SIGIR 2020
XI'AN·CHINA

Incorporating User Micro-behaviors and Item Knowledge into Multi-task Learning for Session-based Recommendation

Wenjing Meng[1], **Deqing Yang**[1*], Yanghua Xiao[2]

[1]*School of Data Science*, [2]*School of Computer Science*, ***Fudan University***, *China*

**crresponding author*

2020-7

Outline

- Introduction
- Methodology
- Evaluation
- Conclusion

Motivations

- Most of previous session-based models only model a session from a macro-level, without taking different operations of users into account.
  - Even though a user interacts with the same item in a session, different operations committed on this item reflect the user's different intentions within this session and different preferences on this item.

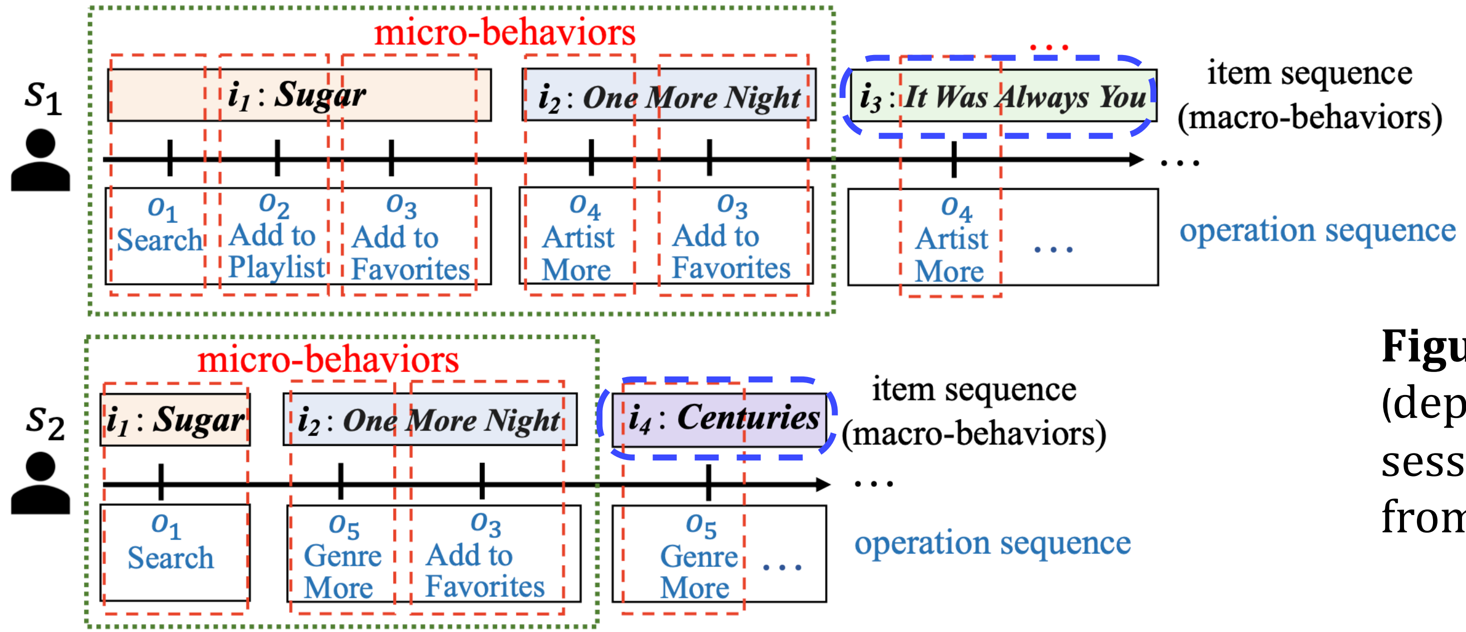
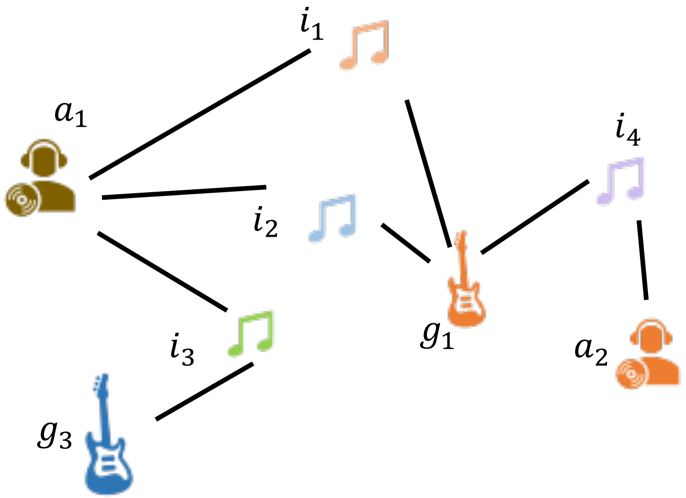


Figure 1: A toy example of user micro-behaviors (depicted by red dashed rectangles) in two sessions (depicted by green dashed rectangles) from a music site.

Motivations

- Insufficient utilization of item knowledge
  - Item knowledge can be used to alleviate the data sparsity problem.
  - Item transitions in terms of micro-behavior can also be indicated by item knowledge.



Transition: $i_2 \xrightarrow{\text{artist more}} i_3$, is indicated by the knowledge triplets <i_1, song-artist, Maroon 5> and <i_2, song-artist, Maroon 5>.

Basic Ideas

- We propose a novel SR model **MKM-SR** which incorporates user **M**icro-behaviors and item **K**nowledge into **M**ulti-task learning for **S**ession-based **R**ecommendation.
- In MKM-SR, a user's sequential pattern in a session is modeled on *micro-behavior level* rather than item-level. Each micro-behavior is actually a combination of an item and its corresponding operation.
  - Item embeddings are learned by Gated Graph Neural Network (GGNN).
  - Operation embeddings are learned by Gated Recurrent Unit (GRU).
- TransH is used to learn the many-to-many/one relations in the item knowledge graph, and regard it as an auxiliary task in the multi-task learning (MTL) paradigm in which the major task is to predict the next interacted item.

Basic Ideas

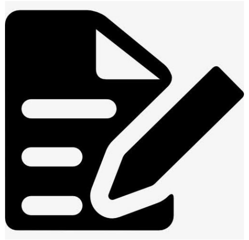
- Item sequence

$$\overrightarrow{\overleftarrow{i_1 \quad i_2 \quad i_3 \quad i_1}}$$

Bidirectional transition pattern

GRU✕ **GGNN**√

- Operation sequence



Browse product review

Add to cart

Place order

Unidirectional transition pattern

GRU√

Contributions

1, We incorporate user micro-behaviors into session modeling to capture the transition pattern between the successive items in a session on a fine-grained level.

2, We incorporate item knowledge through an MTL paradigm which takes knowledge embedding learning as an auxiliary (sub) task of SR, and further validate an optimal training strategy for MTL through extensive comparisons.

3, We provide deep insights on the rationales of model mechanisms, extensive evaluations to justify our model's superiority.

Outline

- Introduction
- Methodology
- Evaluation
- Conclusion

Model Framework

Encoding session information

An item sequence and an operation sequence are extracted from a given session simultaneously, then fed into GGNN and GRU to learn *item embeddings* and *operation embeddings*. These two types of embeddings assemble a sequence of *micro-behavior embeddings.*

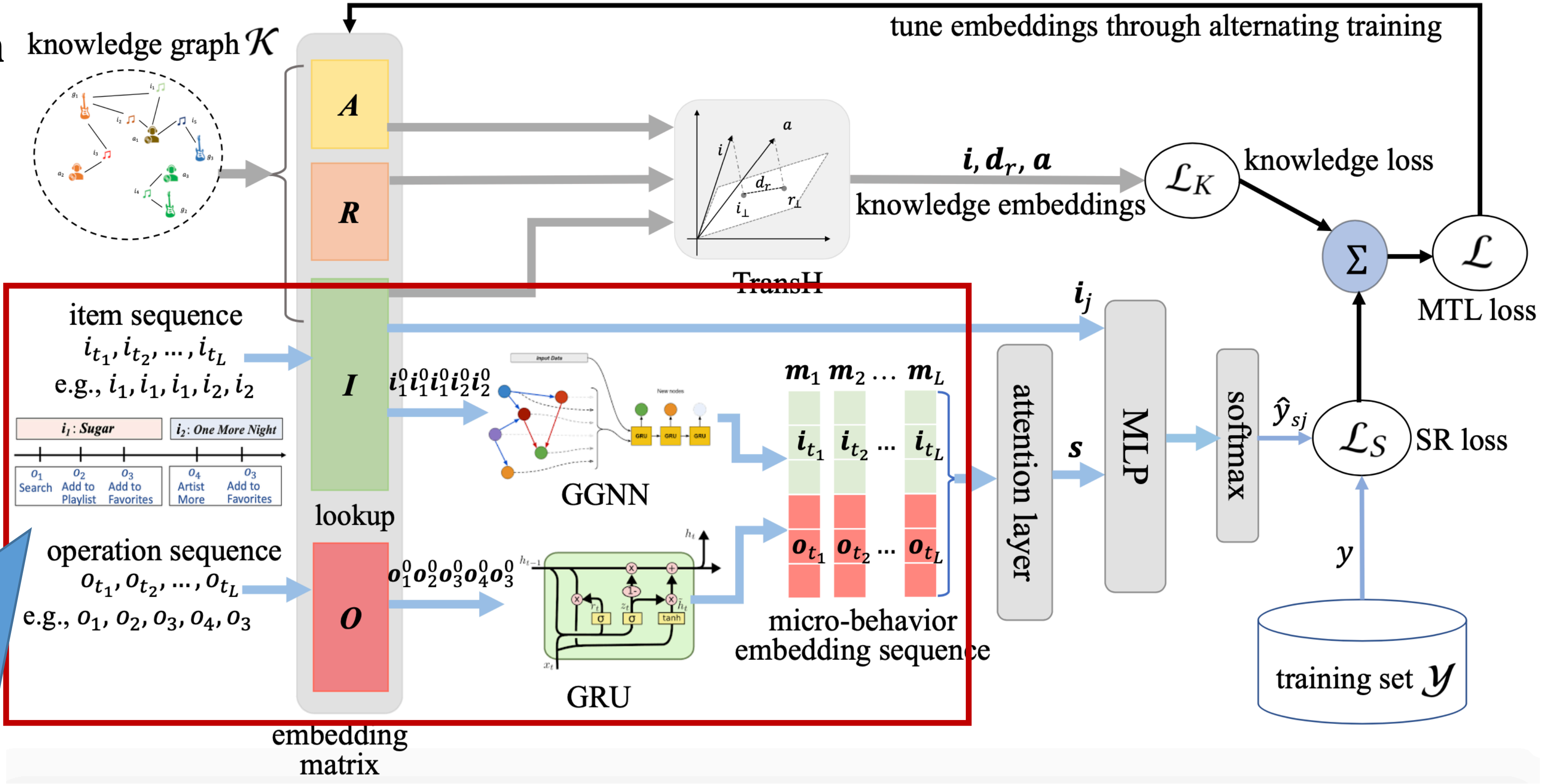


Figure 2: The overall framework of our proposed MKM-SR. The arrows in the figure indicate data flows.

Model Framework

Encoding session information

Micro-behavior embedding sequence is used to generate the session representation $\boldsymbol{s}$. The final score $\hat{y}_{sj}$ is computed by an MLP followed by softmax operation on $\boldsymbol{s}$ and item j's representation $\boldsymbol{i}_j$.

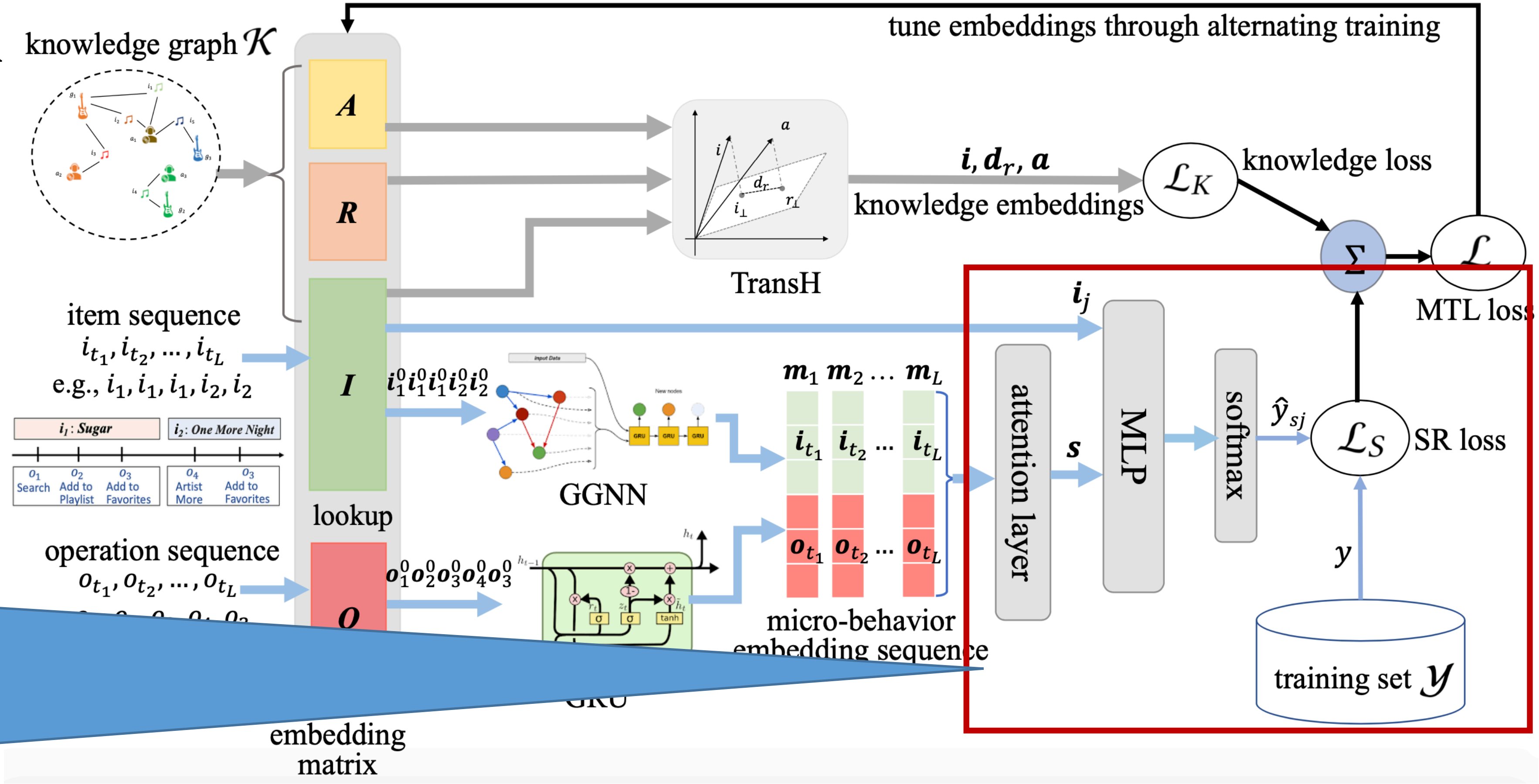


Figure 2: The overall framework of our proposed MKM-SR. The arrows in the figure indicate data flows.

Model Framework

Learning knowledge embeddings

Learn knowledge embeddings by TransH. Knowledge loss $\mathcal{L}_K$ and session prediction loss $\mathcal{L}_S$ are incorporated into a multi-task learning loss function $\mathcal{L}$ to learn better item embeddings.

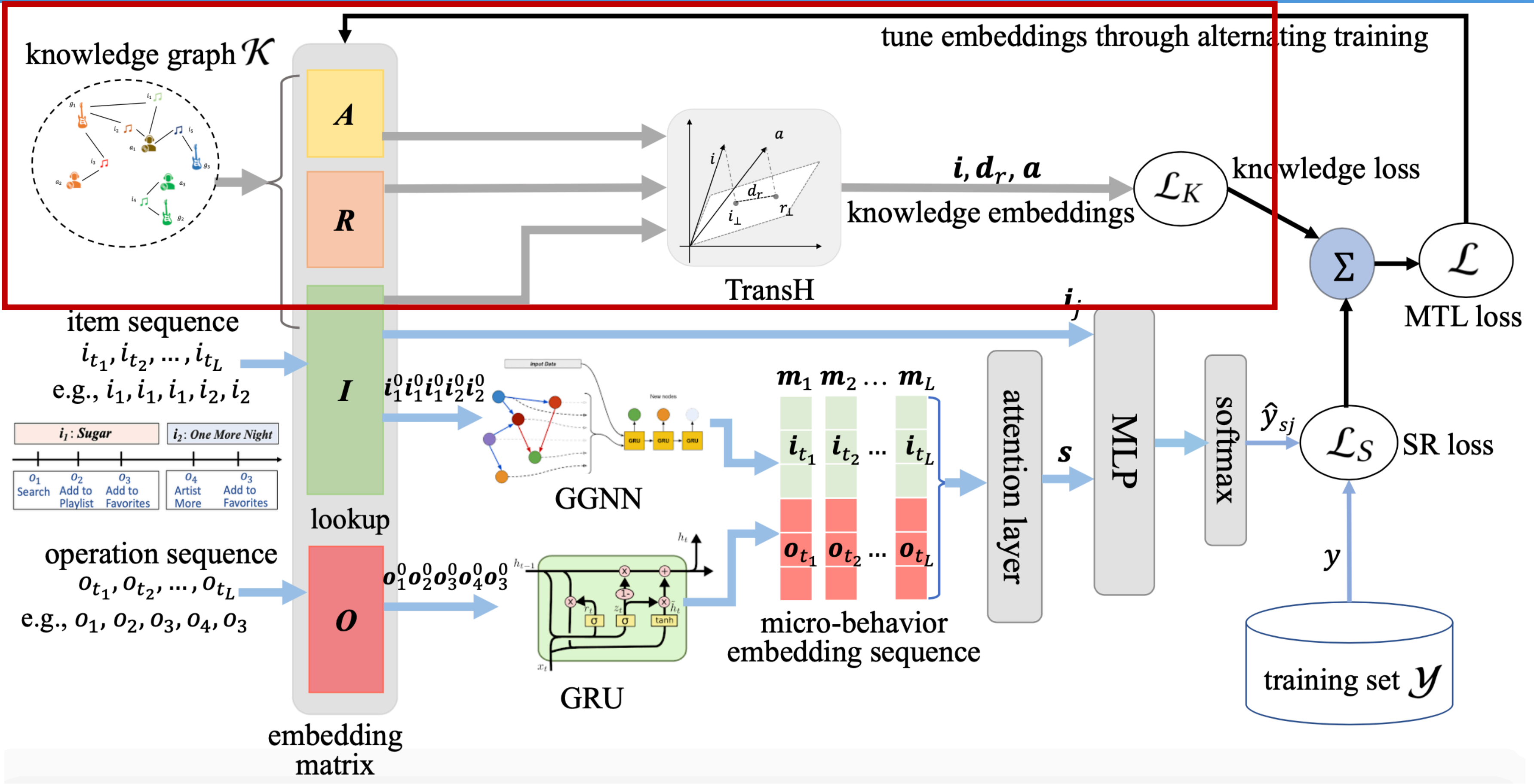


Figure 2: The overall framework of our proposed MKM-SR. The arrows in the figure indicate data flows.

Encoding Session Information

- Learning item embeddings by GGNN

$$\mathbf{a}_v^h = (\mathbf{A}_{v:}^{+} + \mathbf{A}_{v:}^{-})[\mathbf{h}_1^{h-1}, \mathbf{h}_2^{h-1}, \ldots, \mathbf{h}_{|\mathcal{V}|}^{h-1}]^\top + \mathbf{b}$$
$$\mathbf{z}_v^h = \sigma(\mathbf{W}_{az}\mathbf{a}_v^h + \mathbf{W}_{hz}\mathbf{h}_v^{h-1})$$
$$\mathbf{r}_v^h = \sigma(\mathbf{W}_{ar}\mathbf{a}_v^h + \mathbf{W}_{hr}\mathbf{h}_v^{h-1})$$
$$\mathbf{c}_v^h = tanh(\mathbf{W}_{ac}\mathbf{a}_v^h + \mathbf{W}_{hc}(\mathbf{r}_v^h \odot \mathbf{h}_v^{h-1}))$$
$$\mathbf{h}_v^h = (1 - \mathbf{z}_v^h) \odot \mathbf{h}_v^{h-1} + \mathbf{z}_v^h \odot \mathbf{c}_v^h$$
$$\widetilde{\mathbf{I}} = [\mathbf{i}_{t_1}, \mathbf{i}_{t_2} \ldots, \mathbf{i}_{t_L}]^\top = [\mathbf{h}_{t_1}^H, \mathbf{h}_{t_2}^H \ldots, \mathbf{h}_{t_L}^H]^\top \in \mathbb{R}^{L \times d}$$

In a session, an item has only one learned embedding no matter whether it recurs in the sequence.

- Learning operation embeddings by GRU

$$\mathbf{o}_{t_k} = \mathbf{h}_{t_k} = GRU(\mathbf{h}_{t_{k-1}}, \mathbf{o}_{t_k}^0; \Phi_{GRU})$$
$$\widetilde{\mathbf{O}} = [\mathbf{o}_{t_1}, \mathbf{o}_{t_2}, \ldots, \mathbf{o}_{t_L}]^\top \in \mathbb{R}^{L \times d}$$

An operation recurring in an operation sequence has multiple different embeddings.

- Generating Micro-behavior embeddings

$$\widetilde{\mathbf{M}} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_L]^\top = [\mathbf{i}_{t_1} \oplus \mathbf{o}_{t_1}, \mathbf{i}_{t_2} \oplus \mathbf{o}_{t_2}, \ldots, \mathbf{i}_{t_L} \oplus \mathbf{o}_{t_L}]^\top \in \mathbb{R}^{L \times 2d}$$

Two sessions having the same item sequence but different operation sequences still have different representations.

SIGIR 2020
XI'AN·CHINA

Encoding Session Information

KNOWLEDGE WORKS

- Generating Session Representations
  - Adopt *soft-attention mechanism* to assign proper weight for each micro-behavior's embedding in the session

$$\alpha_t = \boldsymbol{\beta}^\top \sigma(\mathbf{W}_1 \mathbf{m}_L + \mathbf{W}_2 \mathbf{m}_t + \mathbf{b}_\alpha)$$

$\mathbf{m}_L$ is the embedding of the most recent micro-behavior.

  - Combine local preference and global preference

$$\mathbf{s}_g = \sum_{t=1}^{L} \alpha_i \mathbf{m}_t \qquad \mathbf{s} = \mathbf{W}_3[\mathbf{m}_L; \mathbf{s}_g] \in \mathbb{R}^d$$

  - Compute the final score

$$\hat{y}_{sj} = softmax\left(MLP(\mathbf{s} \oplus \mathbf{i}_j)\right)$$

  - SR prediction loss function

$$\mathcal{L}_S = -\sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{I}} \{y_{sj} \log(\hat{y}_{sj}) + (1 - y_{sj}) \log(1 - \hat{y}_{sj})\}$$

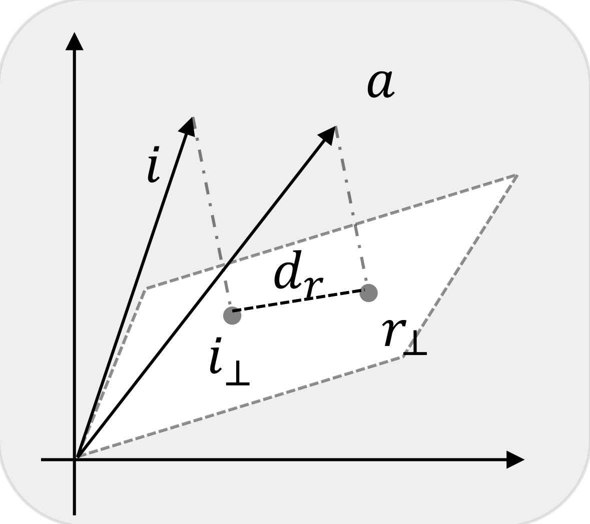
Learning Knowledge Embeddings

- Adopt TransH to learn knowledge embeddings since TransH handles many-to-one and many-to-many relations well.
  - Position a relation-specific translation vector $\mathbf{d}_r$ in the relation-specific hyperplane $\mathbf{w}_r$
  - Project item embedding $\mathbf{i}$ and attribute embedding $\mathbf{a}$ to the hyperplane

$$\mathbf{i}_\perp = \mathbf{i} - \mathbf{w}_r^\top \mathbf{i} \mathbf{w}_r \qquad \mathbf{a}_\perp = \mathbf{a} - \mathbf{w}_r^T \mathbf{a} \mathbf{w}_r$$

  - The loss function for knowledge embedding learning

$$\mathcal{L}_K = \sum_{<i,r,a>\in\mathcal{K}} \|(\mathbf{i} - \mathbf{w}_r^\top \mathbf{i} \mathbf{w}_r) + \mathbf{d}_r - (\mathbf{a} - \mathbf{w}_r^\top \mathbf{a} \mathbf{w}_r)\|_2^2$$



Multi-task Learning Objective

- MTL's objective: maximize the following posterior probability of our model's parameters Φ given knowledge triplet set $\mathcal{K}$ and SR's training set $\mathcal{Y}$.

$$\max p(\Phi|\mathcal{K},\mathcal{Y}) = \max \frac{p(\Phi,\mathcal{K},\mathcal{Y})}{p(\mathcal{K},\mathcal{Y})}$$

$$= \max p(\Phi)p(\mathcal{K}|\Phi)p(\mathcal{Y}|\Phi,\mathcal{K})$$

$$\mathcal{L} = \mathcal{L}_S + \lambda_1 \mathcal{L}_K + \lambda_2 \|\Phi\|_2^2$$

- We adopt *alternating training*

$$\mathcal{L}_{alter} = -\sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{I}}\left[y_{sj}\log(\hat{y}_{sj}) + (1-y_{sj})\log(1-\hat{y}_{sj})\right] +$$

$$\lambda_1 \sum_{<i,r,a>\in\mathcal{K}} \|(\mathbf{i}-\mathbf{w}_r^\top \mathbf{i}\mathbf{w}_r) + \mathbf{d}_r - (\mathbf{a}-\mathbf{w}_r^\top \mathbf{a}\mathbf{w}_r)\|_2^2 + \lambda_2\|\Phi\|_2^2$$

Outline

- Introduction
- Methodology
- Evaluation
- Conclusion

Research Questions

- **RQ1:** Can our model MKM-SR outperform the state-of-the-art SR models?
- **RQ2:** Is it useful to incorporate micro-behaviors and knowledge into our model?
- **RQ3:** Is it rational to obtain a session's representation through learning item embeddings by GGNN and learning operation embeddings by GRU separately?
- **RQ4:** Which training strategy is better for incorporating knowledge learning into MKM-SR?

SIGIR 2020
XI'AN·CHINA

KNOWLEDGE WORKS

Dataset Description

- **KKBOX**[1]
  - 4 music attributes: artist (singer), genre, language, release year
  - 23 operations
- **JDATA**[2]
  - 4 product attributes: brand, shop, category, launch year
  - 5 operations: clicking, ordering, commenting, adding to cart and favorite
- **Demo**: It is a sparse JDATA dataset through retaining the early 1% user behaviors, resulting in a bigger proportion of cold-start items.

Table 1: Dataset statistics

	seesion#	session length	item#(new%)	item frequency	operation#
KKBOX	180,047	4.713	33,454	25.365	23
KKBOX(N)	180,096	4.726	34,120(0.99%)	24.942	23
JDATA	455,481	5.372	134,548	18.186	5
JDATA(N)	456,005	5.383	139,099(3.69%)	17.654	5
Demo	5,633	5.330	12,195	2.301	5
Demo(N)	5,696	4.992	12,917(40.45%)	2.192	5

'(N)' indicates the datasets having some cold-start items, and 'new%' is the proportion of the behaviors involving cold-start items to all behaviors in the test set.

1 https://www.kaggle.com/c/kkbox-music-recommendation-challenge/data
2 https://jdata.jd.com/html/detail.html?id=8

Experiment Settings

- Evaluation metrics

  Hit@k: The proportion of the correct next interacted item in the top-k ranking lists.

  MRR@k: The average reciprocal rank of the correct next interacted item in the top-k ranking list.

- Compared models
  - **FPMC**: Sequential prediction method based on personalized Markov chain
  - **GRU4REC+BPR/CE**: The improved versions of GRU4REC with different loss function
  - **NARM**: GRU-based SR model with an attention to consider the long-term dependency of user preferences
  - **STAMP**: SR model considering both current interests and general interests of users
  - **SR-GNN**: It utilizes graph network to capture complex transition patterns among the items in a session
  - **RIB**: GRU-based model incorporating user operations of which the embeddings are learned by Word2Vec

Experiment Results

	KKBOX		KKBOX(N)		JDATA		JDATA(N)		Demo		Demo(N)	
	Hit@20	MRR@20	Hit@20	MRR@20	Hit@20	MRR@20	Hit@20	MRR@20	Hit@20	MRR@20	Hit@20	MRR@20
FPMC	5.614	1.166	5.530	1.147	7.531	2.623	7.049	2.493	3.787	1.808	3.261	1.644
GRU4REC+BPR	12.795	4.545	12.501	4.693	35.433	13.262	34.827	13.346	12.189	5.124	5.567	2.969
GRU4REC+CE	12.445	4.007	12.429	4.135	35.347	13.956	34.794	13.542	12.965	4.992	9.896	4.505
NARM	14.667	5.839	13.926	5.200	36.867	16.826	35.862	16.677	14.446	5.645	8.056	3.615
STAMP	14.475	4.783	14.287	4.544	35.555	12.936	34.691	12.187	14.609	5.796	9.317	2.902
SR-GNN	14.187	4.476	13.399	4.792	40.588	15.968	38.723	15.203	15.504	7.220	10.317	4.682
RIB	15.982	4.763	13.887	5.328	37.236	14.134	35.551	13.420	12.893	4.887	9.965	4.436
KM-SR	17.680	7.195	17.019	6.301	41.094	16.552	40.480	15.709	23.726	9.363	15.065	6.323
M(GRU)-SR	16.971	5.435	16.865	5.250	37.015	14.034	36.374	13.734	18.507	6.430	11.262	4.747
M(GGNN)-SR	13.262	4.347	13.035	4.098	38.270	16.532	37.231	15.663	16.141	6.811	9.168	3.572
M(GGNNx2)-SR	17.270	5.532	16.983	5.435	41.017	16.544	41.308	15.780	19.782	7.865	12.017	4.734
M-SR	20.998	5.878	20.523	5.707	41.440	16.851	41.019	15.850	20.631	7.969	12.914	5.228
MKM-SR	**22.574**	**7.543**	**22.221**	**6.976**	**42.565**	**17.585**	**41.998**	**16.990**	**24.623**	**9.642**	**15.110**	**6.424**

Table 2: All models' SR performance scores (percentage value) show that MKM-SR outperforms all competitors no matter whether the historical interactions are sparse.

Ablation Study

- Ablated models
  - **KM-SR**: All modules related to operations are removed
  - **M-SR**: The auxiliary task of learning knowledge embeddings is removed
  - **M(GRU/GGNN)-SR**: directly learn micro-behavior embeddings
  - **M(GGNNx2)-SR**: uses two GGNNs to learn operation embeddings and item embeddings respectively
- Experiment conclusions
  - Both micro-behaviors (operations) and item knowledge deserve to be incorporated w.r.t. improve SR performance (answer yes to RQ2).
  - Modeling a session through learning item embeddings and learning operation embeddings separately is more effective than learning micro-behavior embeddings directly.
  - Operation sequences should be learned by GRU rather than GGNN.

Experiment Results

- Strategies of MTL training

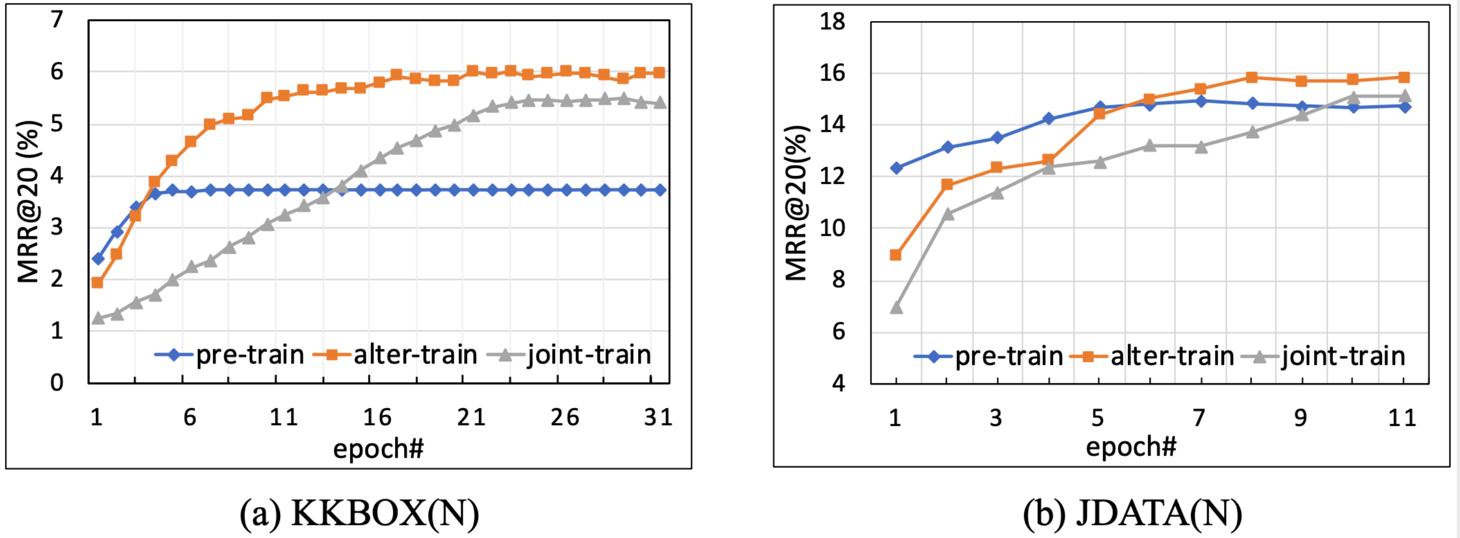
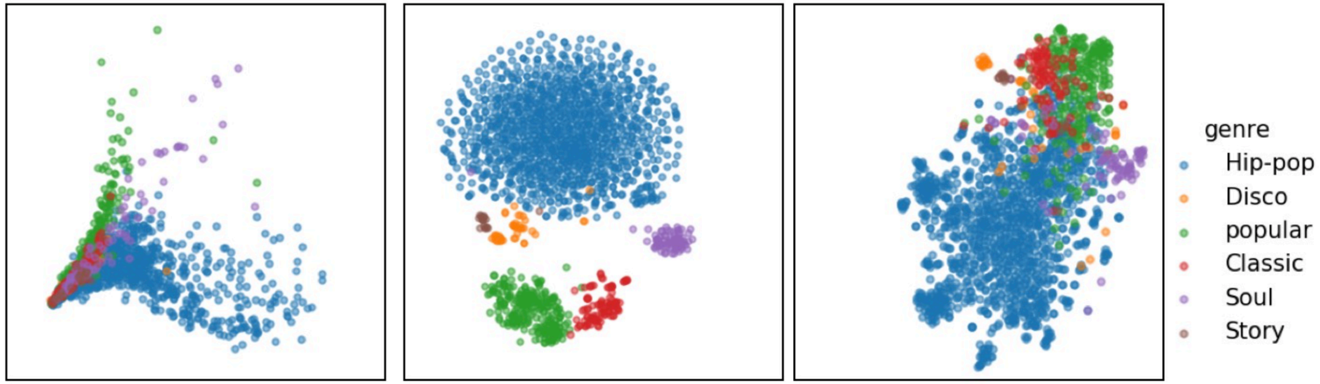


Figure 3: The learning curves of three different strategies to incorporate knowledge learning

- Pre-training model has a better learning start, but is overtaken by the other two rivals on the stage of convergence.
- The items often occurring in the sessions of training set will be tuned multiple times by loss $\mathcal{L}_K$ in each epoch of joint training. It makes the learned embeddings bias to auxiliary task $\mathcal{L}_K$ too much, shrinking the learning effect of the main task $\mathcal{L}_S$.
- Therefore, alternating training is the best strategy for our SR task.

Experiment Results

- Strategies of Incorporating Knowledge Learning



(a) Word2Vec (b) TransH (c) MKM-SR

Figure 4: KKBOX(N)'s item embedding distributions under different learning mechanisms

Fig. 4(a): Feeding item sequences of sessions into Word2Vec. Two items are close in the space if they often cooccur in some sessions. Thus, many songs of different genres too converged and thus can not discriminate different genres.
Fig. 4(b): Item embeddings were learned solely by TransH. It makes the model based on embedding distances hard to predict the item of different genre as the next interacted item.
Fig. 4(c): It is learned by MKM-SR through MTL, and exhibits two characteristics, i.e., they can discriminate different genres for most items, meanwhile keep close distances across different genres.

Outline

- Introduction
- Methodology
- Evaluation
- Conclusion

Conclusion

- We propose a novel session-based recommendation model
- We adopt different mechanisms to learn item embeddings and operation embeddings
- Our experiments highlight that, incorporating micro-behaviors and knowledge can further improve SR performance

Thank YOU！

Our LAB: **Knowledge Works** at Fudan University

http://kw.fudan.edu.cn

Personal page: http://kw.fudan.edu.cn/people/yangdeqing

Source and dataset: https://github.com/ciecus/MKM-SR