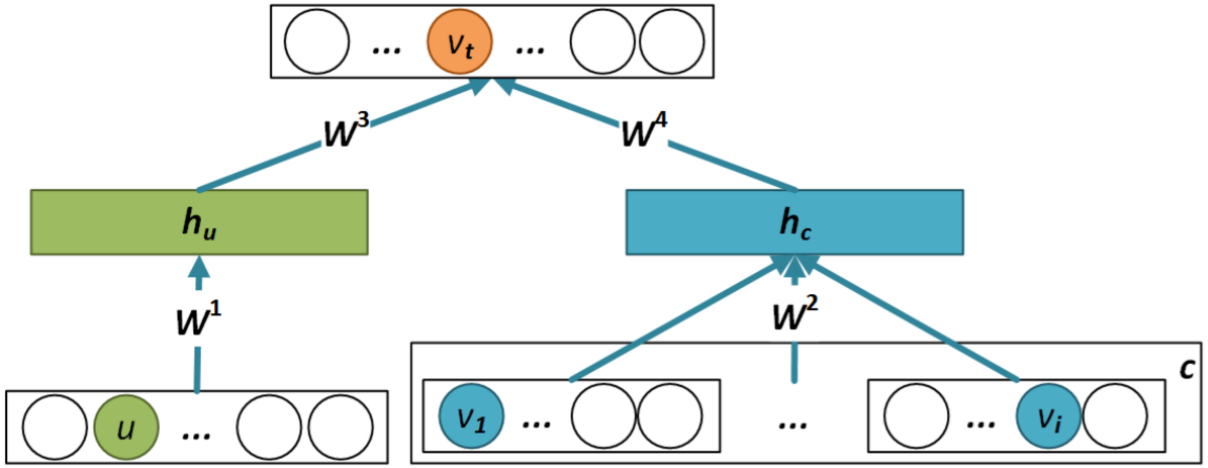
Distributed Representation-based SBRSs

Session-based WIWO (SWIWO)

- MF and FPMC easily suffers from the data sparsity issue.
- SWIWO propose to diversify personalized recommendation results according to user-session contexts.
- SWIWO networks are constructed and trained as probabilistic classifiers that learn to predict a conditional probability distribution.

$P(v_t|\mathbf{c})$ where $\mathbf{c} \subseteq s$

c: context s:session vt: item



[fig 07] SWIWO Architecture

Hu, Liang, et al. "Diversifying Personalized Recommendation with User-session Context." IJCAI. 2017.

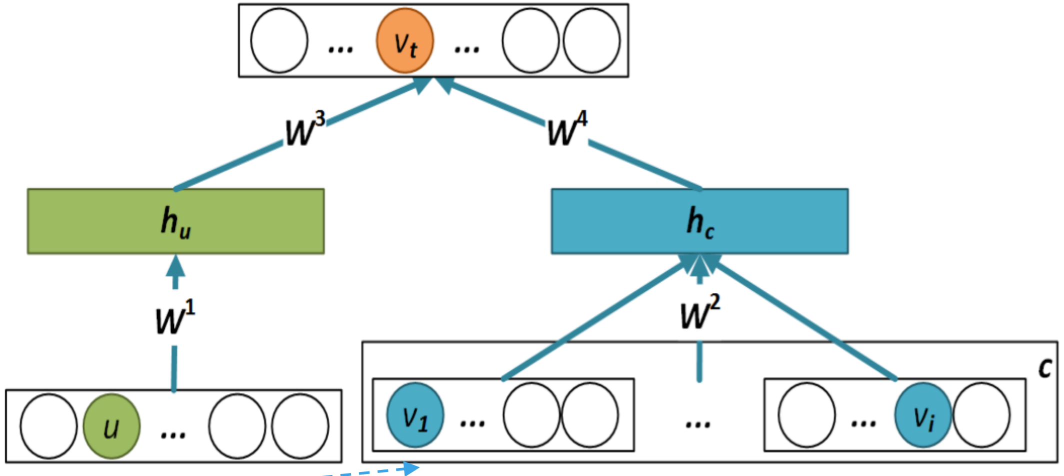
Latent Representation Approaches for SBRSs

Distributed Representation-based SBRSs

Session-based WIWO (SWIWO)

1. Define Formula

$$\mathbf{U} = \{u_1, u_2, \cdots, u_{|\mathbf{U}|}\} \text{ user set}$$
$$\mathbf{V} = \{v_1, v_2, \cdots, v_{|\mathbf{V}|}\} \text{item set}$$
$$\mathbf{S} = \{s_1, s_2, \cdots, s_{|\mathbf{S}|}\} \text{session set}$$



2. Create an embedding layer to map those sparse one-hot user and item vectors

e.g., ui: |U|x1 dimension one-hot vector

vi: |V|x1 dimension one-hot vector

3. Define context window size (e.g., predict Vt score & window size=k)

$$s = \{v_{t-k}, \cdots, v_{t+k}\} \text{ session}$$
$$\mathbf{c} = \{v_{t-k}, \cdots, v_{t-1}, v_{t+1}, \cdots, v_{t+k}\} \text{context}$$

Item List

S=	V1	...	Vt-k	Vt-k	Vt	...	Vt+k	Vt+k+1

C=	V1	...	Vt-k	Vt-k	...	Vt+k	Vt+k+1

Hu, Liang, et al. "Diversifying Personalized Recommendation with User-session Context." IJCAI. 2017.

Latent Representation Approaches for SBRSs

Distributed Representation-based SBRSs

Session-based WIWO (SWIWO)

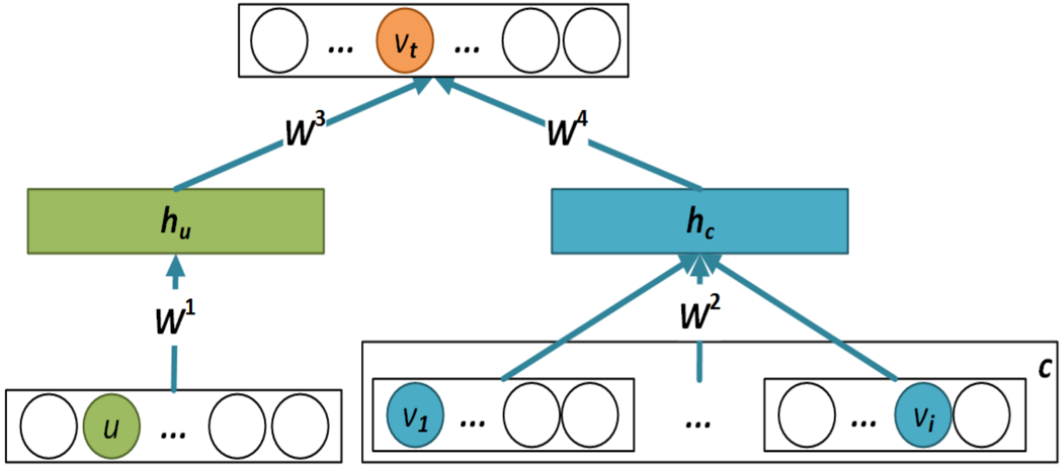
4. Hidden Layer

$$\mathbf{h}_u = \sigma\left(\mathbf{W}^1_{:,u}\right)$$

$$\mathbf{h}_v = \sigma\left(\mathbf{W}^2_{:,v}\right)$$

$$\mathbf{h}_c = \sum_{v \in \mathbf{c}} w_v \mathbf{h}_v \equiv \sum_{v \in \mathbf{c}} w_v \sigma\left(\mathbf{W}^2_{:,v}\right)$$

$$w_v \propto \exp[-\lambda(|v - t| - 1)]$$



5. Compute vt score

$$S_{v_t}(u, \mathbf{c}) = \mathbf{W}^3_{t,:}\mathbf{h}_u + \mathbf{W}^4_{t,:}\mathbf{h}_c$$

6. Compute the conditional distribution with softmax function

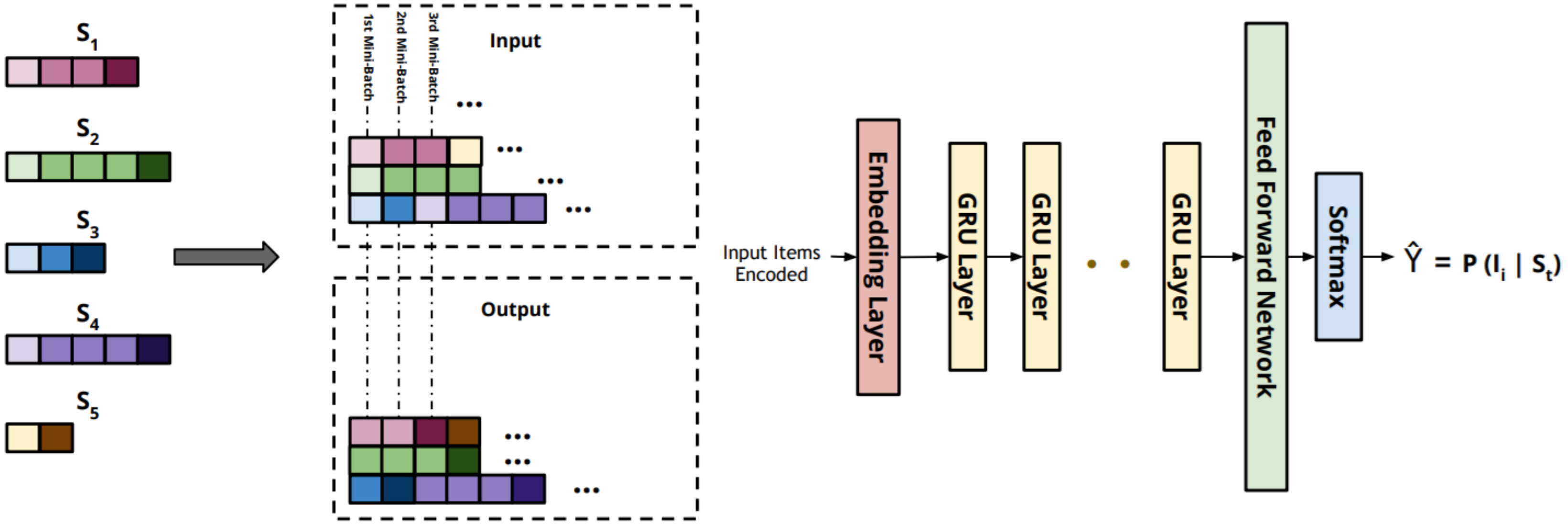
$$P_\Theta(v_t|u, \mathbf{c}) = \frac{\exp(S_{v_t}(u, \mathbf{c}))}{Z(u, \mathbf{c})} \quad Z(u, \mathbf{c}) = \sum_{v \in \mathbf{V}} \exp(S_{v_t}(u, \mathbf{c}))$$

Hu, Liang, et al. "Diversifying Personalized Recommendation with User-session Context." IJCAI. 2017.

Deep Neural Network Approaches for SBRs

Recurrent Neural Networks

GRU4Rec - Architecture



[fig 08] GRU4Rec Architecture

- Input: actual state of the session (item) → Embedding: one-hot encoding (length: the number of items)
- Output: score on items for being the next in the event stream
- GRU based RNN (RNN is worse, LSTM is slower)

Hidasi, Balázs, et al. "Session-based recommendations with recurrent neural networks." *arXiv preprint arXiv:1511.06939* (2015).

Deep Neural Network Approaches for SBRs

Recurrent Neural Networks

GRU4Rec - Minibatch

- **Motivation**

  High variance in the length of sessions (from 2

  100s of events)

  The goal is to capture how sessions evolve

- **Minibatch**

  Input: current events

  Output: next events

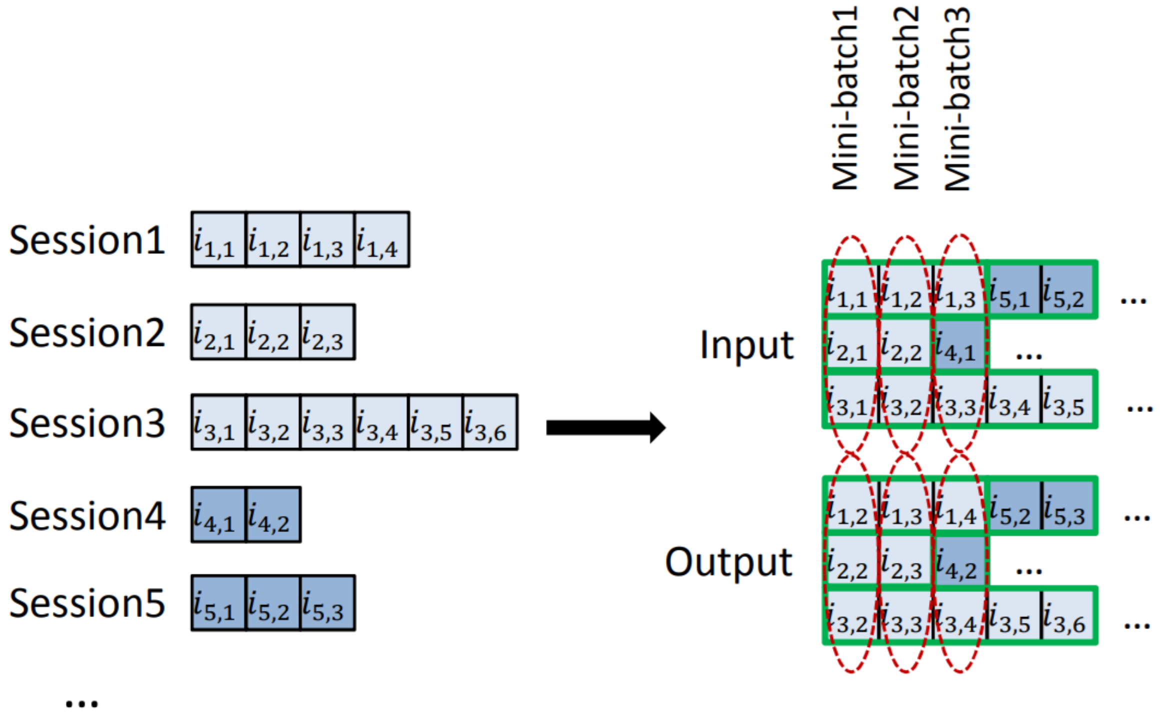
- **Active sessions**

  First X

  Finished sessions replaced by the next available

  Sessions are assumed to be independent, thus reset

  the appropriate hidden state when session switch occurs



[fig 09] GRU4Rec Minibatch

Hidasi, Balázs, et al. "Session-based recommendations with recurrent neural networks." *arXiv preprint arXiv:1511.06939* (2015).

Deep Neural Network Approaches for SBRs

Recurrent Neural Networks

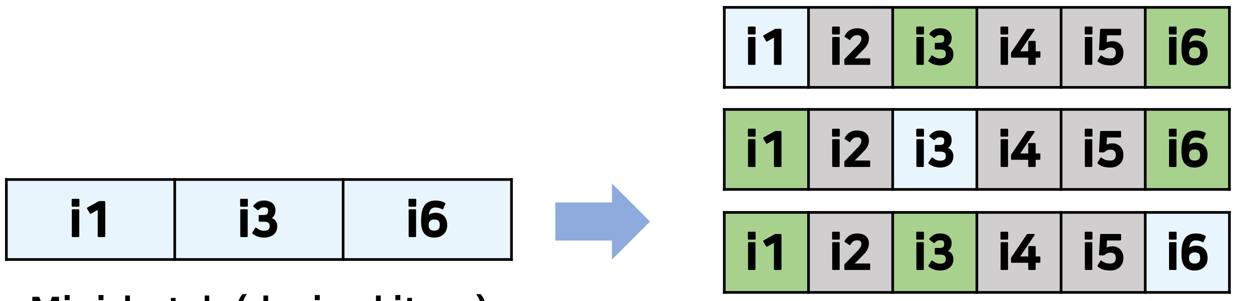
GRU4Rec – Sampling the output Example

- **Motivation**

  Calculate a score for each item is very costly → effective negative sampling

- **Sampling negative items**

  Assume that the user didn't interact some items because the user didn't like it → Consider this popularity when sampling → It can further reduce computational times by skipping the sampling.



Mini-batch (desired item)

Skyblue: Positive Item(target item)
Gray: Inactive outputs (not computed)
Green: Sampled Negative items

$$L_s = -\frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \log\left(\sigma\left(\boxed{\hat{r}_{s,i}} - \boxed{\hat{r}_{s,j}}\right)\right)$$

BPR Loss Function

Hidasi, Balázs, et al. "Session-based recommendations with recurrent neural networks." *arXiv preprint arXiv:1511.06939* (2015).

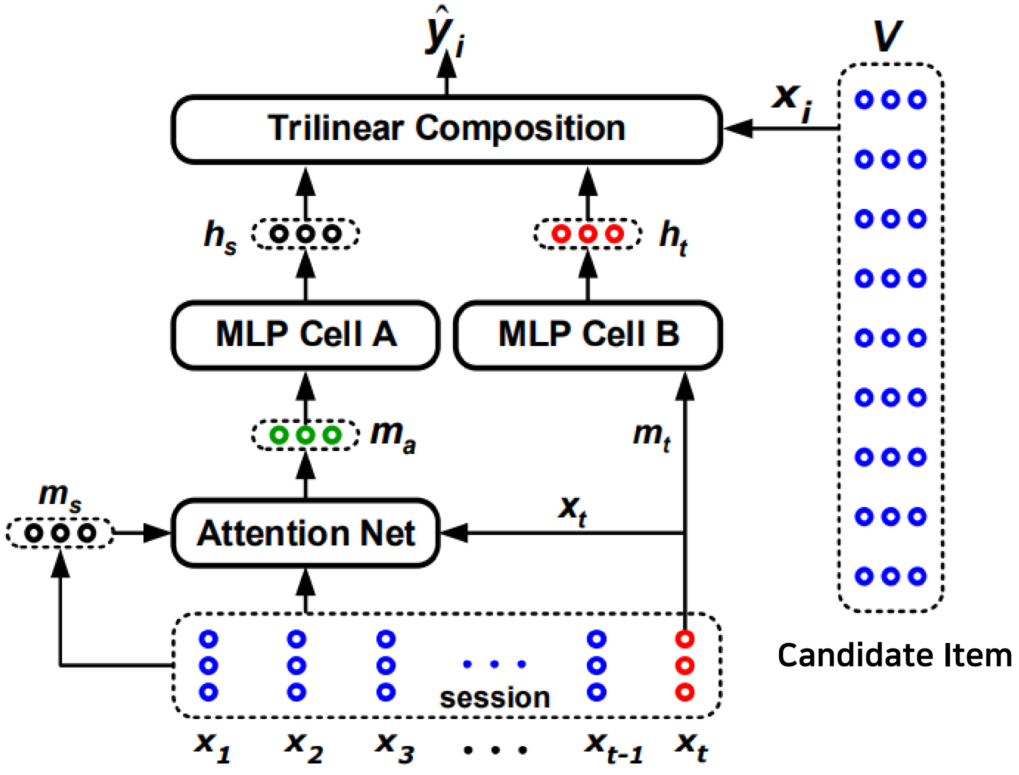
Deep Neural Network Approaches for SBRs

Attention + MLP

STAMP (Short-Term Attention/Memory Priority Model)

- STAMP model captures not only user's general interests from the long-term memory of a session context, but also taking into account user's current interests from the short-term memory of the last-clicks.

- Model architecture is simple, but achieves state-of-the-art performance for three benchmark datasets.



[fig 10] STAMP Architecture

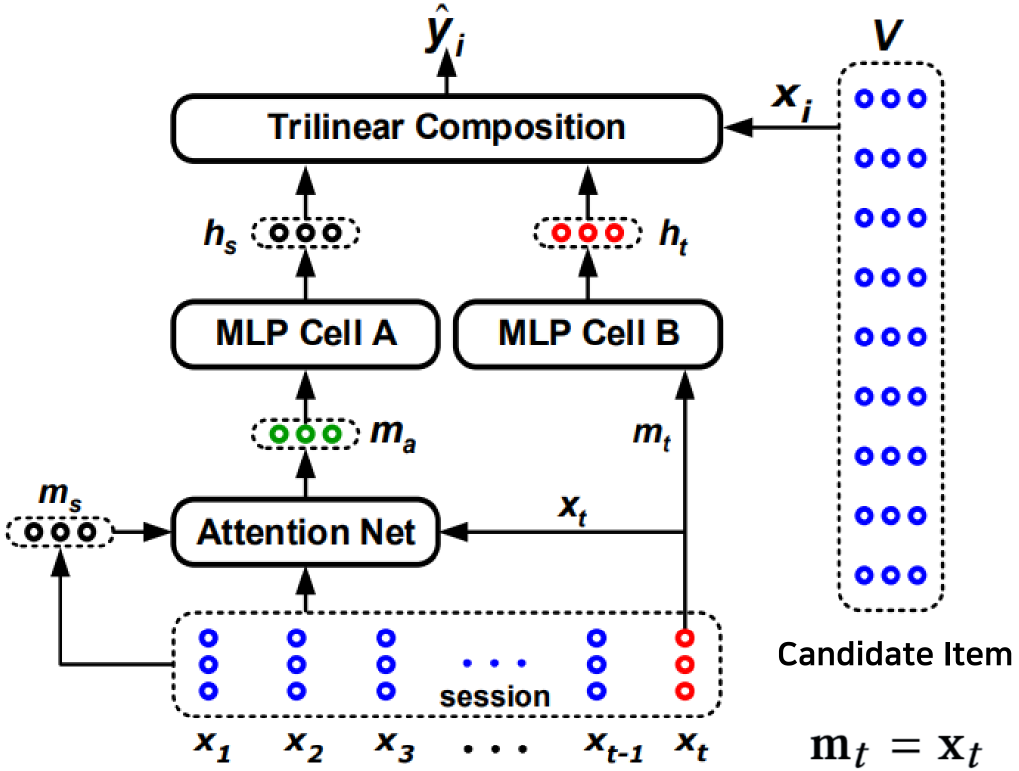
Liu, Qiao, et al. "STAMP: short-term attention/memory priority model for session-based recommendation." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2018.

Deep Neural Network Approaches for SBRs

Attention + MLP

STAMP (Short-Term Attention/Memory Priority Model)

1. External memory of the session (ms): the user's interests in general with respect to current session
$$\mathbf{m}_s = \frac{1}{t}\sum_{i=1}^{t}\mathbf{x}_i$$
2. Attention composite function (αi)
$$\alpha_i = \mathbf{W}_0\sigma(\mathbf{W}_1\mathbf{x}_i + \mathbf{W}_2\mathbf{x}_t + \mathbf{W}_3\mathbf{m}_s + \mathbf{b}_a)$$
Obtain the attention coefficients vector
$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_t)$$
3. ma: Attention based user's interests
$$\mathbf{m}_a = \sum_{i=1}^{t}\alpha_i\mathbf{x}_i$$
4. Hidden state & unnormalized cosine similarity
$$\mathbf{h}_s = f(\mathbf{W}_s\mathbf{m}_a + \mathbf{b}_s) \quad \mathbf{h}_t = f(\mathbf{W}_t\mathbf{x}_t + \mathbf{b}_t) \quad \hat{\mathbf{z}}_i = \sigma(<\mathbf{h}_s, \mathbf{h}_t, \mathbf{x}_i>)$$
5. output (next-click probabilitiy for item vi)
$$\hat{\mathbf{y}} = softmax(\hat{\mathbf{z}})$$



$$\mathbf{m}_t = \mathbf{x}_t$$

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|V|}\}$$ X: item(v) embedding vector sets

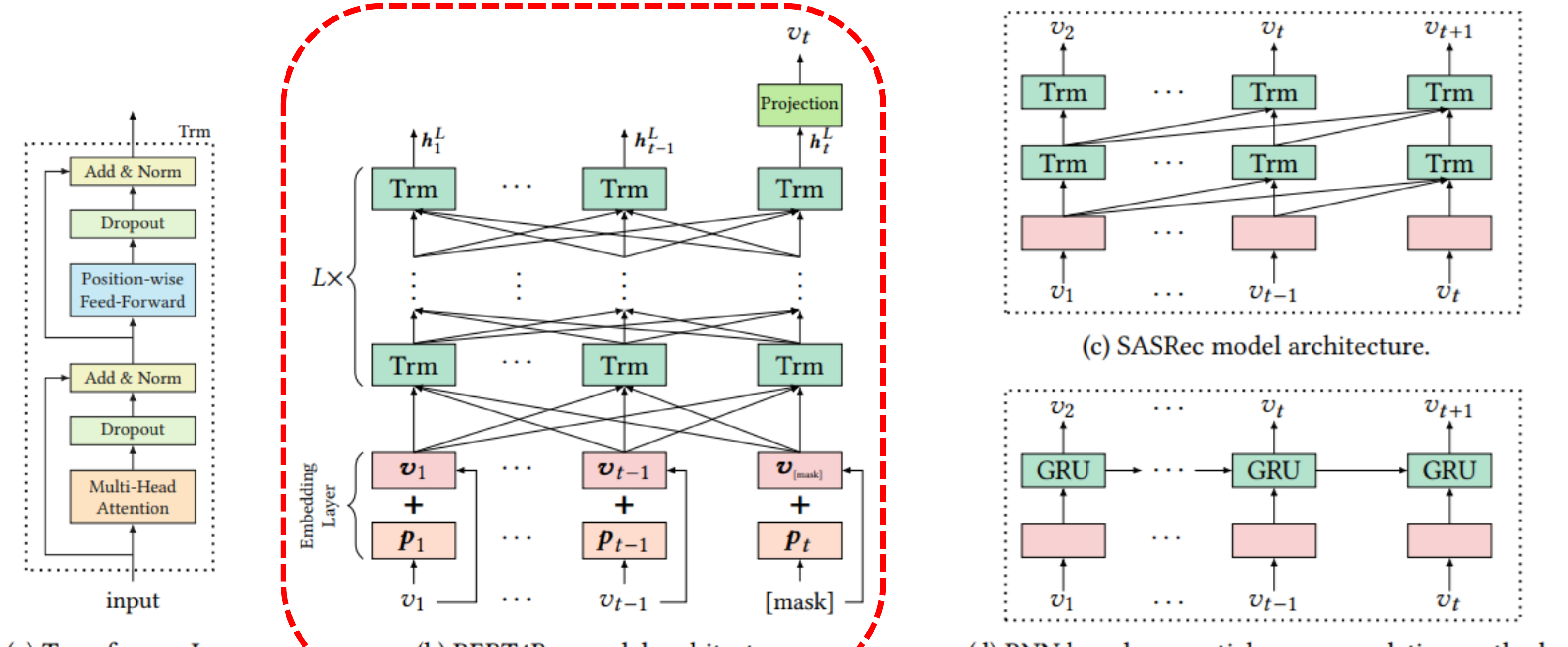
[fig] STAMP Architecture

Liu, Qiao, et al. "STAMP: short-term attention/memory priority model for session-based recommendation." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018.

Deep Neural Network Approaches for SBRs

BERT (self-attention)

BERT4REC



(a) Transformer Layer. (b) BERT4Rec model architecture. (c) SASRec model architecture. (d) RNN based sequential recommendation methods.

Liu, Qiao, et al. "STAMP: short-term attention/memory priority model for session-based recommendation." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018.

Deep Neural Network Approaches for SBRs

BERT (self-attention)

BERT4REC Example

1. Embedding layer $\boldsymbol{h}_i^0 = \boldsymbol{v}_i + \boldsymbol{p}_i$

- Example : user 01 sequential items ={i3,i9,i12,i15,i18}

  embdedding window = W (hyperparameter)

  Item list length (N) > W: recently purchase N

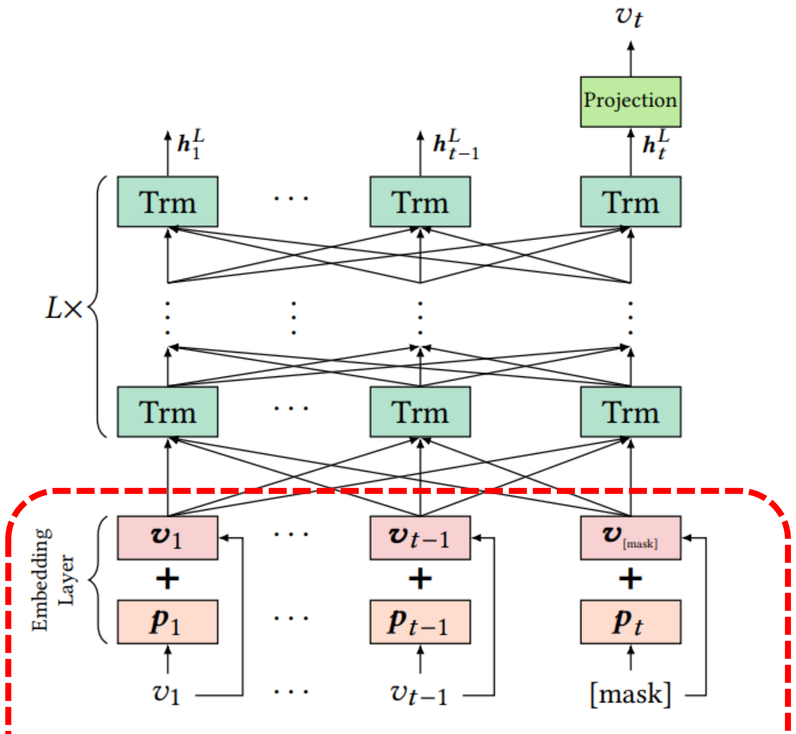
  Item list length (N) < W : zero-padding

  ex) if W=4, i3,i9,i12,i15,i18

  if W=6 0, i3,i9,i12,i15,i18 zero-padding

- embedding = item embedding + positional embedding

Item embedding	i3	i9	i12	i15	i18
Positional embedding	0	1	2	3	4



(b) BERT4Rec model architecture.

$$p_{i,j} = \begin{cases} \sin\left(\frac{i}{10000^{\frac{j}{d_{emb\_dim}}}}\right) & \text{if } j \text{ is even} \\ \cos\left(\frac{i}{10000^{\frac{j-1}{d_{emb\_dim}}}}\right) & \text{if } j \text{ is odd} \end{cases}$$

Cf) Transformer positional encoding

Sun, Fei, et al. "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer." *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019.

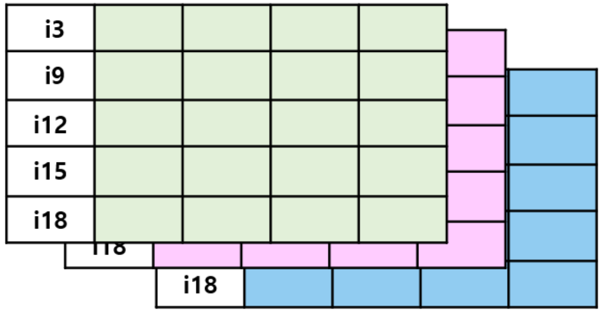
Deep Neural Network Approaches for SBRs

BERT (self-attention)

BERT4REC Example

2. Trm Layer := Transformer encoder

Trm layer is the same step as the transformer encoder's Multi-head attention process.



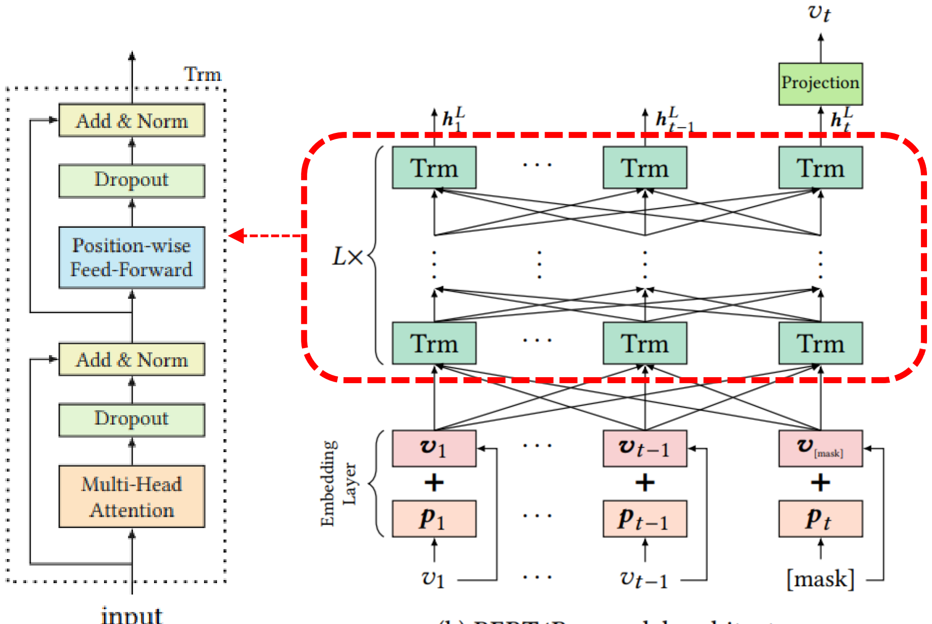
Q K V matrix (embdding*Wq, embdding*Wk, embdding *WV)

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{\top}}{\sqrt{d/h}}\right)V$$

$$\mathrm{PFFN}(H^{l}) = \left[\mathrm{FFN}(h_1^{l})^{\top}; \ldots; \mathrm{FFN}(h_t^{l})^{\top}\right]^{\top}$$

$$\mathrm{FFN}(x) = \mathrm{GELU}\left(xW^{(1)} + b^{(1)}\right)W^{(2)} + b^{(2)}$$

$$\mathrm{GELU}(x) = x\Phi(x)$$



(b) BERT4Rec model architecture.

Difference from Transformer: Activation Function → GELU

Sun, Fei, et al. "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer." *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019.

Deep Neural Network Approaches for SBRs

BERT (self-attention)

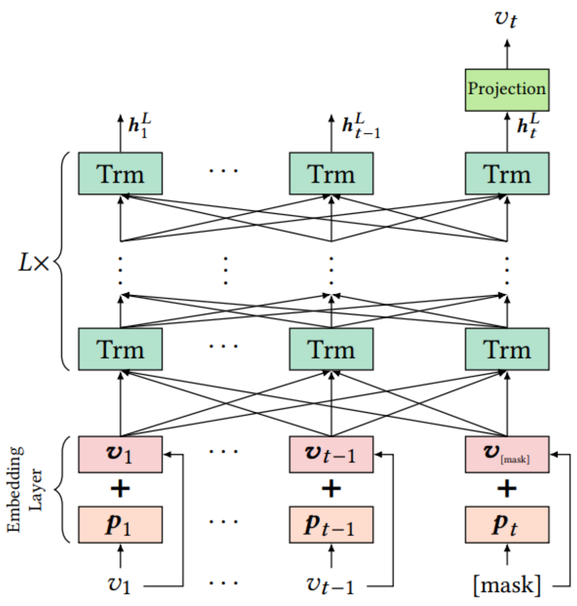
BERT4REC Example

3. Model Train

Masks are randomly placed on the item by a percentage (Hyperparameter).

Input: $[v_1, v_2, v_3, v_4, v_5] \xrightarrow{\text{randomly mask}} [v_1, [\texttt{mask}]_1, v_3, [\texttt{mask}]_2, v_5]$

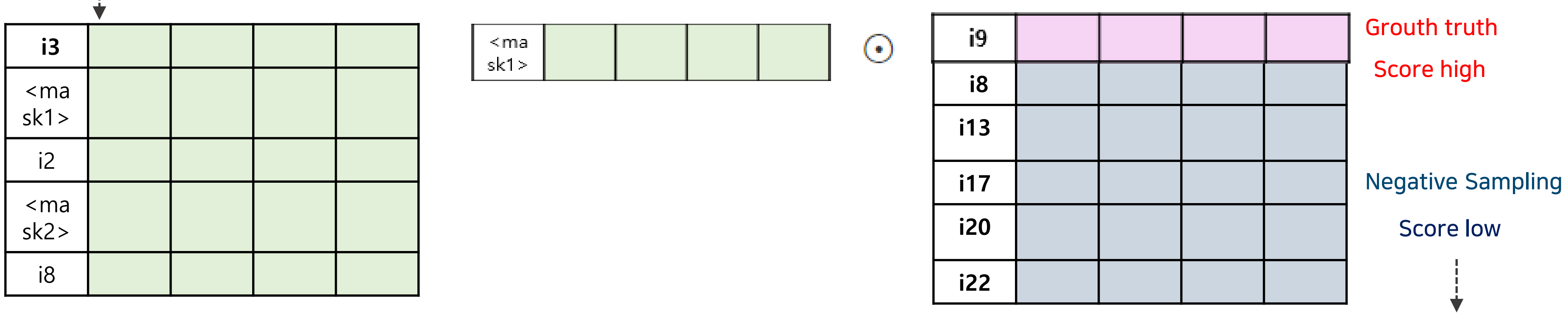
Labels: $[\texttt{mask}]_1 = v_2, \quad [\texttt{mask}]_2 = v_4$



(b) BERT4Rec model architecture.

{i3,i9,i12,i15,i18}

The process of learning the [mask] item is sequentially performed.(mask 01 → mask 02)



Random sampling among the top 100 popularity.

Loss function - masked target negative log likelihood

$$\mathcal{L} = \frac{1}{|S_u^m|} \Sigma_{v_m \in S_u^m} - log P(v_m = v_m^* | \dot{S}_u)$$

$\dot{S}_u$ User behavior history's masked version

v_m^* Masked item Vm true item

Sun, Fei, et al. "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer." *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019.

Deep Neural Network Approaches for SBRs

BERT (self-attention)

BERT4REC Example

Experiments

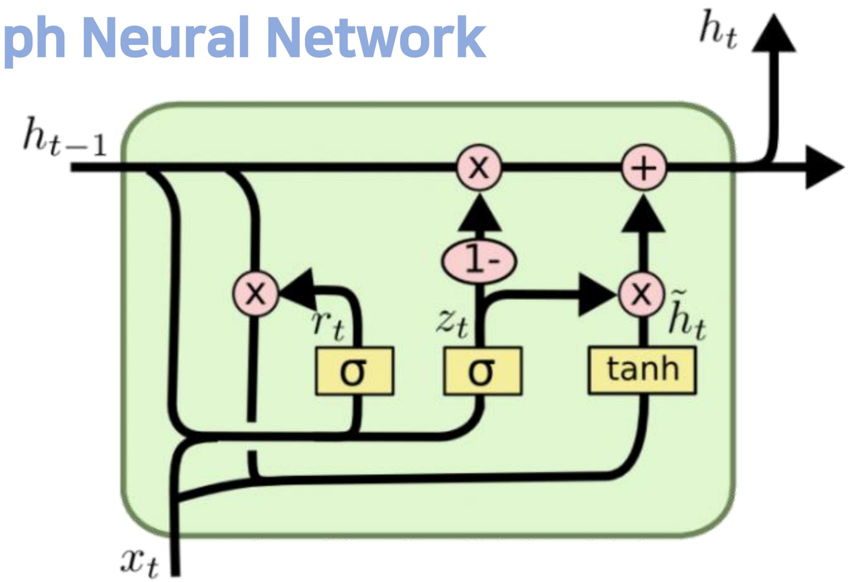
Datasets	Metric	POP	BPR-MF	NCF	FPMC	GRU4Rec	GRU4Rec^{+}	Caser	SASRec	BERT4Rec	Improv.
Beauty	HR@1	0.0077	0.0415	0.0407	0.0435	0.0402	0.0551	0.0475	0.0906	**0.0953**	5.19%
	HR@5	0.0392	0.1209	0.1305	0.1387	0.1315	0.1781	0.1625	0.1934	**0.2207**	14.12%
	HR@10	0.0762	0.1992	0.2142	0.2401	0.2343	0.2654	0.2590	0.2653	**0.3025**	14.02%
	NDCG@5	0.0230	0.0814	0.0855	0.0902	0.0812	0.1172	0.1050	0.1436	**0.1599**	11.35%
	NDCG@10	0.0349	0.1064	0.1124	0.1211	0.1074	0.1453	0.1360	0.1633	**0.1862**	14.02%
	MRR	0.0437	0.1006	0.1043	0.1056	0.1023	0.1299	0.1205	0.1536	**0.1701**	10.74%
Steam	HR@1	0.0159	0.0314	0.0246	0.0358	0.0574	0.0812	0.0495	0.0885	**0.0957**	8.14%
	HR@5	0.0805	0.1177	0.1203	0.1517	0.2171	0.2391	0.1766	0.2559	**0.2710**	5.90%
	HR@10	0.1389	0.1993	0.2169	0.2551	0.3313	0.3594	0.2870	0.3783	**0.4013**	6.08%
	NDCG@5	0.0477	0.0744	0.0717	0.0945	0.1370	0.1613	0.1131	0.1727	**0.1842**	6.66%
	NDCG@10	0.0665	0.1005	0.1026	0.1283	0.1802	0.2053	0.1484	0.2147	**0.2261**	5.31%
	MRR	0.0669	0.0942	0.0932	0.1139	0.1420	0.1757	0.1305	0.1874	**0.1949**	4.00%
ML-1m	HR@1	0.0141	0.0914	0.0397	0.1386	0.1583	0.2092	0.2194	0.2351	**0.2863**	21.78%
	HR@5	0.0715	0.2866	0.1932	0.4297	0.4673	0.5103	0.5353	0.5434	**0.5876**	8.13%
	HR@10	0.1358	0.4301	0.3477	0.5946	0.6207	0.6351	0.6692	0.6629	**0.6970**	4.15%
	NDCG@5	0.0416	0.1903	0.1146	0.2885	0.3196	0.3705	0.3832	0.3980	**0.4454**	11.91%
	NDCG@10	0.0621	0.2365	0.1640	0.3439	0.3627	0.4064	0.4268	0.4368	**0.4818**	10.32%
	MRR	0.0627	0.2009	0.1358	0.2891	0.3041	0.3462	0.3648	0.3790	**0.4254**	12.24%
ML-20m	HR@1	0.0221	0.0553	0.0231	0.1079	0.1459	0.2021	0.1232	0.2544	**0.3440**	35.22%
	HR@5	0.0805	0.2128	0.1358	0.3601	0.4657	0.5118	0.3804	0.5727	**0.6323**	10.41%
	HR@10	0.1378	0.3538	0.2922	0.5201	0.5844	0.6524	0.5427	0.7136	**0.7473**	4.72%
	NDCG@5	0.0511	0.1332	0.0771	0.2239	0.3090	0.3630	0.2538	0.4208	**0.4967**	18.04%
	NDCG@10	0.0695	0.1786	0.1271	0.2895	0.3637	0.4087	0.3062	0.4665	**0.5340**	14.47%
	MRR	0.0709	0.1503	0.1072	0.2273	0.2967	0.3476	0.2529	0.4026	**0.4785**	18.85%

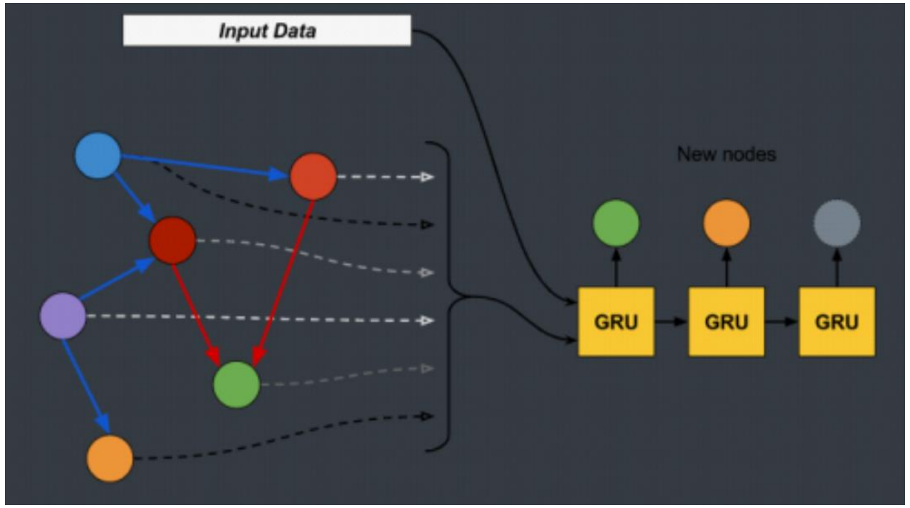
Sun, Fei, et al. "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer." *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019.

Deep Neural Network Approaches for SBRs

Graph Neural Networks

Gated Graph Neural Network





Reset gate $r^{(t)} = \sigma\left(W_r h^{(t-1)} + U_r x^{(t)}\right)$

Update gate $u^{(t)} = \sigma\left(W_u h^{(t-1)} + U_u x^{(t)}\right)$

Candidate gate $\tilde{h}^{(t)} = \tau\left(W h^{(t-1)} * r^{(t)} + U x^{(t)}\right)$

$$h^{(t)} = (1 - u^{(t)}) * h^{(t-1)} + u^{(t)} * \tilde{h}^{(t)}$$

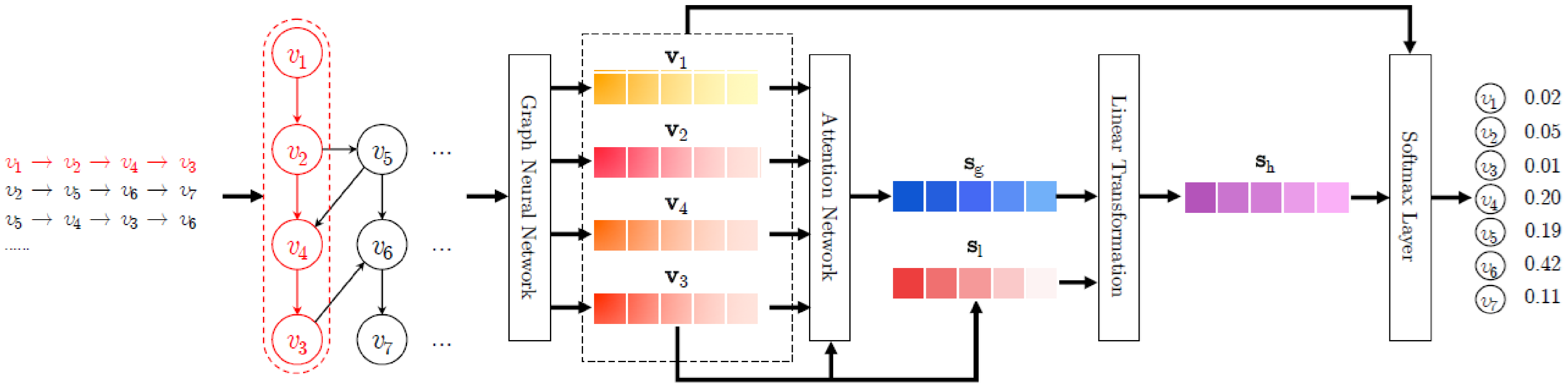
$$\begin{aligned}
\boxed{\mathbf{a}^{(t)}} &= \mathbf{A}\mathbf{h}^{(t-1)} + \mathbf{b} \\
\mathbf{r}_v^t &= \sigma\left(\mathbf{W}^r \mathbf{a}_v^{(t)} + \mathbf{U}^r \mathbf{h}_v^{(t-1)}\right) \\
\mathbf{z}_v^t &= \sigma\left(\mathbf{W}^z \mathbf{a}_v^{(t)} + \mathbf{U}^z \mathbf{h}_v^{(t-1)}\right) \\
\widetilde{\mathbf{h}_v^{(t)}} &= \tanh\left(\mathbf{W}\mathbf{a}_v^{(t)} + \mathbf{U}\left(\mathbf{r}_v^t \odot \mathbf{h}_v^{(t-1)}\right)\right) \\
\mathbf{h}_v^{(t)} &= (1 - \mathbf{z}_v^t) \odot \mathbf{h}_v^{(t-1)} + \mathbf{z}_v^t \odot \widetilde{\mathbf{h}_v^{(t)}}
\end{aligned}$$

Li, Yujia, et al. "Gated graph sequence neural networks." *arXiv preprint arXiv:1511.05493* (2015).

Deep Neural Network Approaches for SBRs

Graph Neural Networks

SR-GNN



$V = \{v_1, v_2, \ldots, v_m\}$ $s = [v_{s,1}, v_{s,2}, \ldots, v_{s,n}]$

[fig 11] SR-GNN Architecture

- SR-GNN aims to clarify the representation of the items by grasping many possible relationships between items through GNN.
- Directed graphs are constructed from past sequences (each sequence can be referred to as a subgraph)

- Wu, Shu, et al. "Session-based recommendation with graph neural networks." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019.

Deep Neural Network Approaches for SBRs

Graph Neural Networks

SR-GNN

1. Graph Neural Network Layer → Gated Graph Neural Network

list of the node vectors in session s

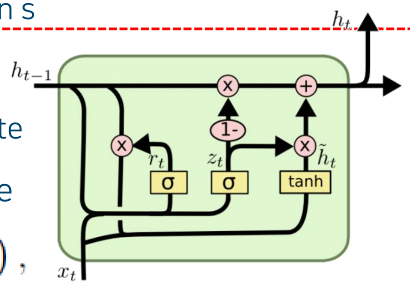
$$\mathbf{a}_{s,i}^t = \boxed{\mathbf{A}_{s,i}} \left[\mathbf{v}_1^{t-1}, \ldots, \mathbf{v}_n^{t-1}\right]^\top \mathbf{H} + \mathbf{b},$$

$$\mathbf{z}_{s,i}^t = \sigma\left(\mathbf{W}_z \mathbf{a}_{s,i}^t + \mathbf{U}_z \mathbf{v}_i^{t-1}\right), \quad \text{Update gate}$$

$$\mathbf{r}_{s,i}^t = \sigma\left(\mathbf{W}_r \mathbf{a}_{s,i}^t + \mathbf{U}_r \mathbf{v}_i^{t-1}\right), \quad \text{reset gate}$$

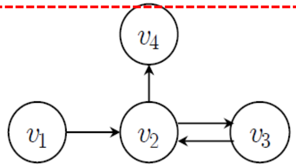
$$\widetilde{\mathbf{v}_i^t} = \tanh\left(\mathbf{W}_o \mathbf{a}_{s,i}^t + \mathbf{U}_o \left(\mathbf{r}_{s,i}^t \odot \mathbf{v}_i^{t-1}\right)\right),$$

$$\mathbf{v}_i^t = \left(1 - \mathbf{z}_{s,i}^t\right) \odot \mathbf{v}_i^{t-1} + \mathbf{z}_{s,i}^t \odot \widetilde{\mathbf{v}_i^t},$$



Idea

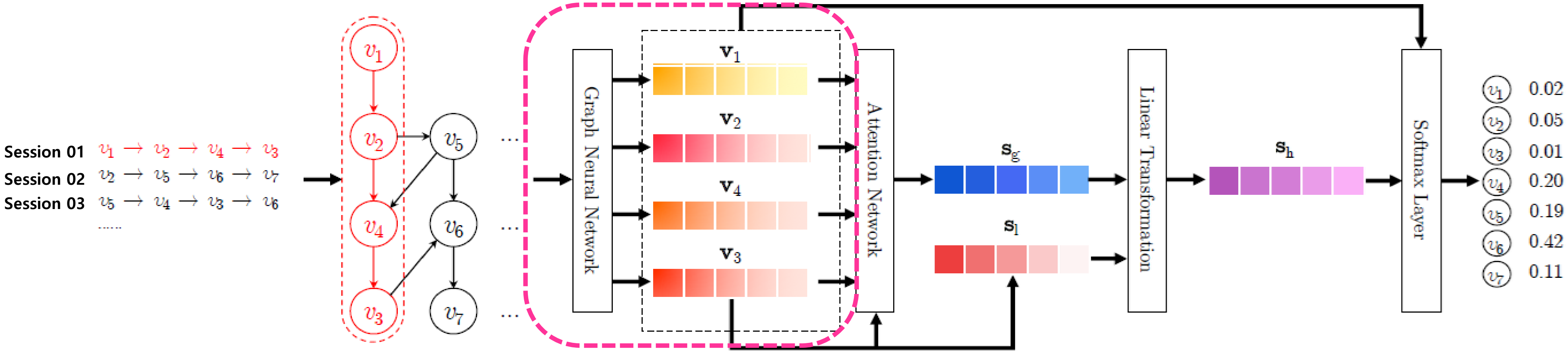
$$s = [v_1, v_2, v_3, v_2, v_4]$$



	Outgoing edges 1	2	3	4	Incoming edges 1	2	3	4
1	0	1	0	0	0	0	0	0
2	0	0	1/2	1/2	1/2	0	1/2	0
3	0	1	0	0	0	1	0	0
4	0	0	0	0	0	1	0	0

matrix $\mathbf{A}_s$

[Connected Matrix]



Wu, Shu, et al. "Session-based recommendation with graph neural networks." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019.

Deep Neural Network Approaches for SBRs

Graph Neural Networks

SR-GNN

2. Generating Session Embeddings: global session embedding(Sg) & local session embedding (Sl)

- **Global session embedding (Sg)**

  Aggregate nodes → Soft-Attention Mechanism

- **Local session embedding(Sl)**

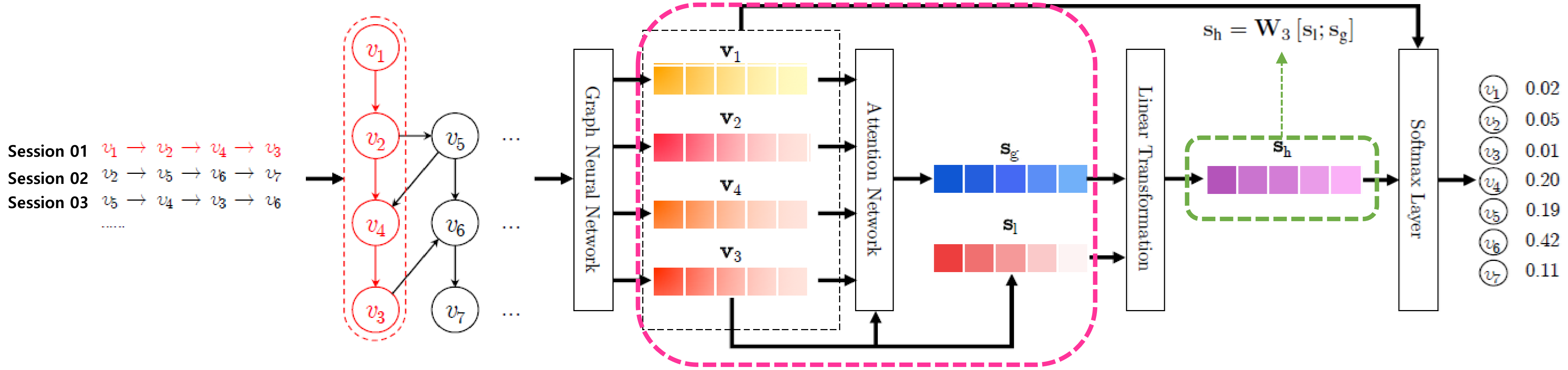
  The last item(vs) user clicked embedding vector $\mathbf{s}_l = \mathbf{v}_n$

* Soft-Attention Mechanism

$$\alpha_i = \mathbf{q}^\top \sigma(\mathbf{W}_1 \mathbf{v}_n + \mathbf{W}_2 \mathbf{v}_i + \mathbf{c}),$$

$$\mathbf{s}_g = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i,$$

3. sg,sl concatenate = Sh



Wu, Shu, et al. "Session-based recommendation with graph neural networks." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019.

Deep Neural Network Approaches for SBRs

Graph Neural Networks

SR-GNN

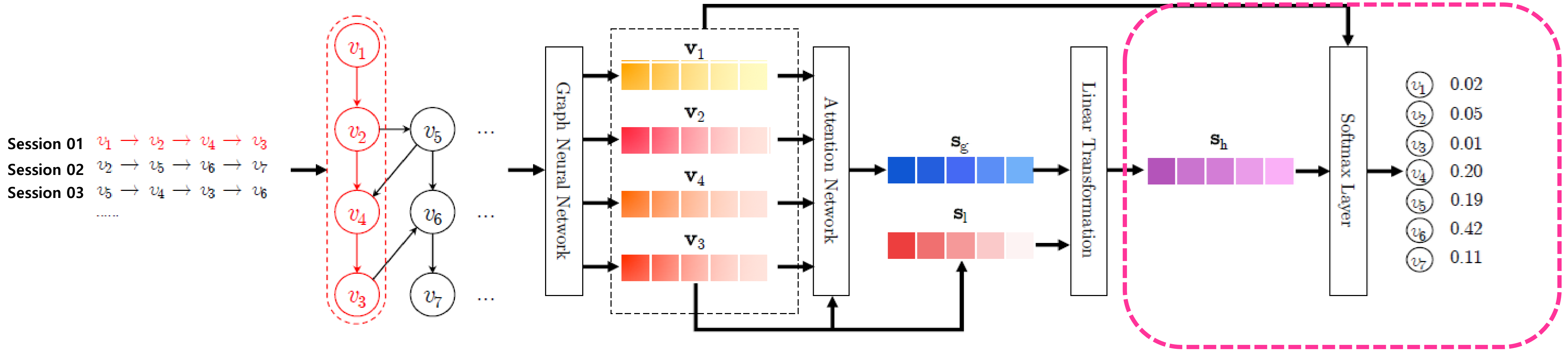
4. Prediction Next Item

$$\hat{z}_i = s_h^\top v_i \dashrightarrow \hat{y} = \mathrm{softmax}(\hat{z})$$

*** Loss Function: Cross-Entropy Loss**

$$\mathcal{L}(\hat{y}) = -\sum_{i=1}^{m} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

Y: denotes the one-hot encoding vector of the ground truth item



Wu, Shu, et al. "Session-based recommendation with graph neural networks." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019.

References

[1] Wang, Shoujin, et al. "A survey on session-based recommender systems." *ACM Computing Surveys (CSUR)* 54.7 (2021): 1-38.

[2] Ludewig, Malte, and Dietmar Jannach. "Evaluation of session-based recommendation algorithms." *User Modeling and User-Adapted Interaction* 28.4 (2018): 331-390.

[3] Rendle, Steffen, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing personalized markov chains for next-basket recommendation." *Proceedings of the 19th international conference on World wide web*. 2010.

[4] Hu, Liang, et al. "Diversifying Personalized Recommendation with User-session Context." IJCAI. 2017.

[5] Hidasi, Balázs, et al. "Session-based recommendations with recurrent neural networks." *arXiv preprint arXiv:1511.06939* (2015).

[6] Liu, Qiao, et al. "STAMP: short-term attention/memory priority model for session-based recommendation." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018.

[7] Li, Yujia, et al. "Gated graph sequence neural networks." *arXiv preprint arXiv:1511.05493* (2015).

[8] Sun, Fei, et al. "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer." *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019.

[9] Wu, Shu, et al. "Session-based recommendation with graph neural networks." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. No. 01. 2019.