GNN in Session-based Recommendation

GNN-RS Study Selected Paper Session

Seungwook Jin

22.02.08

Contents

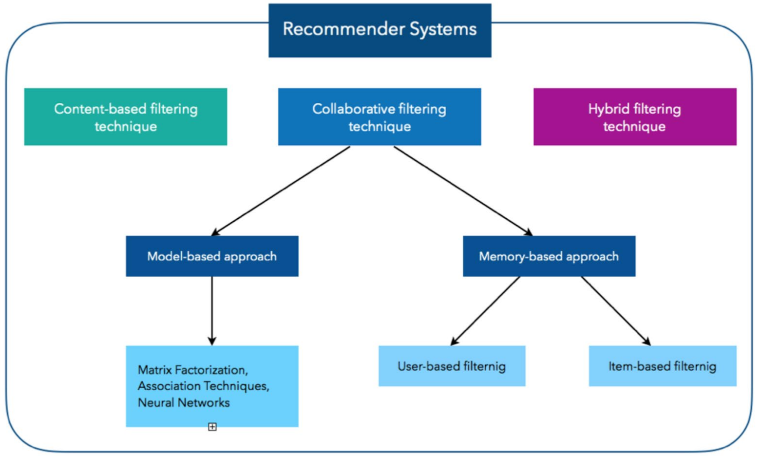
Introduction to SBRS

- Session-Based Recommender Systems (SBRS)

- SBRS vs. SRS (Sequential Recommender Systems)

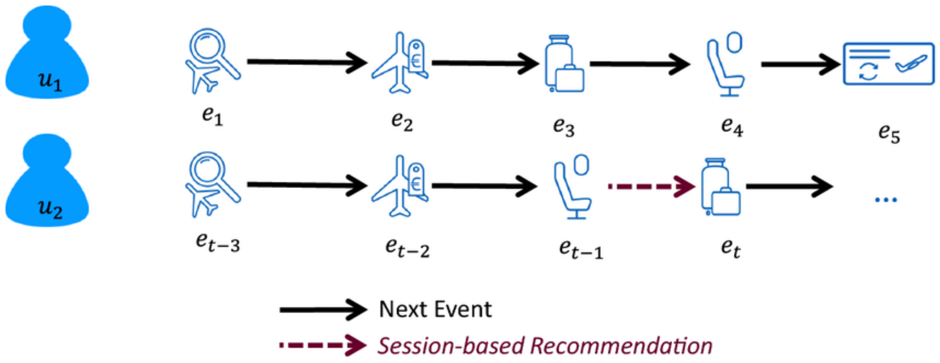
Session-Based Recommender Systems (SBRS)

Classical RS



- "History-based"
- Consider user's all historical interactions
- Learn long-term & static preference

Session-Based RS



- Setting : Mainly E-commerce click data
- Considering user's recent preference is needed
- In sessions, it's hard to specify user taste
  - usually short & anonymous

SBRS vs. SRS (Sequential Recommender Systems)

- Sequential RS : closely relevant but different from SBRS
    - Session : clear boundary exists
        - Can be ordered / unordered
        - Co-occurrence inside a session is also learnt
    - Sequence : no boundaries
        - No time intervals considered
        - Mainly sequential pattern is learnt

Table 1. A comparison between session data and sequence data

Data type		Boundary	Order	Time interval	Main relations embedded
Session data	Unordered session	Multiple	No	Non-identical	Co-occurrence-based dependencies
	Ordered session	Multiple	Yes	Non-identical	Co-occurrence-based dependencies and sequential dependencies
Sequence data		Single	Yes	Not included	Sequential dependencies

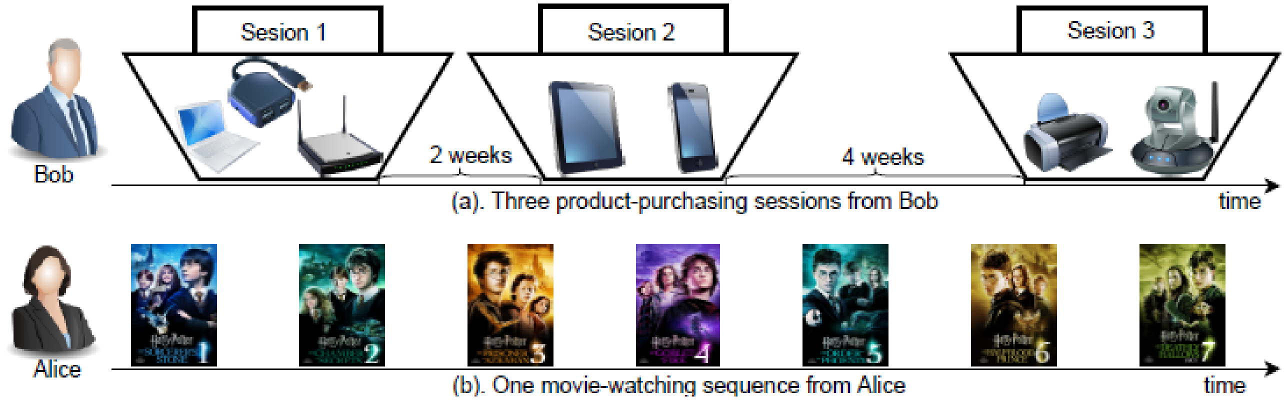


Fig. 1. Session data vs. sequence data

Wang et al. (2021) A Survey on Session-based Recommender Systems, *ACM Comput. Surv.*

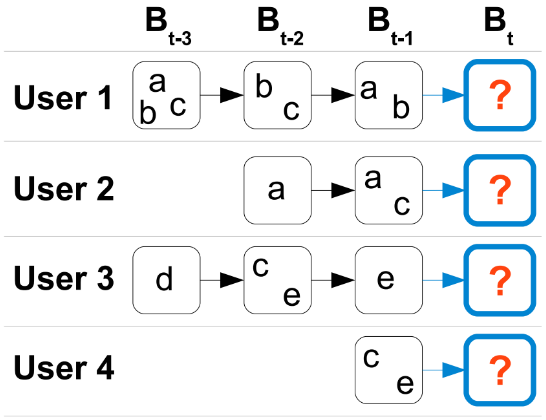
SBRS before GNN

- Conventional Recommendation Methods

- Deep-learning Based Methods

Conventional Recommendation Methods

- Matrix Factorization (Mnih and Salakhutdinov 2007; Koren, Bell, and Volinsky 2009; Koren and Bell 2011)
  - In session, user preference should be provided only by some positive clicks
- Item-based neighborhood methods (Sarwaret al. 2001)
  - Difficulty considering sequential order of items
- Markov chain Methods
  - FPMC (Rendle, Freudenthaler, and Schmidt-Thieme 2010)
  - Consider sequence, but combine past components independently

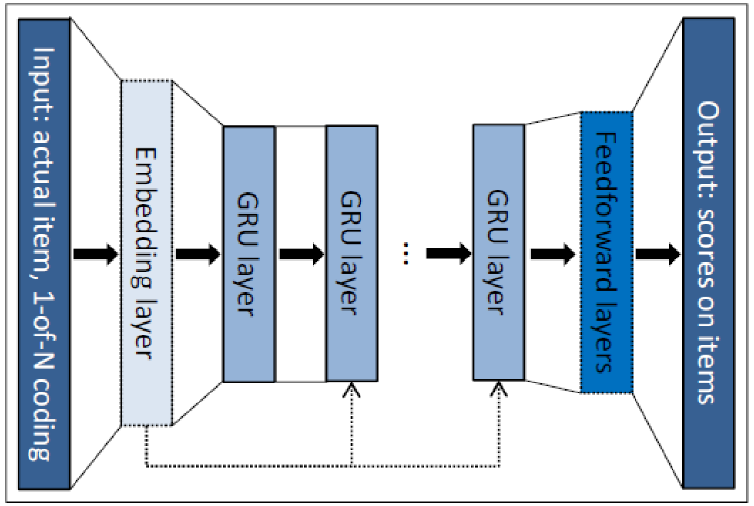


Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019.*

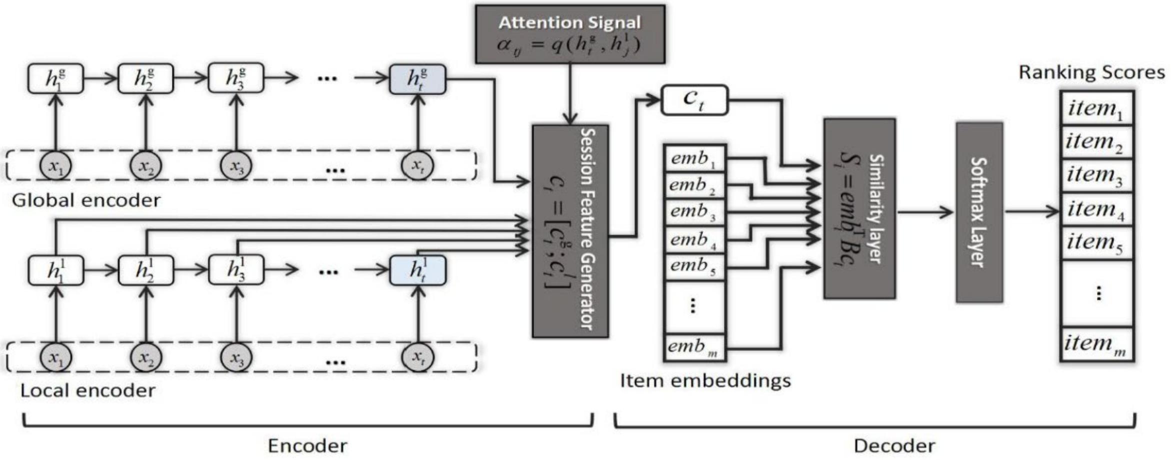
Deep-learning Based Methods

- Mostly, RNN based methods were successful
  - GRU4Rec : Gated Recurrent units (GRU) layer on session
  - NARM : Attention mechanism on RNN
- Limitations
  - Usually, the hidden vectors of RNN are treated as the user representations (hard in a short session)
  - Neglect the transitions among the contexts, i.e. complex transitions among distant items

GRU4Rec



NARM



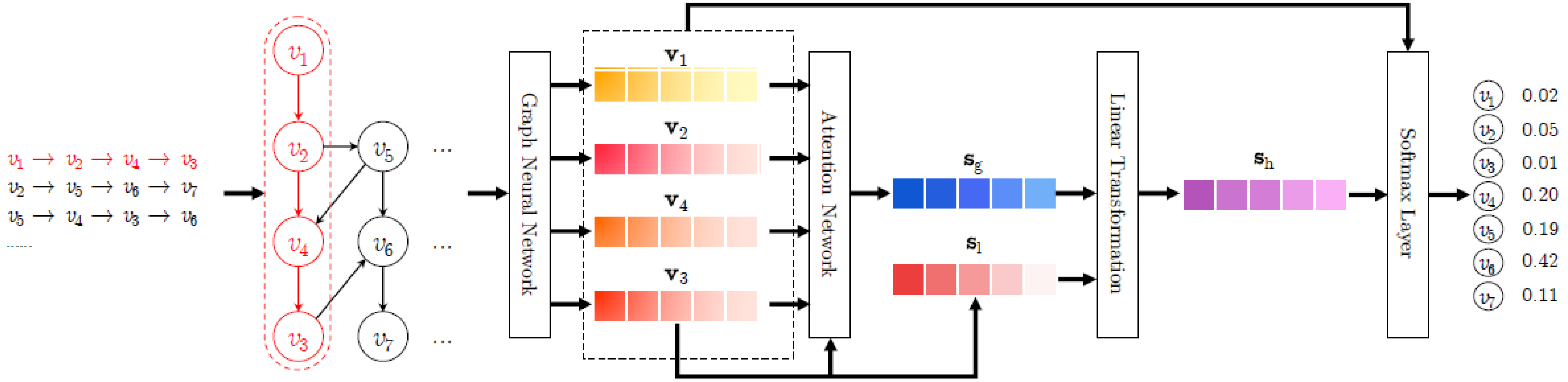
Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019*.

SR-GNN

- Overall Introduction
- Proposed Method
- Experimental Results

Overall Introduction - Workflow

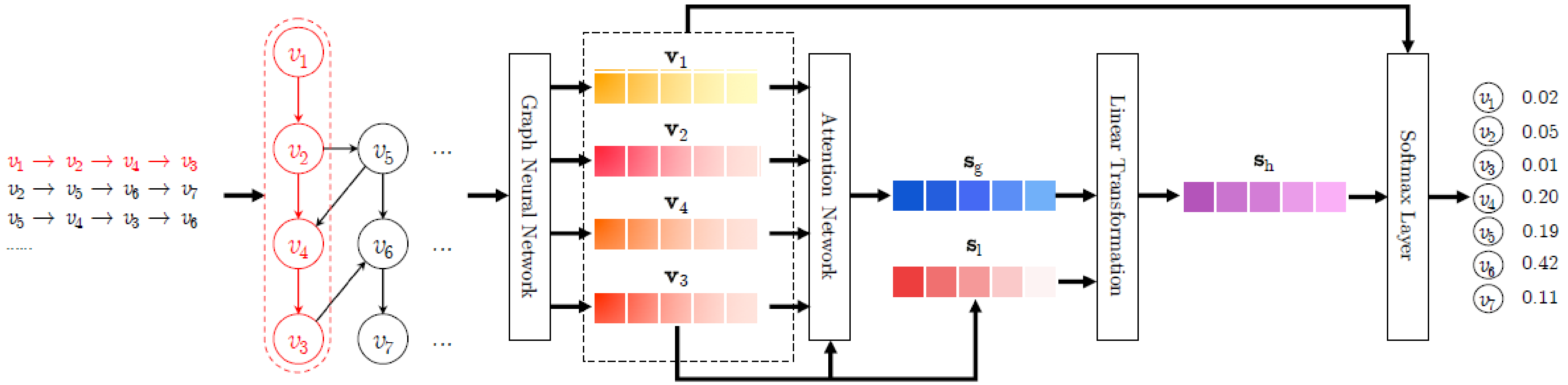
1. All session sequences are modeled as directed session graph
   - session sequence treated as a subgraph
2. Latent vectors for all nodes obtained through gated graph neural networks (GGNN)
   - proceed through information propagation
3. Represent each session as a global preference + current interest
   - both composed by the latent vectors of nodes
4. For each session, predict the probability of each item to be the next click.



Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019*.

Overall Introduction - Contributions

1. First to model separated session sequences into graph structured data
   - Before : only considered the whole sequence graph
2. Do not rely on User Representations, but use Session Embedding for prediction
   - Good performance on real-world datasets



Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019.*

Proposed Method

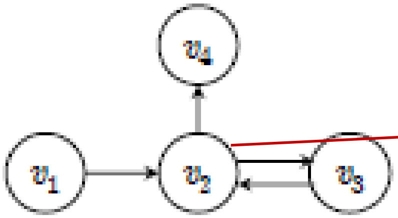
- Notations
  - Node $V = \{v_1, v_2, \ldots, v_m\}$: unique item set
  - Session $s = [v_{s,1}, v_{s,2}, \ldots, v_{s,n}]$: list of items
    - Predict next item $v_{s,n+1}$

기호	설명
$V = [v_1, v_2, \ldots, v_m]$	모든 세션에 속해 있는 모든 Unique한 Item의 집합
m	모든 Unique한 Item의 수
$s = [v_{s,1}, \ldots, v_{s,n}]$	특정 세션 s에 속해 있는 Item의 집합, 시간 순서에 의해 정렬됨
n	특정 세션 s에 속해 있는 Item의 수
$v_{s,n+1}$	세션 s 에서 다음 클릭의 대상자가 될 Item

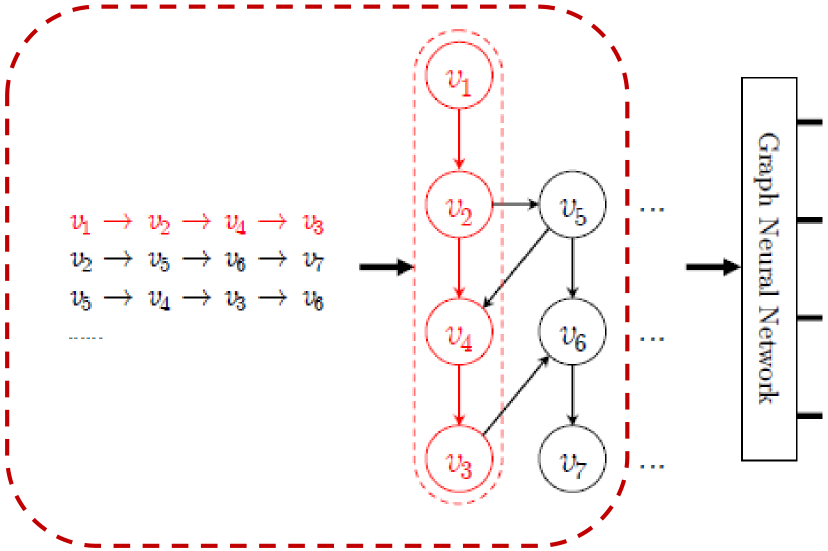
https://greeksharifa.github.io/machine_learning/2021/07/03/SRGNN/

Proposed Method

- Constructing Session Graphs
  - Directed Graph
  - Assign each edge with a normalized weight



	Outgoing edges				Incoming edges			
	1	2	3	4	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



Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019.*

Proposed Method

- Learning Item Embeddings on Session Graphs
  - Prior Work : Gated Graph Neural Networks (ICLR 2015)
    - Revised version of GNN, which uses GRU cell to update node vectors

$\mathbf{a}_v^{(t)} \in \mathbb{R}^{2D}$

Activation, instead of xt,

from Edges of both directions

Update Gate

Reset Gate

$$\mathbf{h}_v^{(1)} = [\boldsymbol{x}_v^\top, \mathbf{0}]^\top \quad (1)$$

$$\mathbf{a}_v^{(t)} = \mathbf{A}_{v:}^\top \left[\mathbf{h}_1^{(t-1)\top} \dots \mathbf{h}_{|\mathcal{V}|}^{(t-1)\top}\right]^\top + \mathbf{b} \quad (2)$$

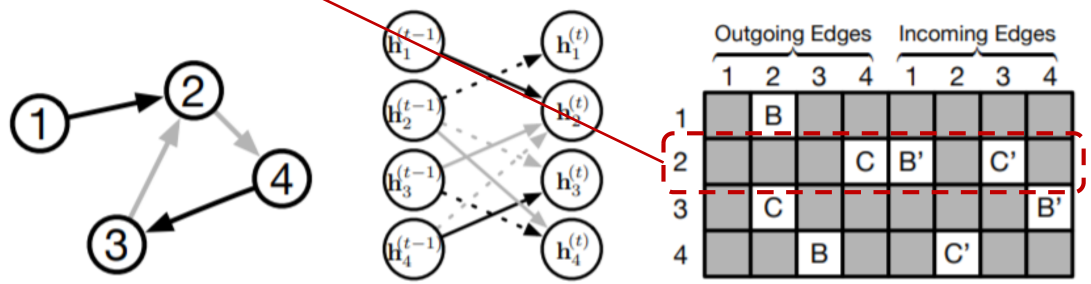
$$\mathbf{z}_v^t = \sigma\left(\mathbf{W}^z \mathbf{a}_v^{(t)} + \mathbf{U}^z \mathbf{h}_v^{(t-1)}\right) \quad (3)$$

$$\mathbf{r}_v^t = \sigma\left(\mathbf{W}^r \mathbf{a}_v^{(t)} + \mathbf{U}^r \mathbf{h}_v^{(t-1)}\right) \quad (4)$$

$$\widetilde{\mathbf{h}_v^{(t)}} = \tanh\left(\mathbf{W}\mathbf{a}_v^{(t)} + \mathbf{U}\left(\mathbf{r}_v^t \odot \mathbf{h}_v^{(t-1)}\right)\right) \quad (5)$$

$$\mathbf{h}_v^{(t)} = (1 - \mathbf{z}_v^t) \odot \mathbf{h}_v^{(t-1)} + \mathbf{z}_v^t \odot \widetilde{\mathbf{h}_v^{(t)}}. \quad (6)$$

Update Node Vector



(a) (b) (c) $\mathbf{A} = \left[\mathbf{A}^{(\text{out})}, \mathbf{A}^{(\text{in})}\right]$

Figure 1: (a) Example graph. Color denotes edge types. (b) Unrolled one timestep. (c) Parameter tying and sparsity in recurrent matrix. Letters denote edge types with B' corresponding to the reverse edge of type B. B and B' denote distinct parameters.

Li et al. (2015) Gated Graph Sequence Neural Networks, *ICLR 2015.*

Proposed Method

- Learning Item Embeddings on Session Graphs
  - SR-GNN : GGNN proceeds the same, with the session graph (subgraph) instead of Entire Graph
    - 1) Extract the latent vectors of neighborhoods
    - 2) Update and reset gate decide what information to be preserved and discarded
    - 3) Construct the candidate state by the previous state, the current state, and the reset gate
    - 4) Final state : combination of the previous hidden state and the candidate state,

$$\mathbf{h}_v^{(1)} = [\boldsymbol{x}_v^\top, \mathbf{0}]^\top \quad (1)$$

$$\mathbf{a}_v^{(t)} = \mathbf{A}_{v:}^\top \left[\mathbf{h}_1^{(t-1)\top} \ldots \mathbf{h}_{|\mathcal{V}|}^{(t-1)\top}\right]^\top + \mathbf{b} \quad (2)$$

$$\mathbf{z}_v^t = \sigma\left(\mathbf{W}^z \mathbf{a}_v^{(t)} + \mathbf{U}^z \mathbf{h}_v^{(t-1)}\right) \quad (3)$$

$$\mathbf{r}_v^t = \sigma\left(\mathbf{W}^r \mathbf{a}_v^{(t)} + \mathbf{U}^r \mathbf{h}_v^{(t-1)}\right) \quad (4)$$

$$\widetilde{\mathbf{h}_v^{(t)}} = \tanh\left(\mathbf{W}\mathbf{a}_v^{(t)} + \mathbf{U}\left(\mathbf{r}_v^t \odot \mathbf{h}_v^{(t-1)}\right)\right) \quad (5)$$

$$\mathbf{h}_v^{(t)} = (1 - \mathbf{z}_v^t) \odot \mathbf{h}_v^{(t-1)} + \mathbf{z}_v^t \odot \widetilde{\mathbf{h}_v^{(t)}}. \quad (6)$$

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

$$\mathbf{z}_{s,i}^t = \sigma\left(\mathbf{W}_z \mathbf{a}_{s,i}^t + \mathbf{U}_z \mathbf{v}_i^{t-1}\right), \quad (2)$$

$$\mathbf{r}_{s,i}^t = \sigma\left(\mathbf{W}_r \mathbf{a}_{s,i}^t + \mathbf{U}_r \mathbf{v}_i^{t-1}\right), \quad (3)$$

$$\widetilde{\mathbf{v}_i^t} = \tanh\left(\mathbf{W}_o \mathbf{a}_{s,i}^t + \mathbf{U}_o\left(\mathbf{r}_{s,i}^t \odot \mathbf{v}_i^{t-1}\right)\right), \quad (4)$$

$$\mathbf{v}_i^t = \left(1 - \mathbf{z}_{s,i}^t\right) \odot \mathbf{v}_i^{t-1} + \mathbf{z}_{s,i}^t \odot \widetilde{\mathbf{v}_i^t}, \quad (5)$$

Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019*.

Proposed Method

- A_s,i : Matrix Representation

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

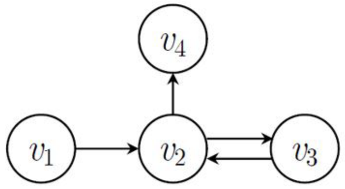
$$\mathbf{z}_{s,i}^t = \sigma\left(\mathbf{W}_z \mathbf{a}_{s,i}^t + \mathbf{U}_z \mathbf{v}_i^{t-1}\right), \tag{2}$$

$$\mathbf{r}_{s,i}^t = \sigma\left(\mathbf{W}_r \mathbf{a}_{s,i}^t + \mathbf{U}_r \mathbf{v}_i^{t-1}\right), \tag{3}$$

$$\widetilde{\mathbf{v}_i^t} = \tanh\left(\mathbf{W}_o \mathbf{a}_{s,i}^t + \mathbf{U}_o \left(\mathbf{r}_{s,i}^t \odot \mathbf{v}_i^{t-1}\right)\right), \tag{4}$$

$$\mathbf{v}_i^t = \left(1 - \mathbf{z}_{s,i}^t\right) \odot \mathbf{v}_i^{t-1} + \mathbf{z}_{s,i}^t \odot \widetilde{\mathbf{v}_i^t}, \tag{5}$$

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



	Outgoing edges				Incoming edges			
	1	2	3	4	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

Figure 2: A example of a session graph and the connection matrix $\mathbf{A}_s$

Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019.*

Proposed Method

- Vector Dimensions

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

$$\mathbf{z}_{s,i}^t = \sigma\left(\mathbf{W}_z \mathbf{a}_{s,i}^t + \mathbf{U}_z \mathbf{v}_i^{t-1}\right), \tag{2}$$

$$\mathbf{r}_{s,i}^t = \sigma\left(\mathbf{W}_r \mathbf{a}_{s,i}^t + \mathbf{U}_r \mathbf{v}_i^{t-1}\right), \tag{3}$$

$$\widetilde{\mathbf{v}_i^t} = \tanh\left(\mathbf{W}_o \mathbf{a}_{s,i}^t + \mathbf{U}_o \left(\mathbf{r}_{s,i}^t \odot \mathbf{v}_i^{t-1}\right)\right), \tag{4}$$

$$\mathbf{v}_i^t = \left(1 - \mathbf{z}_{s,i}^t\right) \odot \mathbf{v}_i^{t-1} + \mathbf{z}_{s,i}^t \odot \widetilde{\mathbf{v}_i^t}, \tag{5}$$

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

(1,2d) (1,2n) (n+n, d) (d,2d)

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

(d,1) (d,2d) (2d,1) (d,d) (d,1)

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

(d,1) (d,2d) (2d,1) (d,d) (d,1)

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

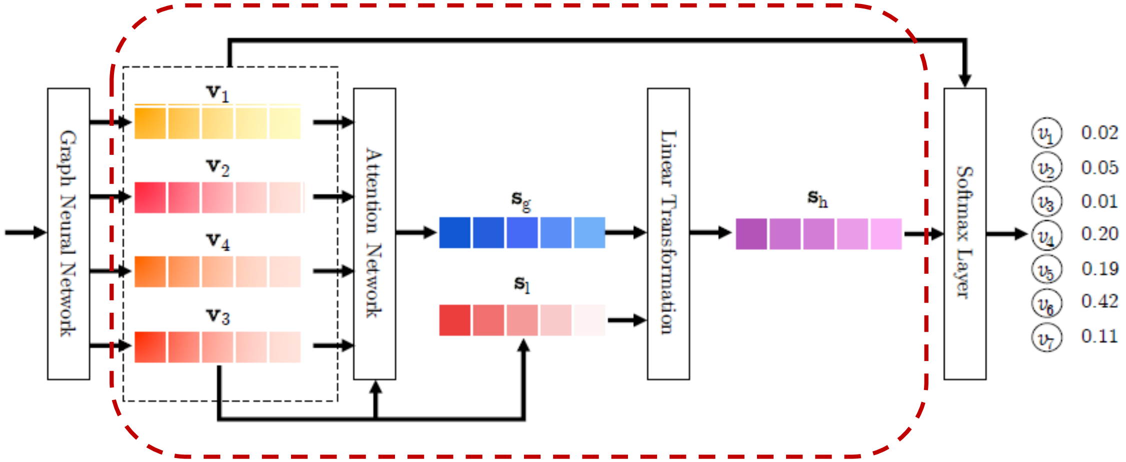
(d,1) (d,1) (d,1) (d,1) (d,1)

https://greeksharifa.github.io/machine_learning/2021/07/03/SRGNN/

Proposed Method

- Generating Session Embeddings
  - NO assumptions on user vector

  - Local Embedding (short-term preference)
    - Simply the last-clicked item $\mathbf{s}_l = \mathbf{v}_n$
  - Global Embedding (long-term preference)
    - Soft-Attention with the last clicked item
  - Final Session Embedding
    - Linear transformation over the concatenation of local & global



$$\mathbf{s}_l = \mathbf{v}_n$$

+

$$\alpha_i = \mathbf{q}^\top \sigma(\mathbf{W}_1 \mathbf{v}_n + \mathbf{W}_2 \mathbf{v}_i + \mathbf{c}),$$
$$\mathbf{s}_g = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i,$$

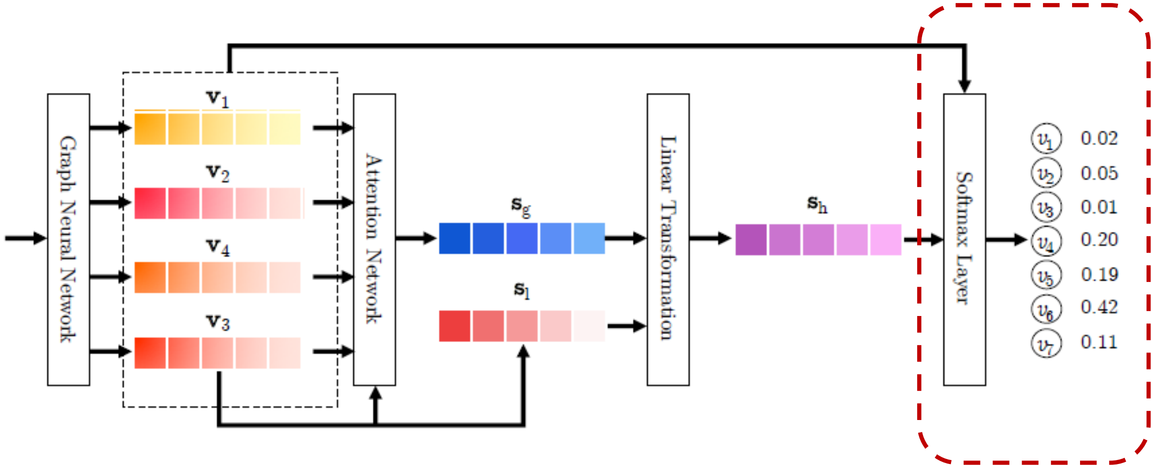
=

$$\mathbf{s}_h = \mathbf{W}_3 [\mathbf{s}_l ; \mathbf{s}_g]$$

Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019.*

Proposed Method

- Making Recommendation and Model Training
  - Compute score z_hat
    - Multiply item embedding v_i by session representation s_h
  - Softmax to get probability y_hat of next click
  - Loss : categorical cross-entropy
    - BPTT (Back-Propagation Through Time) Algorithm
    - Short epoch to prevent overfitting



$$\hat{\mathbf{z}}_i = \mathbf{s}_h^\top \mathbf{v}_i \quad \Rightarrow \quad \hat{\mathbf{y}} = \operatorname{softmax}(\hat{\mathbf{z}})$$

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

$$\underset{(d,1)}{\mathbf{s}_h} = \underset{(d,2d)}{\mathbf{W}_3} \underset{(2d,1)}{[\mathbf{s}_l; \mathbf{s}_g]}$$

$$\underset{(1,)}{\hat{\mathbf{z}}_i} = \underset{(1,d)}{\mathbf{s}_h^\top} \underset{(d,1)}{\mathbf{v}_i} \longrightarrow \underset{(m,1)}{\hat{z}} = \underset{(1,d)}{s_h^\top} \underset{(d,m)}{\mathbf{V}} \qquad \mathbf{V} = [v_1, \ldots, v_m]$$

행렬

$$\underset{(m,1)}{\hat{\mathbf{y}}} = \operatorname{softmax}\underset{(m,1)}{(\hat{\mathbf{z}})}$$

Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019.*

Experimental Results

- Dataset
  - Youchoose Dataset
    - RecSys 2015 Challenge
  - Dignetica
    - Also E-commerce
  - Discard length 1 session, items of freq < 5
    - To compare fairly

Statistics	*Yoochoose 1/64*	*Yoochoose 1/4*	*Diginetica*
# of clicks	557,248	8,326,407	982,961
# of training sessions	369,859	5,917,745	719,470
# of test sessions	55,898	55,898	60,858
# of items	16,766	29,618	43,097
Average length	6.16	5.71	5.12

- Training
  - BPTT (Back-Propagation Through Time) Algorithm
  - Short epoch to prevent overfitting

$$s = [v_{s,1}, v_{s,2}, \ldots, v_{s,n}] \Rightarrow ([v_{s,1}], v_{s,2}), ([v_{s,1}, v_{s,2}], v_{s,3}), \ldots, ([v_{s,1}, v_{s,2}, \ldots, v_{s,n-1}], v_{s,n})$$

Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019*.

Experimental Results

- Baseline Algorithms
  - Pop and S-Pop
  - Item-KNN (Sarwar et al. 2001)
  - BPR-MF (Rendle et al. 2009)
  - FPMC (Rendle, Freudenthaler, and Schmidt-Thieme 2010)
  - GRU4Rec (Hidasi et al. 2016a)
  - NARM (Li et al. 2017a)
  - STAMP (Liu et al. 2018)

- Evaluation Metrics
  - Precision @ 20
  - MRR @ 20 (Mean Reciprocal Rank)

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\mathrm{rank}_i}$$

Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019*.

Experimental Results

- Results
  - Item-KNN : Quite good
    - considering its simplicity
  - GRU4Rec : Good
    - outperform conventional methods
  - NARM and STAMP : Better
    - NARM – RNN + Attention
    - STAMP – Current Interest + General Interest
  - SR-GNN : Best
    - Consider item transitions
    - Soft-attention to capture significant interactions

Method	Yoochoose 1/64		Yoochoose 1/4		Diginetica	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
POP	6.71	1.65	1.33	0.30	0.89	0.20
S-POP	30.44	18.35	27.08	17.75	21.06	13.68
Item-KNN	51.60	21.81	52.31	21.70	35.75	11.57
BPR-MF	31.31	12.08	3.40	1.57	5.24	1.98
FPMC	45.62	15.01	–	–	26.53	6.95
GRU4REC	60.64	22.89	59.53	22.60	29.45	8.33
NARM	68.32	28.63	69.73	29.23	49.70	16.17
STAMP	68.74	29.67	70.44	30.00	45.64	14.32
SR-GNN	**70.57**	**30.94**	**71.36**	**31.89**	**50.73**	**17.59**

Wu et al. (2019) Session-Based Recommendation with Graph Neural Networks, *AAAI 2019*.

GNN in SBRS after SR-GNN

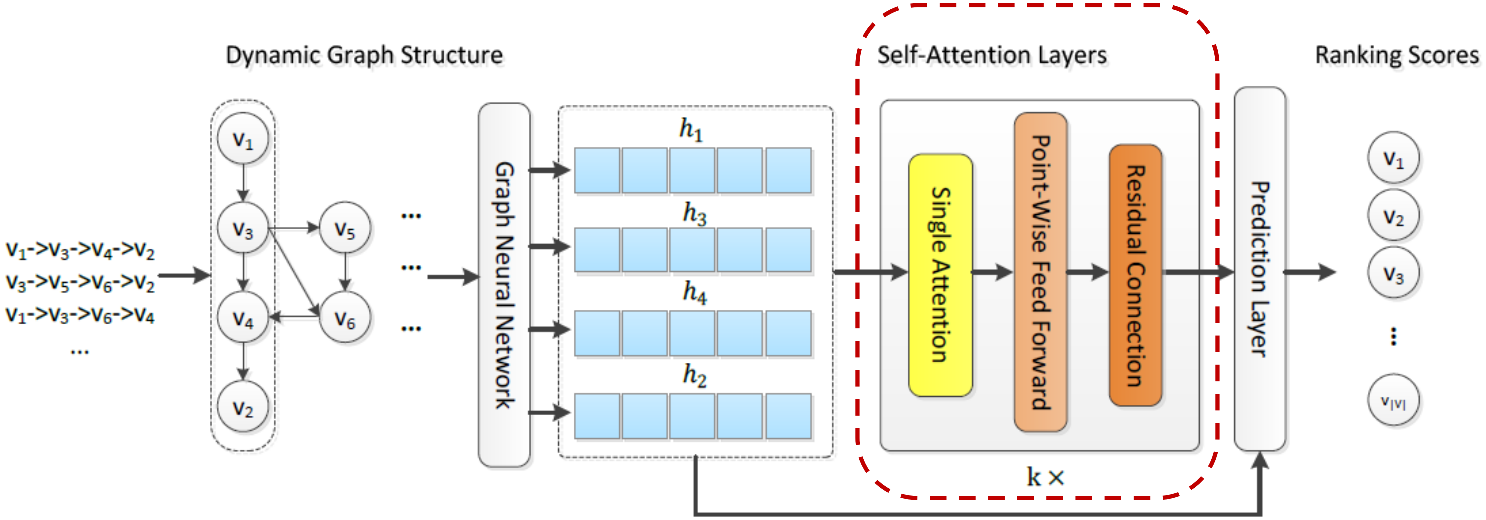
- GC-SAN

- Illustration of GNN models after SR-GNN

- Details of GNN models after SR-GNN

GC-SAN (Graph Contextualized Self-Attention Network)

- Overall Structure Similar, but added Multi-layer Self-Attention Networks



$$\mathbf{F} = softmax\left(\frac{(\mathbf{HW}^Q)(\mathbf{HW}^K)^T}{\sqrt{d}}\right)(\mathbf{HW}^V)$$

$$\mathbf{E} = ReLU(\mathbf{FW}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 + \mathbf{F}$$

$$\mathbf{E}^{(k)} = SAN(\mathbf{E}^{(k-1)})$$

Zhao et al. (2019Graph Contextualized Self-Attention Network for Session-based Recommendation, IJCAI 2019

GC-SAN (Graph Contextualized Self-Attention Network)

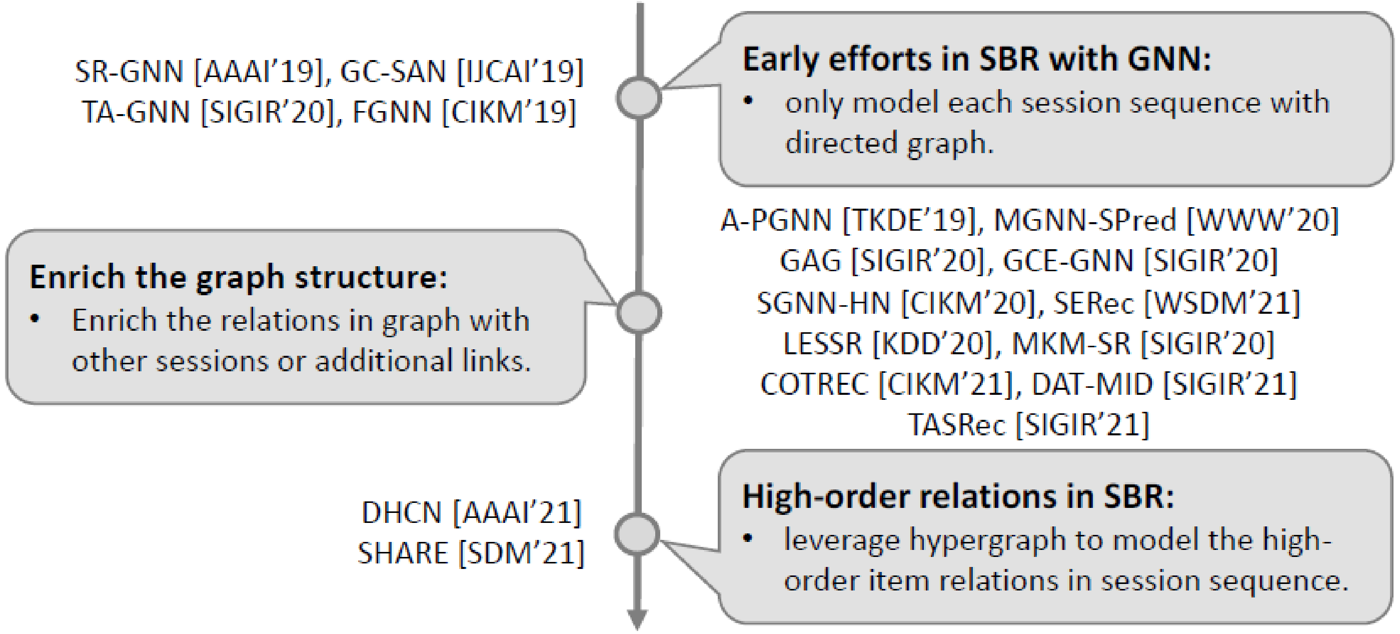
- Performance Results

Datasets	Diginetica						Retailrocket					
Measures	HR@5	HR@10	MRR@5	MRR@10	NDCG@5	NDCG@10	HR@5	HR@10	MRR@5	MRR@10	NDCG@5	NDCG@10
Pop	0.0036	0.0077	0.0019	0.0025	0.0023	0.0037	0.0133	0.0208	0.0066	0.0076	0.0082	0.0107
BPR-MF	0.1060	0.1292	0.0789	0.0842	0.0586	0.0672	0.2106	0.2719	0.1356	0.1407	0.1138	0.1322
IKNN	0.1407	0.2083	0.0776	0.0867	0.0693	0.0902	0.1709	0.2248	0.0972	0.1043	0.0855	0.1020
FPMC	0.1855	0.2309	0.0875	0.0986	0.0811	0.1037	0.1732	0.2319	0.1013	0.1152	0.0901	0.1095
GRU4Rec	0.2577	0.3657	0.1434	0.1577	0.1276	0.1607	0.2196	0.2869	0.1286	0.1489	0.1076	0.1323
STAMP	0.3998	0.5014	0.2357	0.2469	0.2039	0.2394	0.3287	0.3972	0.2241	0.2334	0.1758	0.1970
SR-GNN	0.4082	0.5269	0.2439	0.2599	0.2078	0.2443	0.3502	0.4268	0.2422	0.2525	0.1885	0.2121
GC-SAN	**0.4280**	**0.5351**	**0.2694**	**0.2838**	**0.2223**	**0.2552**	**0.3644**	**0.4380**	**0.2506**	**0.2604**	**0.1956**	**0.2181**
Improv.	4.84%	1.56%	10.46%	9.20%	6.98%	4.44%	4.07%	2.62%	3.49%	3.15%	3.79%	2.85%

Zhao et al. (2019Graph Contextualized Self-Attention Network for Session-based Recommendation, IJCAI 2019

Illustration of GNN models after SR-GNN

- Early Efforts
  - Directed graph only
- Later Developments
  - Use information cross sessions and additional edge links
- Capture High-order Relations
  - Hypergraph



Gao et al. (2021) Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions, *arXiv preprint.*

Details of GNN models after SR-GNN

- Summary Table

Model	Graph	GNN	Enrich Graph Structure
SR-GNN [169]	directed graph	gated GNN	-
GC-SAN [181]	directed graph	gated GNN	-
TA-GNN [190]	directed graph	gated GNN	-
FGNN [122]	directed graph	GAT	-
A-PGNN [170]	directed graph	gated GNN	cross sessions
MGNN-SPred [150]	directed & multi-relational item graph	GraphSAGE	cross sessions
GAG [123]	directed graph	GCN	cross sessions
SGNN-HN [116]	star graph	gated GNN	additional edges
GCE-GNN [157]	directed graph + global graph	GAT	cross sessions
DGTN [208]	directed graph	GCN	cross sessions
DHCN [176]	hypergraph + line graph	HyperGCN	cross sessions
SHARE [148]	session hypergraph	HyperGAT	additional edges
SERec [24]	KG + directed graph	GAT + gated GNN	cross sessions
LESSR [23]	directed graph	GAT	additional edges
CAGE [128]	KG + article-level graph	GCN	cross sessions
MKM-SR [109]	KG + directed graph	gated GNN	cross sessions
COTREC [175]	item graph + line graph	GCN	cross sessions
DAT-MDI [15]	directed graph + global graph	GAT	cross sessions
TASRec [211]	dynamic graph	GCN	cross sessions

Gao et al. (2021) Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions, *arXiv preprint.*

Thoughts

- 세션 간, 세션 내 Item 간의 Time Interval을 고려한 후속 연구가 있나?
    - 보통 Attention을 통해 Noise 및 Item별 관심도가 잡힐 것 같긴 하지만
    - 세션 내 아이템당 체류 시간을 Edge Weight로 줘서 학습에 넣는 방향을 고려해보는 것도 좋을 것 같다

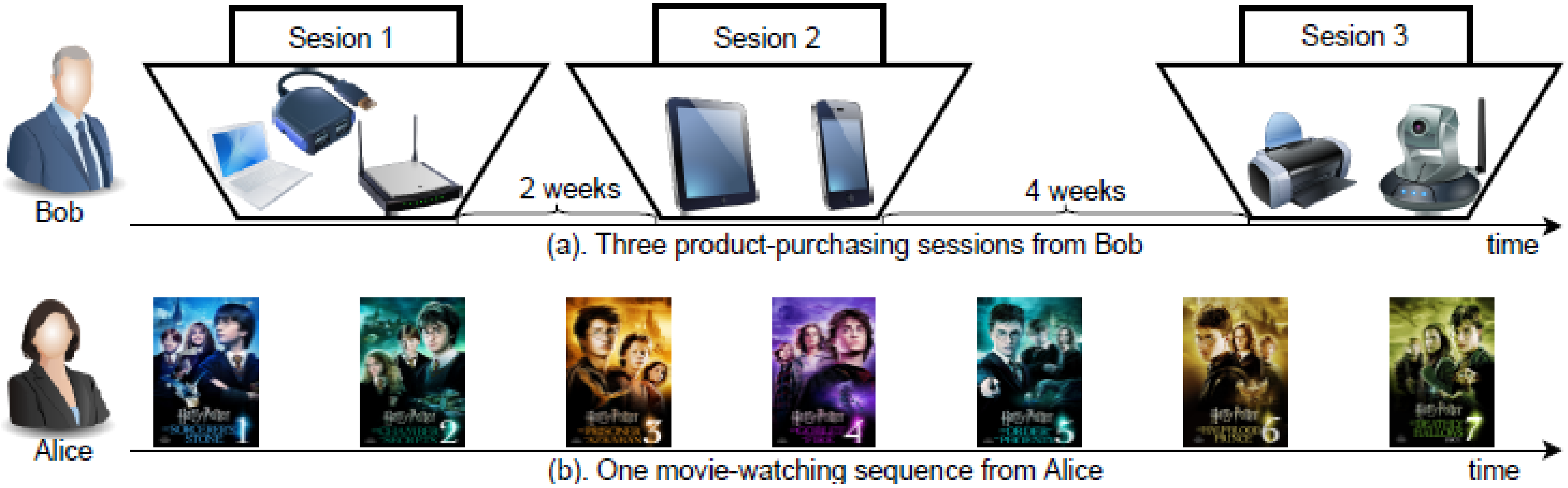


Fig. 1. Session data vs. sequence data

Thank You