Deep Learning-based Sequential Recommender Systems: Concepts, Algorithms, and Evaluations

Hui Fang, Guibing Guo, **Danning Zhang**, and Yiheng Shu

2019.6.11

Contents

Research background

Why sequential recommendation?

- RS are widely used in many fields .
- effectively address information overload problems .
- The records form always is sessions

Traditional RS fail to consider the 'time' information

Sequential RS not only capture user's long-term preferences, but also model sequential dependencies among interactions.

Research background

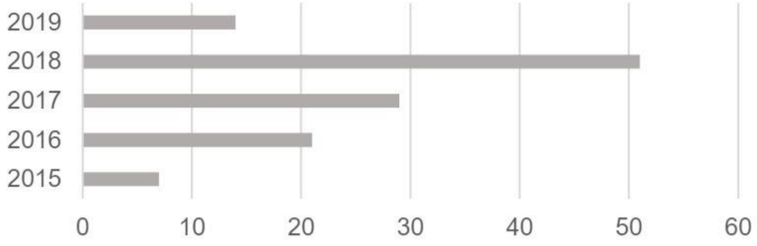
Why study deep learning techniques?

Traditional sequential recommendation methods fail to thoroughly model user’s long-term patterns

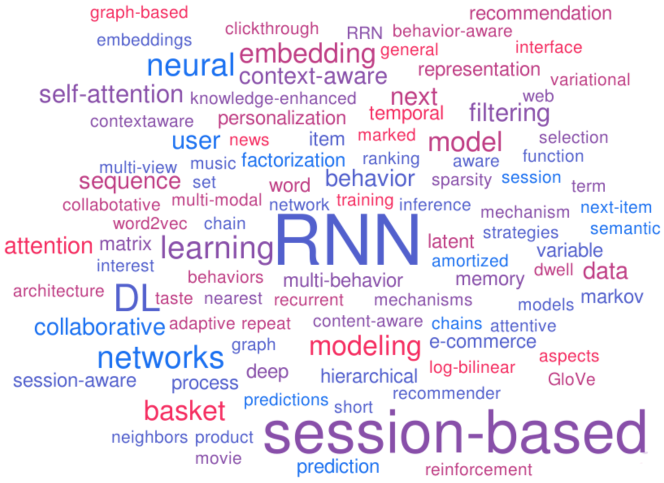
- The architectures of DL models are suitable for modeling sequential information.
- The successes in NLP prove their advantages.

Many DL-based models have achieved state-of-the-art performance.

Research background



the number of sequential recommendation related articles published on arXiv in recent five years



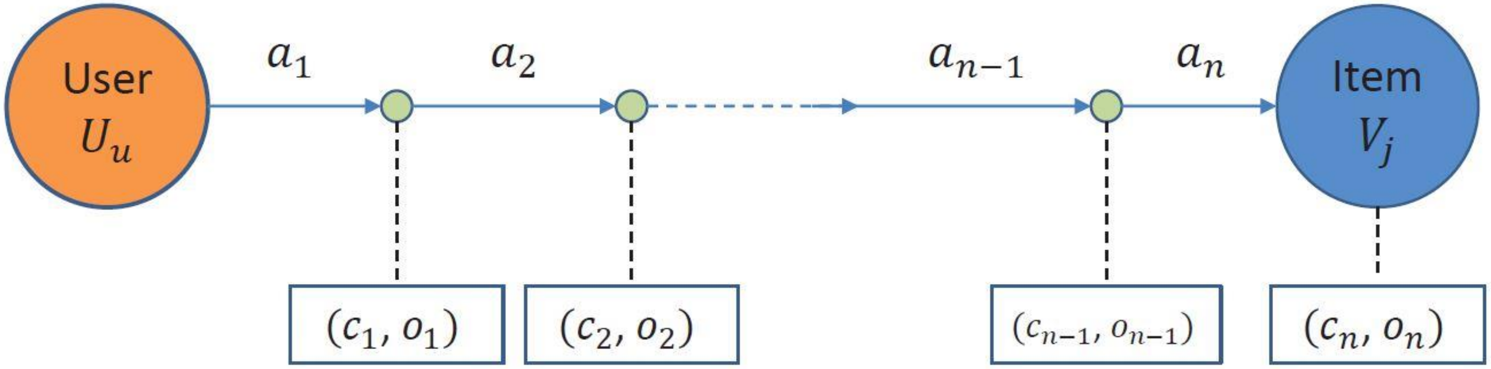
the word cloud of the keywords in DL-based sequential recommendation related articles

- The number of relevant arXiv articles grows year by year
- The interest in sequential recommendation has increased phenomenally
- The word with the highest word frequency is RNN
- Most models are based on session information

Sequential recommendation

1. Behavior Object   Items or services that a user chooses to interact with
2. Behavior Type   The way that a user interacts with items or services

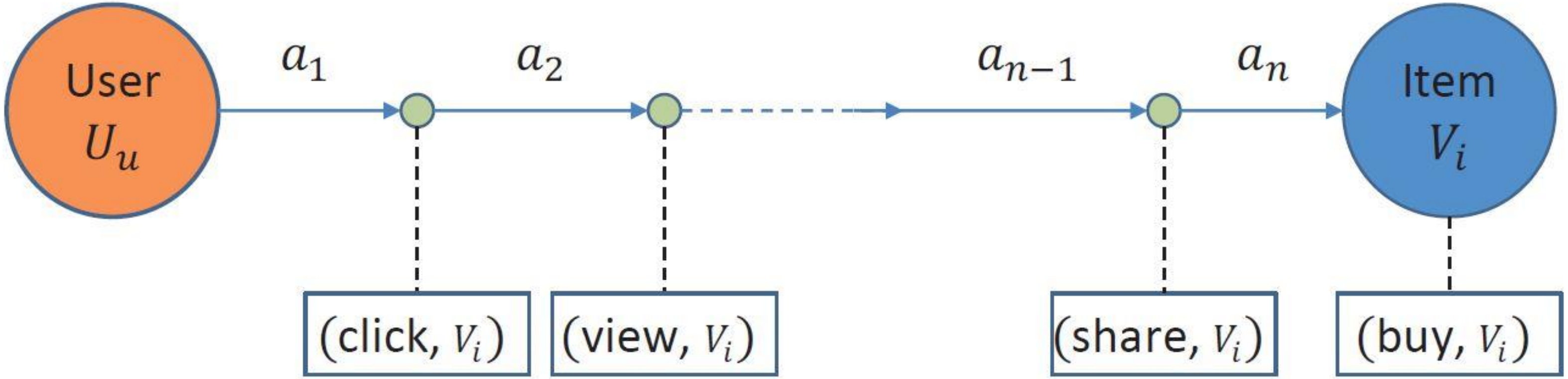
Behavior: a combination of behavior type and behavior object.
Behavior trajectory : a behavior sequence consisting of multiple user behaviors.
Sequential recommender system : convert user's behavior trajectory into recommended items or services.



Sequential recommendation

Experience-based behavior sequence

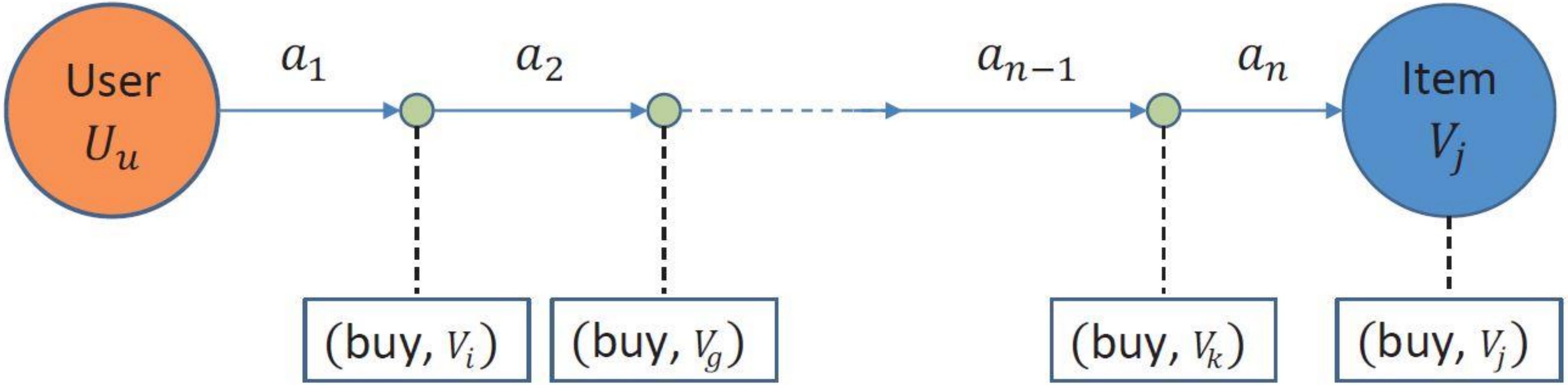
same object , different behavior types



Sequential recommendation

Transaction-based behavior sequence

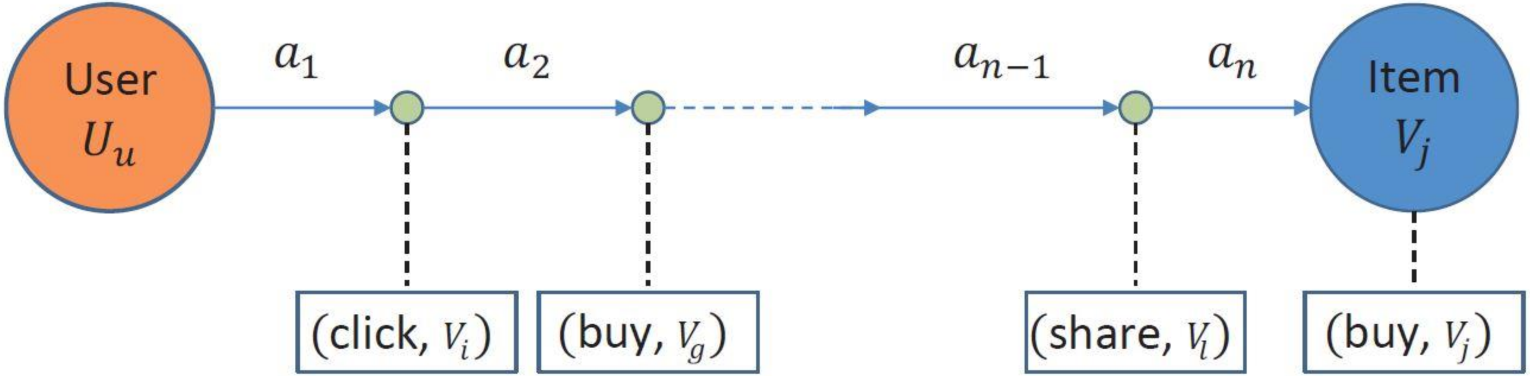
different items, same behavior type (i.e. buy)



Sequential recommendation

Interaction-based behavior sequence

multiple items, different behavior types



Categorizations

Based on behavior type (input sequence types)

- Experience-based sequential recommendation

  Take **experience-based behavior sequence** as input
  Predict the **next behavior type** the user will impose on the given item.

- Transaction-based sequential recommendation

  Take **transaction-based behavior sequence** as input
  Predict the **next item(s)** the user will buy

- Interaction-based sequential recommendation

  Take **interaction-based behavior sequence** as input
  Predict the **next item(s)** the user will interact with

Categorizations

Based on behavior object

- Next-item recommendation

In **next-item recommendation**, a user behavior contains only one object (i.e. item).
Model sequential dependencies among behaviors.

- Next-basket recommendation

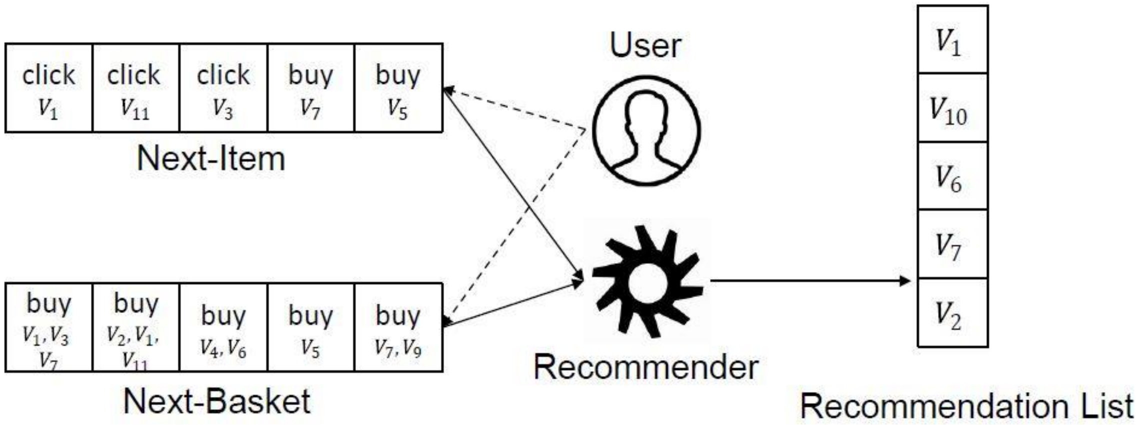
In **next-basket recommendation**, a user behavior contains more than one objects. We call a behavior as a basket.
Model correlations among items in the same basket as well as the sequential dependencies among baskets.

Categorizations

Based on behavior object

Next-item recommendation &

Next-basket recommendation



Related techniques

Traditional methods

- Frequent pattern mining

$$score_{FPM}(i,s) = \frac{1}{\sum_{p \in S_p} \sum_{x=2}^{|p|} 1_{EQ}(s_{|s|}, p_x) \cdot x} \sum_{p \in S_p} \sum_{x=2}^{|p|} \sum_{y=1}^{x-1} 1_{EQ}(s_{|s|}, p_y) \cdot 1_{EQ}(i, p_x) \cdot w(x-y)$$

- easy to implement and relatively explicable for user
- time-consuming when matching patterns and hard to determine threshold

- K-nearest neighbor

$$score_{SKNN}(i,s) = \sum_{n \in N_s} sim(s,n) \cdot 1_n(i)$$

- make explainable recommendation
- the similarities can also be pre-calculated
- sequential dependencies among items are ignored

Related techniques

Traditional methods

- Markov Chain

$$score_{MC}(i,s) = \frac{1}{\sum_{p\in S_p}\sum_{x=1}^{|p|-1} 1_{EQ}(s_{|s|},p_x)}\sum_{p\in S_p}\sum_{x=1}^{|p|-1} 1_{EQ}(s_{|s|},p_x)\cdot 1_{EQ}(i,p_{x+1})$$

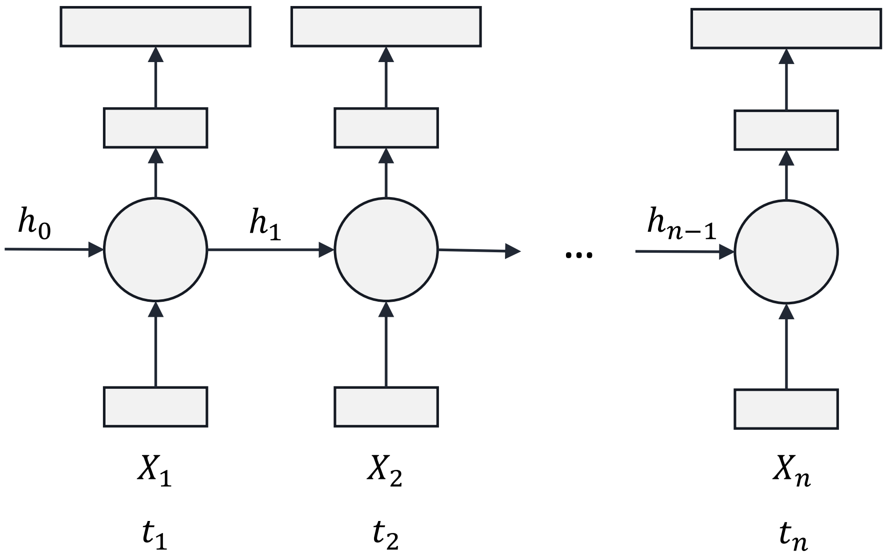
  - can model sequential dependency
  - only consider the last or last few behaviors, fail to capture intricate dynamic in a long sequence

- Factorization-based methods
  - computation-cost
  - ignore sequential dependency

Related techniques

Deep learning techniques

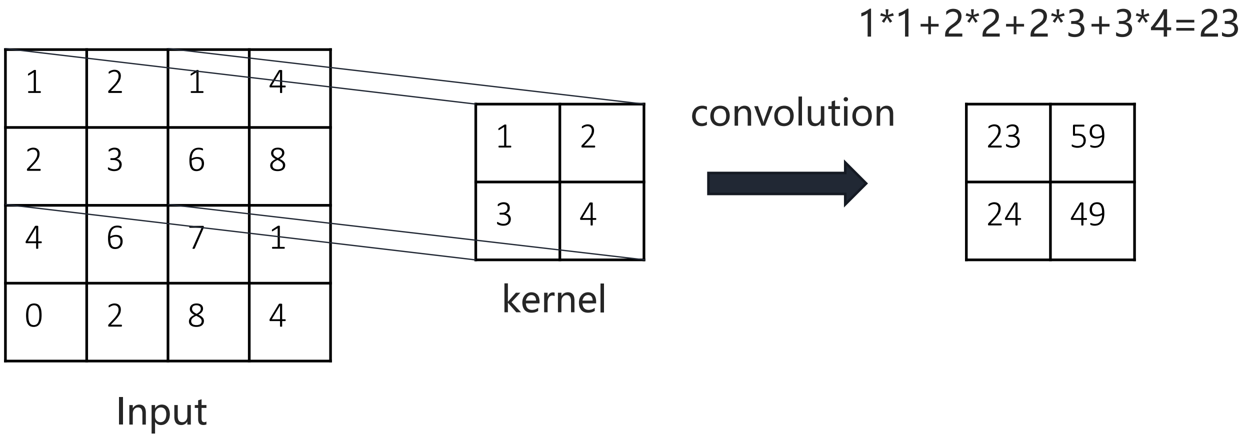
- RNN
  - suitable for modeling sequential data
  - training cost increases for long sequences



Related techniques

Deep learning techniques

- CNN
  - suitable to capture the dependent relationship across local information

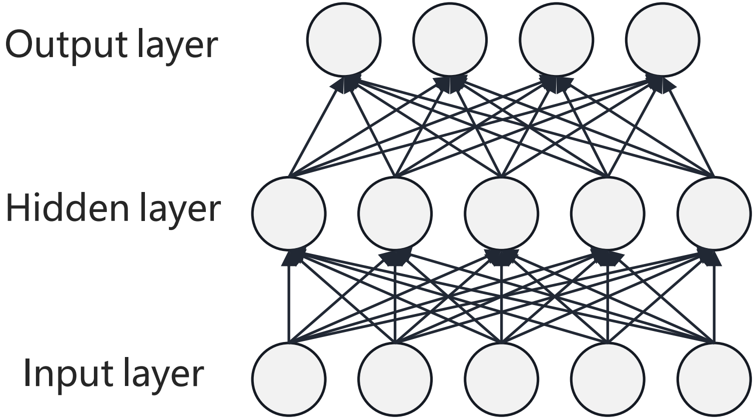


window size=2 step size =2

Related techniques

Deep learning techniques

- MLP
  - active function can be linear, tanh, relu, and so on.
  - learn non-linear relationship



Related techniques

Deep learning techniques

- Attention mechanism
  - can capture more important parts of the target object
  - include vanilla attention and self-attention

$$a_t = align(m_t, m_s) = \frac{\exp(f(m_t, m_s))}{\sum_{s'} \exp(f(m_t, m_{s'}))}$$

$$f(m_t, m_s) = \begin{cases} m_t^T m_s \\ m_t^T W_a m_s \\ v_a^T \tanh(W_a m_t + U_a m_s) \end{cases}$$

Vanilla attention

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Self-attention

Related techniques

Advantages

- utilize much longer sequences, and are effective for theme learning
- DL methods are more flexible, robust to sparse data
- can adapt to varied length of the input sequence

Disadvantages

- lack of explanability.
- The optimization is generally very challenging
- more training data is required for complex network.

DL-based Algorithms

In this section, we introduce DL-based sequential recommender systems based on the three types of recommendations mentioned before : experience-based sequential recommendation, transaction-based sequential recommendation and interaction-based sequential recommendation.

We will introduce the representative algorithms under each type in detail.

RNN Model (MCBD)

- A buying decision process describes a number of stages a consumer goes through before and after buying a particular product.
- Existing recommender systems do not explicitly model the consumer buying decision process
- multi-task learning model with LSTM to learn consumer buying decision process.
- Prediction tasks: If direct buying and next stage predictions. makes recommendations accordingly.

Stage design rules:
Need-recognition stages: first click
Research stage: look-at-comments, ask-the-seller or look-at-QuestionAll behavior after click
Consideration stage: add-to-cart or mark-as-favorite behavior after click
Buying stage: buy after click
Feedback stage: comment after click

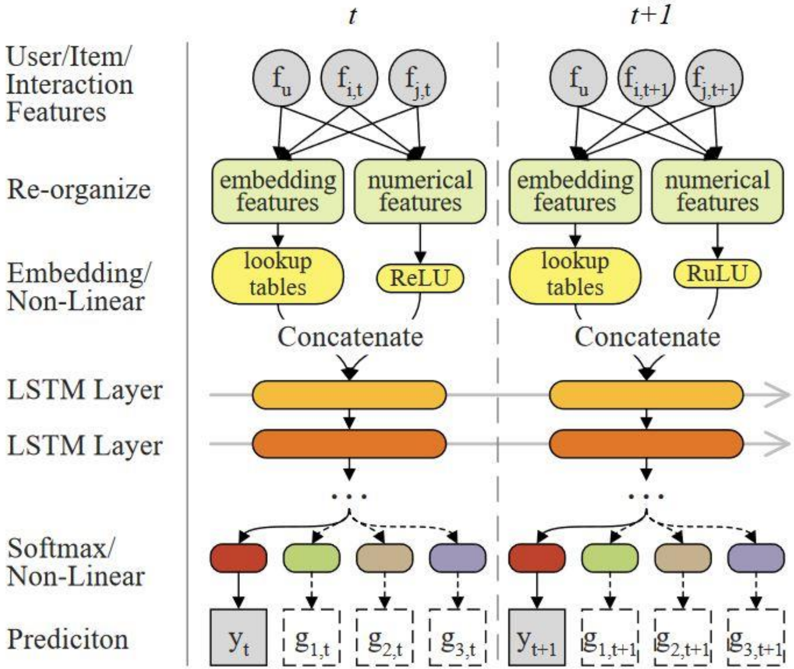
Q. Xia, P. Jiang, F. Sun, Y. Zhang, X. Wang, and Z. Sui, “Modeling consumer buying decision for recommendation based on multi-task deep learning,” in CIKM, 2018, pp. 1703–1706.

RNN Model (MCBD)

- **Input** :Item feature, User feature and Interaction feature
  Output: stage and if direct buying

$$P(\emptyset_k | t, c_1^u, c_2^u, \ldots, c_t^u) = g_{k,t} = \sigma(V_k h_t + b_k)$$

$$P(\omega_i | t, c_1^u, c_2^u, \ldots, c_t^u) = y_{i,t} = \frac{e^{(W_s h_t + b_s)i}}{\sum_{j\in\Omega} e^{(W_s h_t + b_s)j}}$$



DL-based Algorithms

Experience-based Sequential Recommendation

RNN Model (GRU4Rec)

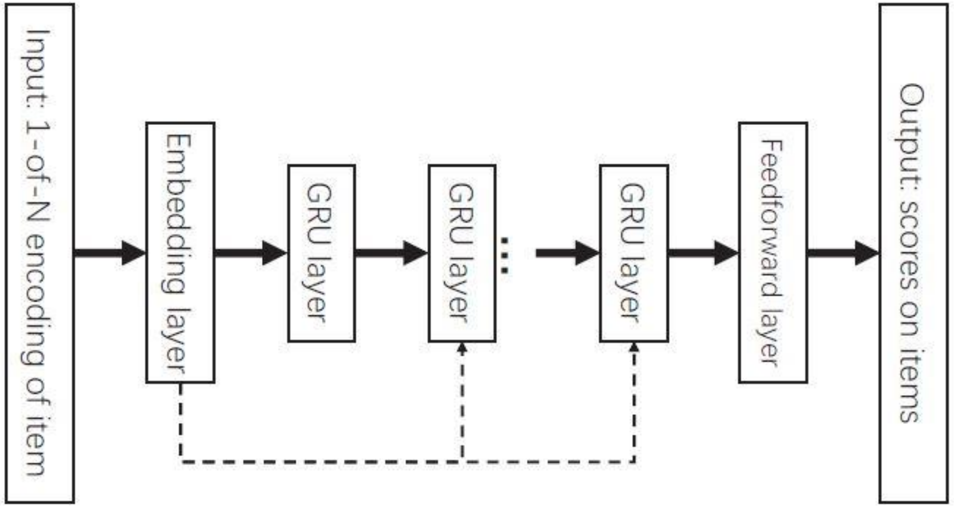
- The first model that applies RNN to sequential recommendation. Lots of articles choose GRU4Rec as their baselines.

- - It utilizes the memory function of RNN to model sequential dependencies of sessions
  - deals with the issues that arise when modeling sparse sequential data
  - adapt the RNN models to the recommender setting by introduce a new ranking loss function (TOP1)
  - propose a new mini-batch method(session parallel mini-batch for training)

B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, “Session-based recommendations with recurrent neural networks,” arXiv preprint arXiv:1511.06939, 2015.

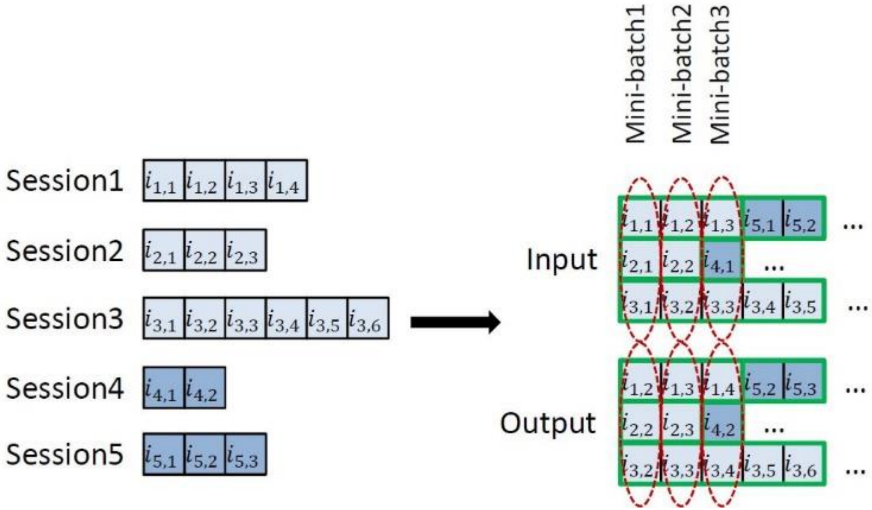
RNN Model (GRU4Rec)

- The core of the model is the GRU layer(s).



RNN Model (GRU4Rec)

- Session parallel mini-batch



RNN Model (GRU4Rec)

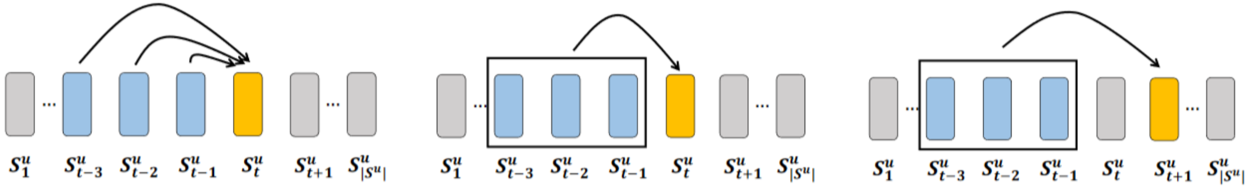
- Loss function

  - BPR $$L_s = -\frac{1}{N_s} \cdot \sum_{j=1}^{N_s} \log(\sigma(\hat{r}_{s,i} - \hat{r}_{s,j}))$$

  - TOP1 $$L_s = \frac{1}{N_s} \cdot \sum_{j=1}^{N_s} \sigma\left(\hat{r}_{s,i} - \hat{r}_{s,j}\right) + \sigma(\hat{r}_{s,j}^2)$$

CNN Model (caser)

- Previous works fail to explicitly capture union level sequential patterns.
- Fail to allow skip behaviors



Point-level | union-level, no skip | union-level, skip once

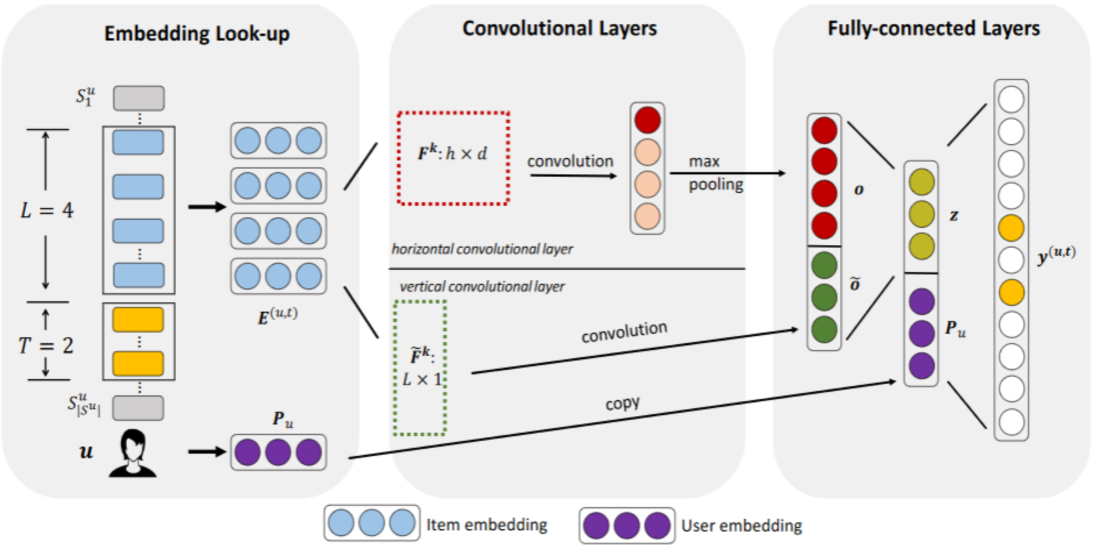
J. Tang and K. Wang, “Personalized top-n sequential recommendation via convolutional sequence embedding,” in WSDM, 2018, pp. 565–573.

DL-based Algorithms

Transaction-based Sequential Recommendation

CNN Model (caser)

- Caser views the embedding matrix of L previous items as an 'image'
- uses horizontal convolutional layer and vertical convolutional layer to capture point-level and union-level sequential patterns.
- captures long-term user preferences through user embedding.



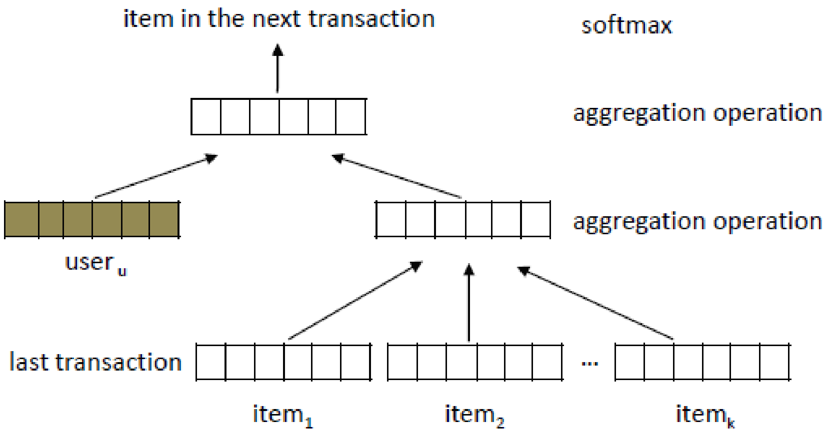
DL-based Algorithms

Transaction-based Sequential Recommendation

MLP Model (HRM)

- non-linear operations for complex correlations between user's behavior and relationships between user's short-term interest and her long-term preference.

- - The core is the two aggregation layers
  - aggregation operation can be either average pooling or max pooling.



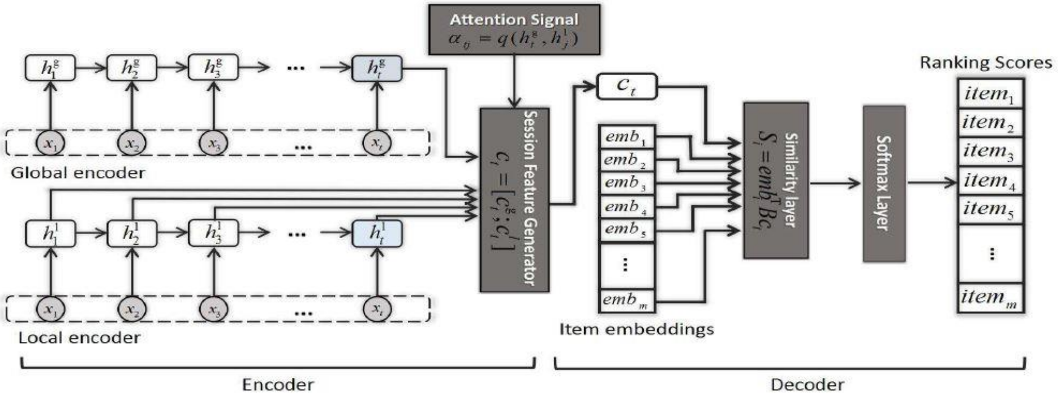
P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for next basket recommendation," in SIGIR, 2015, pp. 403–412.

DL-based Algorithms

Transaction-based Sequential Recommendation

Attention Model (NARM)

- Previous works only focus on sequential dependency, ignore the user's main purpose.
- NARM incorporates RNN with attention mechanism to model sequential dependencies as well as capture user's main purpose in the current sequence.
- An encoder-decoder framework, consisting of two sub-encoders: global encoder and local encoder.



J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in CIKM, 2017, pp. 1419–1428.

DL-based Algorithms

Transaction-based Sequential Recommendation

RNN Model (CBS)

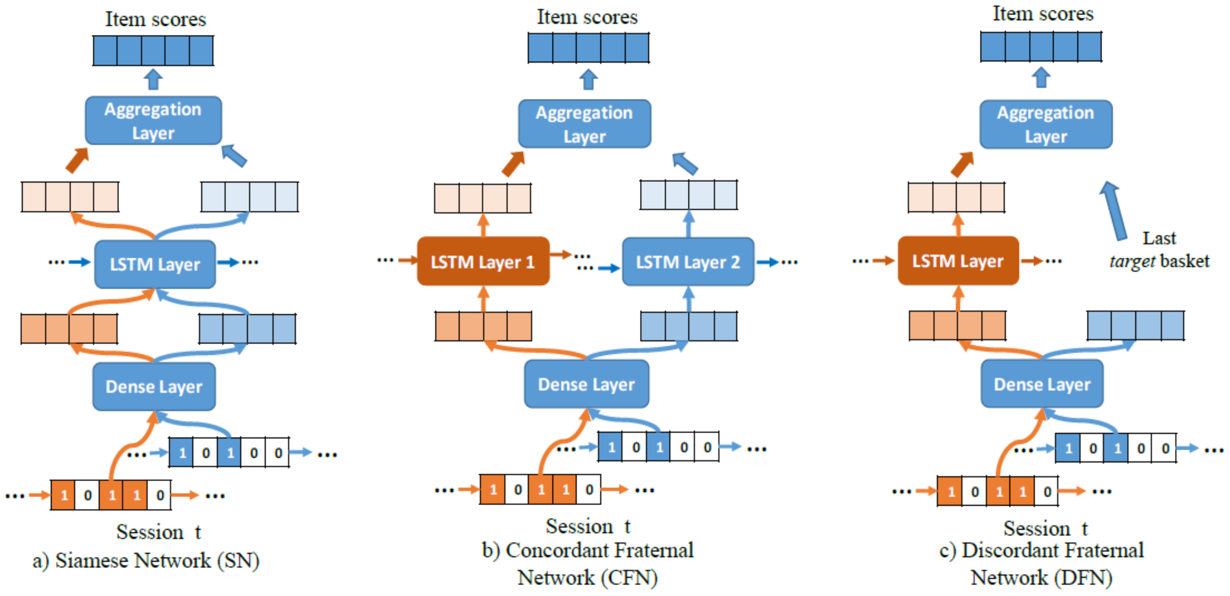
- Most of previous works in modeling behavior sequence are preoccupied with only one sequence type.

- The basis idea of this model is that the target behavior (e.g., purchase) contains the most efficient information for the prediction task, and the remaining behaviors (e.g., click) can thus utilized as the support sequence that can facilitate and assist the next-item prediction task in target sequence.

- It proposes three assumptions and designs one specific structures for each assumption.

D.-T. Le, H. W. Lauw, and Y. Fang, "Modeling contemporaneous basket sequences with twin networks for next-item recommendation," in IJCAI, 2018, pp. 3414–3420.

RNN Model (CBS)



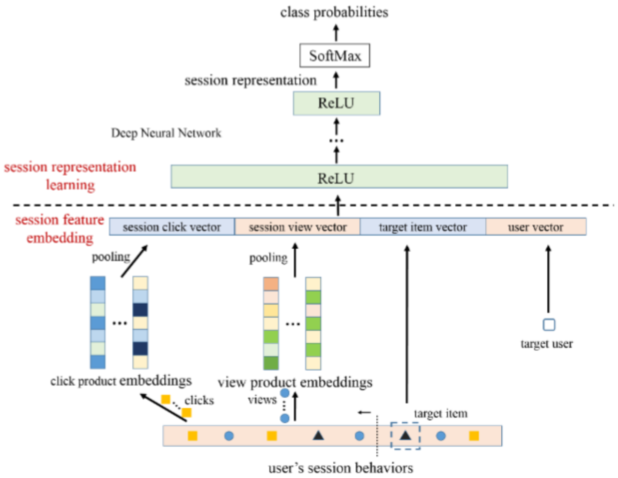
- Different sequence types reflect the same underlying phenomenon
- Different sequence types reflect different underlying phenomenon, but the sequential dependencies are same
- Browsing and clicking may have longer-term dependency than purchases

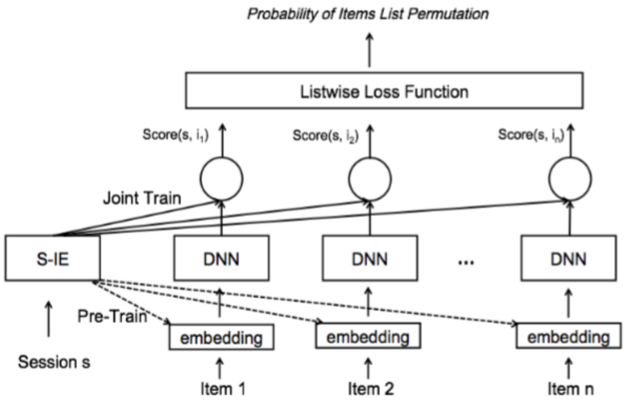
DL-based Algorithms

Interaction-based Sequential Recommendation

MLP Model

- consist of two parts: SIE and list-wise ranking.
- The SIE part is for pretraining a session representation and item embeddings.
- List-wise ranking model calculates relevance scores between user’s session and candidate items





C. Wu and M. Yan, “Session-aware information embedding for e-commerce product recommendation,” in CIKM, 2017, pp. 2379–2382.

DL-based Algorithms

Interaction-based Sequential Recommendation

Attention Model (ATRank)

- ATRank considers polymorphism of user behaviors, utilizes both self-attention and vanilla attention mechanisms to model it.
- It divides behaviors in a sequence into different groups in terms of behavior type, and then projects all types of behaviors into multiple latent semantic spaces
- It argues that heterogenous behaviors could have very different power. Thus, their embedding spaces could be in both different sizes and meanings
- $u_{ij} = emb_i(o_j) + lookup_i^i\left(bucketize_i(t_j)\right) + lookup_i^a(a_j)$

$$B = \{u_{bg1}, u_{bg2}, \dots, u_{bgn}\}$$

$$S = concat^{(0)}(F_{M_1}(u_{bg1}), F_{M_2}(u_{bg2}), \dots, F_{M_n}(u_{bgn}))$$

$$S_k = F_{P_k}(S)$$

$$A_k = softmax(a(S_k, S; \theta_k))$$

$$a(S_k, S; \theta_k) = S_k W_k S^T$$

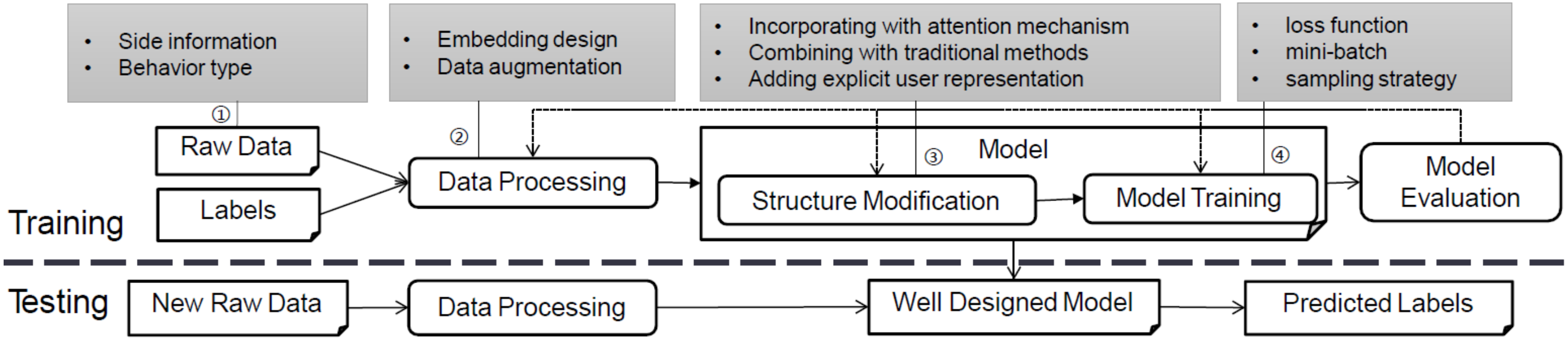
C. Zhou, J. Bai, J. Song, X. Liu, Z. Zhao, X. Chen, and J. Gao, “Atrank: An attention-based user behavior modeling framework for recommendation,” arXiv preprint arXiv:1711.06632, 2017.

DL-based Algorithms

Interaction-based Sequential Recommendation

Influential Factors

Based on the flow of the designation of a recommender system, we summarize influential factors in each module



Influential factors

Input module

- Side Information

  - information: item description context, images, and so on. User-related information: User profiles. Transition-related information: dwell time.
  - P-RNN* exceeds GRU4Rec by 1.1%

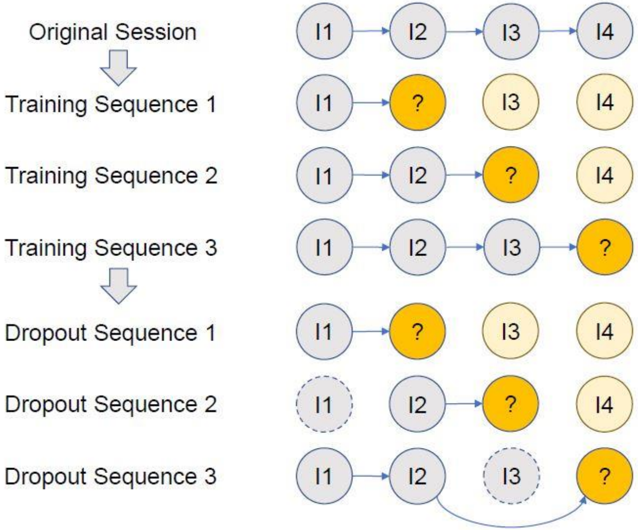
- Behavior type

  - Behaviors are usually heterogeneous and polysemous.
  - Project different types of behaviors into different embedding spaces.
  - A specially designed network for a certain behavior type (i.e, purchase)

* B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, “Parallel recurrent neural network architectures for feature-rich session-based recommendations,” in RecSys, 2016, pp. 241–248.

- Embedding Designs
  - adopt pre-training model in NLP (BERT).
  - w-item2vec* (inspired by word2vec).
  - design a session embedding for pre-training.

- Data augmentation



* P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for next basket recommendation," in SIGIR, 2015, pp. 403–412.

* Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in DLRS, 2016, pp. 17–22.

Influential factors

Data process module

Influential factors

Model structure module

- Incorporating Attention Mechanism
  - incorporating CNN or RNN with vanilla attention
  - just building a self-attention model for sequential recommendation

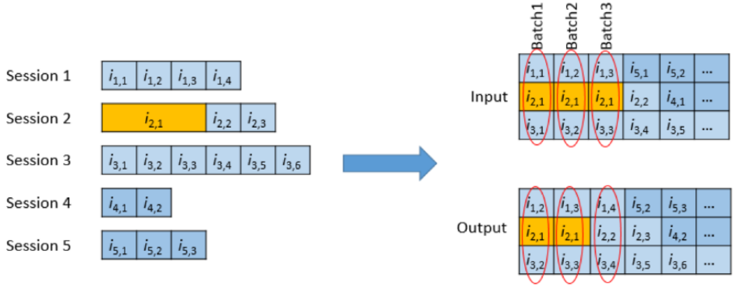
- Combining with conventional methods
  - Jannach et al, combines session-based KNN with GRU4Rec
  - AttRec combines self-attention and metric learning

- Adding explicit user representation
  - learning a simple embedding matrix for users while training the model(User embedding models)
  - design a specific network, dynamically model user representation (user recurrent models)

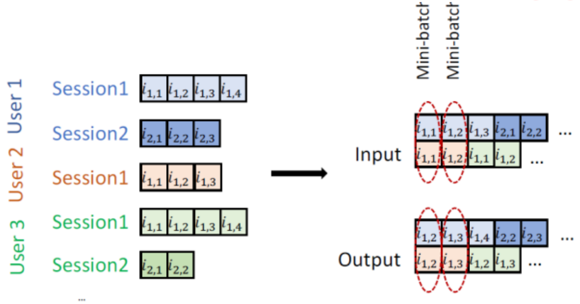
Influential factors

Model training module

- Negative sampling
  - uniform negative sampling
  - popularity-based negative sampling
  - Negative sample size.
- Mini-batch creation
  - Session-parallel mini-batch
  - Two variants: item boosting and user-parallel mini-batch



Item boosting



User-parallel

- Loss function

$$L_{BPR} = -\frac{1}{N}\cdot\sum_{j=1}^{N}\log(\sigma(\hat{r}_i - \hat{r}_j))$$

$$L_{TOP1} = \frac{1}{N}\cdot\sum_{j=1}^{N}\sigma\left(\hat{r}_i - \hat{r}_j\right) + \sigma(\hat{r}_j^2)$$

$$L_{BPR-max} = -log\sum_{j=1}^{N}s_j(\sigma(\hat{r}_i - \hat{r}_j))$$

$$L_{TOP1-max} = \sum_{j=1}^{N}s_j(\sigma\left(\hat{r}_i - \hat{r}_j\right) + \sigma\left(\hat{r}_j^2\right))$$

$$L_{XE} = -\sum_{i\in C}y_i log\widehat{y}_i + (1 - y_{i)})\log(1 - \widehat{y}_i)$$

Influential factors

Model training module

Datasets

Feature	RSC15	RSC19	RSC19 (user)	LastFM
Sessions	7,981,581	356,318	1,885	23,230
Items	37,483	151,039	3,992	122,816
Behaviors	31,708,461	3,452,695	49,747	683,907
Users	–	279,915	144	277
ABS	3.97	9.69	26.39	29.44
ASU	–	1.27	13.09	83.86

AES: Average Behaviors per Session
ASU: Average Sessions per User

Influential factors

Experiment results

Influential factors

Experiment results

- Recall : the coverage of the corrected recommended items in terms of target items
- MRR : how well a model ranks the target item.
- MAP : a high MAP indicates that items in ground-truth list appear at a higher ranking orders in the top-k recommended list.
- NDCG : a high NDCG implies that the order in which an item appear in the top-k recommendation list is close to its order in ground-truth list.

Model	RSC15				Model	RSC19			
	Recall@20	MRR@20	MAP@20	NDCG@20		Recall@20	MRR@20	MAP@20	NDCG@20
GRU4Rec	0.53621	0.19788	0.00742	0.04701	GRU4Rec	0.60346	0.38475	**0.00275**	**0.01775**
C-GRU	**0.54664**	0.19832	0.00884	0.05318	B-GRU	**0.61484**	**0.38901**	0.00216	0.01428
P-GRU	0.54356	**0.20483**	**0.00887**	**0.05322**					

- C-GRU: consider item category, concatenate
- P-GRU: consider item category, parallel networks
- B-GRU: consider behavior type

- Both C-GRU and P-GRU outperforms GRU4Rec on all evaluation metrics.
- B-GRU outperforms on Recall and MRR, but performs worse on MAP and NDCG. The main reason might be that RSC19 only contains four behavior types and one of them accounts for 62%

Influential factors

Experiment results

Factor	Variable	RSC15				RSC19			
		Recall@20	MRR@20	MAP@20	NDCG@20	Recall@20	MRR@20	MAP@20	NDCG@20
Dwell time	0	0.71820	0.31448	**0.01012**	0.05698	0.75335	0.55942	**0.00241**	0.01254
	(75, 45)	**0.88276**	**0.70885**	0.00491	0.07217	**0.89598**	**0.78898**	0.00109	**0.01442**
	(100, 60)	0.86111	0.65478	0.00579	**0.07380**	0.87224	0.75365	0.00116	0.01195
Data augmentation	Off	0.71820	0.31448	0.01012	**0.05698**	0.75335	0.55942	**0.00241**	**0.01254**
	On	**0.71836**	**0.31493**	**0.01013**	0.05692	**0.75638**	**0.56547**	0.00223	0.01075
Attention mechanism	Off	0.67886	0.27126	**0.00889**	**0.05868**	0.65055	0.41590	0.00162	**0.00946**
	On	**0.69827**	**0.30292**	0.00878	0.05542	**0.65623**	**0.41735**	**0.00164**	0.00885
KNN weight	0	0.71820	0.31448	0.01012	**0.05698**	0.75335	0.55942	**0.00241**	**0.01254**
	0.1	0.72022	**0.31547**	0.01308	0.05183	0.75675	0.56576	0.00128	0.00689
	0.3	**0.72307**	0.31315	**0.01340**	0.05206	**0.76662**	**0.57872**	0.00132	0.00696

- dwell time can greatly improve the performance.
- Model with data augmentation outperforms the basic model in terms of most metrics except NDCG.
- Incorporating attention mechanism enhances the performance of the model almost for all the scenarios, except NDCG.
- KNN weight of 0.3 provides better performance than that of 0.1

Influential factors

Experiment results

Influential factors

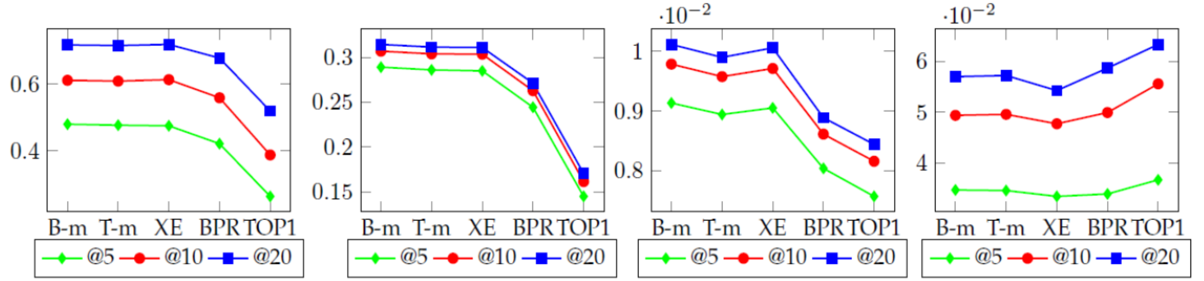
Experiment results

Factor	Variable	LastFM				RSC19 (user)			
		Recall@20	MRR@20	MAP@20	NDCG@20	Recall@20	MRR@20	MAP@20	NDCG@20
User Representation	Implicit	**0.16996**	**0.12496**	0.00408	0.08126	**0.67981**	**0.56814**	0.01452	0.08368
	Embedded	0.01634	0.00436	0.00837	0.21537	0.00479	0.00378	0.00773	0.20750
	Recurrent	0.00346	0.00058	**0.01230**	**0.42749**	0.06276	0.03058	**0.04508**	**0.79612**

- sharp decrease on Recall@20 and MRR@20, whether embedded or recurrent one.
- in terms of NDCG@20 and MAP@20, user representation models greatly outperform the basic model
- the user recurrent model performs better than the user embedded model

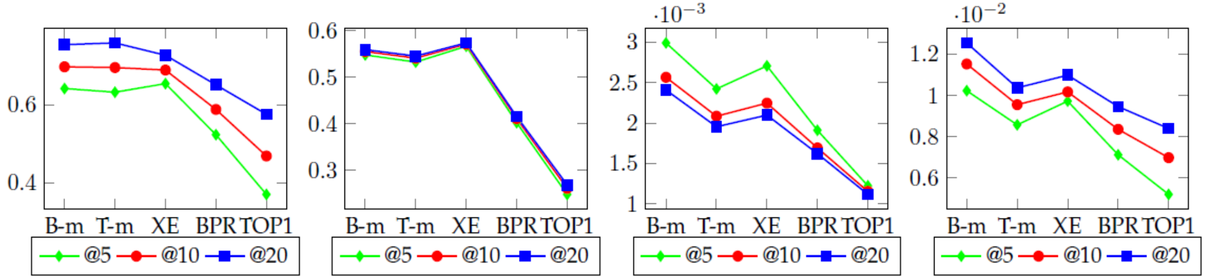
Influential factors

Experiment results



(a) Recall (b) MRR (c) MAP (d) NDCG

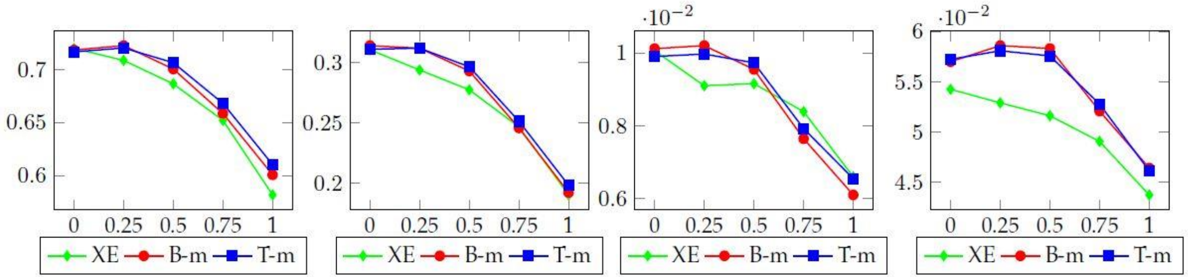
Loss function on RSC15



(a) Recall (b) MRR (c) MAP (d) NDCG

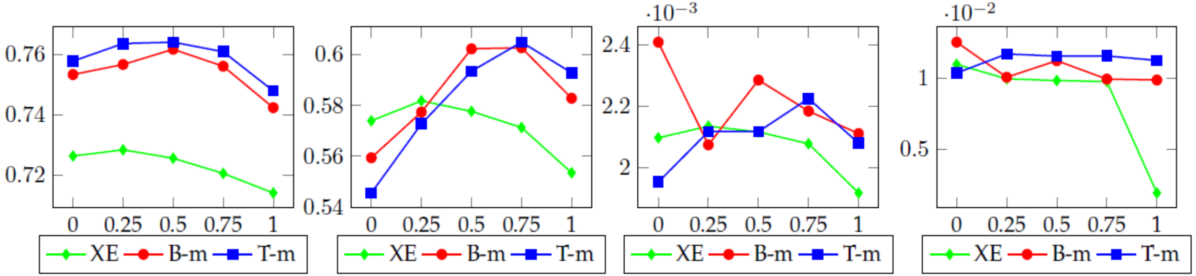
Loss function on RSC19

- BPR-max, TOP1-max, and cross-entropy perform better than those with BPR and TOP1 in terms of all metrics (except NDCG)
- deploy these three loss functions in real-world applications.



(a) Recall@20 (b) MRR@20 (c) MAP@20 (d) NDCG@20

Sample alpha on RSC15



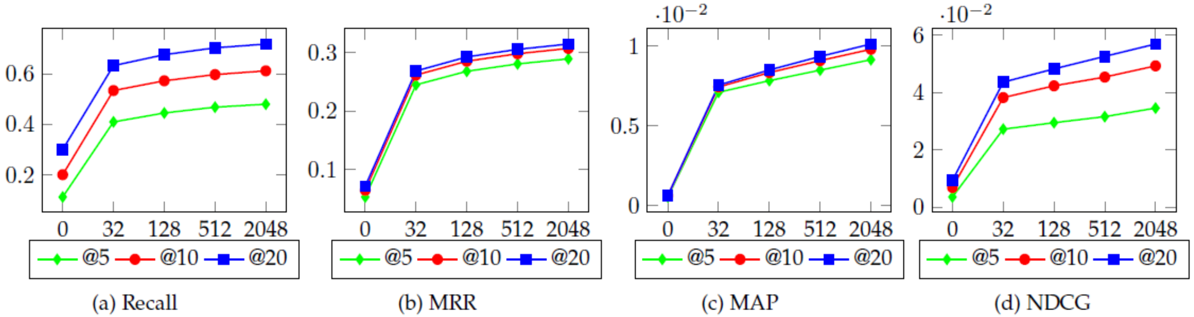
(a) Recall@20 (b) MRR@20 (c) MAP@20 (d) NDCG@20

Sample alpha on RSC19

- Alpha represents the proportion of samples from popularity-based sampling method
- Alpha has a great impact on model performance
- The results on different datasets are varied

Influential factors

Experiment results



Sample size on RSC15

- the larger the negative sample size is, the better the basic model performs regarding all evaluation measurements.
- additional negative sampling leads to higher computing costs

Influential factors

Experiment results

Influential factors

Experiment results

- Try all possible side information (such as texts and images), and carefully design the corresponding modules

- Well consider the connections between other behavior types with the target behavior.
- be careful about the possible noisy information.

- incorporate data argumentation.
- TOP1-max, BPR-max and cross-entroy loss functions for training
- keep a balance between model performance and size of negative samples

- Incorporating with attention mechanism
- combing with traditional sequential learning
- well explicit user representation.

Open Issues and Future Directions

- Objective and comprehensive evaluations across different models

  The baselines used in each paper are different. Lack of a reasonable and unified baseline for sequential recommendation

- More designs on embedding methods

  It is challenging to pre-train an embedding model as the information is constantly changing
  The incorporation of embedding vectors in existing sequential recommendation models are also in a relatively simple way.

- Advanced sampling strategies

  Most existing works use the sampling strategies of uniform, popularity-based, or their straightforward combination (i.e., additional sampling), which are comparatively simple contrasting with the ones used in NLP.

Open Issues and Future Directions

- Better modeling user long-term preference

  The module in DL-based models for user representation (especially the long-term preference) is still far from satisfactory, compared to the designed modules for item representation.

- Personalized recommendation based on polymorphic behavior trajectory.

  There is relatively few studies that well distinguish the behavior types and model their connections in sequential recommendation.
  Our empirical evaluation also indicates that well considering another behavior type for a target type is very challenging.

Open Issues and Future Directions

- ## Learning behavior sequences in real time

  Recommendation systems are expected to ideally capture user interest transfer and timely justify the recommendation strategies.
  Reinforcement Learning(RL) is suitable for this.

- ## Sequential recommendation for specific domains

  Future research can be conducted to design specific models for particular areas by capturing the characteristics of these areas, which is more useful for real-world applications.

Q&A

arXiv link:https://arxiv.org/abs/1905.01997