TIME-SENSITIVE COLLABORATIVE INTEREST AWARE MODEL FOR SESSION-BASED RECOMMENDATION

Yang Lv†, Liansheng Zhuang*†,Pengyu Luo‡,Houqiang Li*,Zhengjun Zha**

*University of Science and Technology of China,
†Peng Cheng Laboratory,‡Hefei University of Technology
lvyang19@mail.ustc.edu.cn, lszhuang@ustc.edu.cn

ABSTRACT

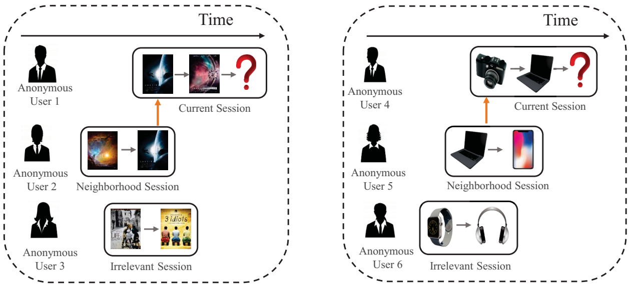
Session-based recommendation, which aims at predicting user's action based on anonymous sessions, is a challenging problem due to the uncertainty of user's behaviors and limited clicked information. Existing methods model users' interests to relieve the uncertainty of user's behavior prediction. However, most methods mainly focus on the current session, ignoring collaborative information (i.e., collaborative interest) in neighborhood sessions with similar interests. We argue that relying on limited implicit feedbacks within a session is insufficient to precisely infer user's interest, especially in the absence of user's profiles and historical behaviors. This paper proposes a novel model called Time-Sensitive Collaborative Interest Aware (TSCIA) to tackle this problem. It explicitly aggregates similar interests from neighborhood sessions to model the general collaborative interest, and simultaneously takes users' interest drifts into account. Finally, both current session and collaborative information are used for next-item prediction. Extensive experiments on public datasets demonstrate the effectiveness of our model.

***Index Terms*—** Recommender Systems, Session-based Recommendation, Nearest Neighbors, Neural Network, Collaborative Filtering

1. INTRODUCTION

Recommender Systems (RS) are critical for online users to alleviate information overload. In many common situations, user's profiles or past interactions are not available for recommendation systems, because some users are anonymous/first-time visitors or the online platform only tracks the identifier of session [1]. To address this problem, session-based recommendation is proposed. The task is defined as predicting the next item relying on limited interactions in the current session, while general recommendation methods can utilize user's profiles and long-term interactions [1, 2].

This work was supported in part to Dr. Liansheng Zhuang by NSFC under contract No.61976199, and in part to Dr. Houqiang Li by NSFC under contract No.61836011. Dr. Liansheng Zhuang is the corresponding author.



(a) Example 1 (b) Example 2

Fig. 1. Impact of neighborhood sessions. Example 1 shows a neighborhood session provides general collaborative information, e.g., other science fiction movies. Example 2 shows a neighborhood session provides popular and seasonal information, which may reflect the user's interest drift with time, e.g., the latest version of the iPhone.

The key challenges of session-based recommendation are the uncertainty of the user's behaviors and limited clicked information, making it difficult to precisely predict the user's next action in the current session. To address the challenges, existing methods have highlighted the importance of modeling user's sequential behavior patterns and capturing user's interest of current session to relieve the uncertainty of user's behavior prediction [1, 2, 3, 4]. For instance, Hidasi et al. [1] apply Gated Recurrent Unit(GRU) [5] to model sequential behavior patterns within the session, and propose the GRU4Rec model. Li et al. [2] improve GRU4Rec by capturing user's interest within the session (i.e., main intention) additionally. Liu et al. [3] propose a hybrid model, which learns both the user's short-term and long-term interest within the session.

Although achieving encouraging progress, existing methods mainly exploit a user's own information from the current session, without considering the collaborative information from neighborhood sessions that display similar behavior patterns and reflect similar user interests as the current session. We argue that relying on several implicit feedbacks is insufficient to precisely infer user's interest, especially in the

978-1-7281-1331-9/20/$31.00 ©2020 IEEE

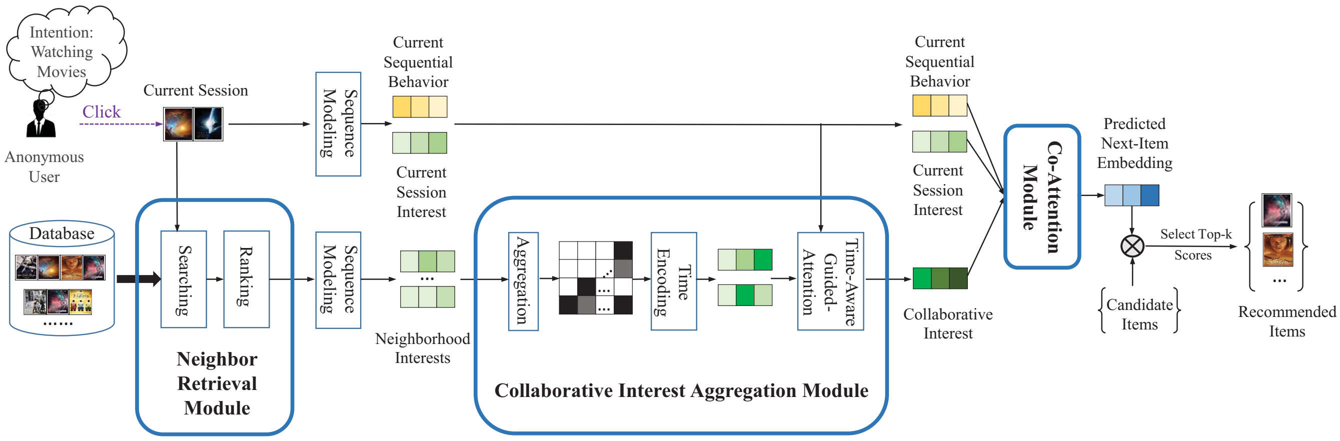


Fig. 2. Pipeline of our proposed model

absence of user's profiles and historical behaviors. Considering people with similar interests tend to have similar behavior patterns, it is desirable to exploit the potential of collaborative information (i.e., collaborative interest) in neighborhood sessions, to help infer the user's interest and improve the recommendation in the current session. Motivated by this observation, some work try to exploit the collaborative information in session-based recommendations [6, 7]. For example, Wang et al. [6] propose a hybrid framework called Collaborative Session-based Recommendation Machine (CSRM) to apply collaborative neighborhood information from memory network to session-based recommendations, and achieve impressive performances on public benchmarks. Though these methods exploit the collaborative sessions, they ignore taking the effect of user's interest drifts into account. In fact, user's interest is dynamic over time, and is easily affected by seasonal and popular factors, which causes some items to be time-sensitive and clicked multiple times in a short period of time in the online platform. Conceptually, recent neighborhood sessions should have more similar interests to current session than those generated a long time ago. Figure 1 illustrates the impact of neighborhood sessions by two practical cases. Therefore, modeling user's interest drifts is important to improve the performance of recommendation systems.

Inspired by the above insights, this paper proposes a novel model, namely Time-Sensitive Collaborative Interest Aware (TSCIA) for session-based recommendation. Our key idea is to exploit the collaborative information (i.e., collaborative interest) from neighborhood sessions and simultaneously model user's interest drifts. The collaborative information complements the user's interest in current session, which can help to relieve the uncertainty of the user's behavior prediction. In particular, user's interest drift is used to refine the weights of different neighborhood sessions, where these weights reflect the similarities of users' interests between neighborhood sessions and current session. Experiments on public benchmarks show our model can predict the user's next behavior more precisely. Our main contributions are listed as follows:

- A novel TSCIA model is proposed for session-based recommendation, which simultaneously exploits both the general collaborative information and the user's interest drifts. To our knowledge, this is the first effort to incorporate the user's interest drifts into the collaborative filtering when constructing a neural network model for session-based recommendations.
- A novel aggregation module is proposed to explicitly exploit collaborative interests from neighborhood sessions, by considering both the clicked items and the occurrence time of sessions.
- Extensive experiments are conducted on two real-world datasets. Experimental results show that our proposed model achieves the state-of-the-art performance.

2. METHODS

Let $V = \{v_1, v_2, ..., v_m\}$ denotes a set of all unique items that appear in all sessions, and we call it item dictionary. Let $s = \{v_{s,1}, v_{s,2}, ..., v_{s,t}\}$ denotes session s, where $v_{s,t} \in V$ denotes a item being clicked at timestamp t in session s. Our model can generate a ranking list over the item dictionary, and calculate the predicted probability for each item, i.e., $\hat{y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_m\}$, where $\hat{y}_i$ denotes the prediction score for the i-th item in V. Finally, the top-k items in $\hat{y}$ are recommended.

We propose a Time-Sensitive Collaborative Interest Aware (TSCIA) model for session-based recommendation. Figure 2 illustrates the pipeline of our proposed model (TSCIA1). Our model mainly consists of three modules: a neighbor retrieval module, a collaborative interest aggregation module, and a co-attention module. Sequence modeling method is used additionally as a feature extractor to learn sequential characteristics from sessions. Firstly, we introduce an efficient neighbor retrieval module to find out k most similar sessions for the current session as neighborhood sessions. After that, we use sequence modeling method to learn sequential characteristics from current session and neighborhood sessions. Secondly, the collaborative interest aggregation module is proposed to explicitly capture the collaborative interest from neighborhood sessions, in which sequential characteristics and the temporal recency (i.e., nearness of sessions in time) are both considered. Finally, a co-attention module is used to selectively combine current session characteristics and collaborative interest for next-item prediction.

2.1. Neighbor Retrieval Module

As shown in Figure 2, the first module is neighbor retrieval module, which provides the foundation for learning dynamic collaborative interest for the current session. The whole process is finished in two steps.

Step1: Searching similar sessions. Following Session-KNN [7], we focus on the whole session and find out all sessions that interact with the items existing in the current session as its neighbors. The process is implemented by two hash tables, which is highly efficient.

Step2: Ranking and selecting the k most similar neighbors. Technically, we first choose an appropriate similarity measure, e.g., the cosine similarity. To be specific, each session is represented as a binary vector in the m-dimensional space of items(value of 1 for the n-th dimension means the n-th item in item dictionary is in this session). Cosine similarity between target session s and a neighbor s_j is given as follow:

$$sim\left(s, s_j\right) = \frac{\vec{s} \cdot \vec{s}_j}{\sqrt{l\left(s\right) \cdot l\left(s_j\right)}}. \tag{1}$$

Now, given the current session s, its whole neighbors can be found and we choose the k most similar sessions as neighborhood sessions N_s .

2.2. Sequence Modeling

We proceed to present two sequential modeling methods by applying our proposed model. Two models are used as examples, i.e., NARM [2] and STAMP [3]. Both NARM and STAMP model the current session interest (i.e., interest of the entire current session). Moreover, NARM emphasizes the current sequential behavior of the entire current session, while STAMP highlights the importance of the latest clicked item within the current session.

NARM consists of a global encoder to model the current sequential behavior (i.e., c_t^{seq}) and a local encoder to capture the current session interest (i.e., $c_t^{interest}$). In detail, firstly, the RNN layer converts the items of input session into high-dimension hidden representations. Secondly, the current session interest is calculated by the weighted sum of all hidden states of current session, and the current sequential behavior is calculated by the last hidden state of current session.

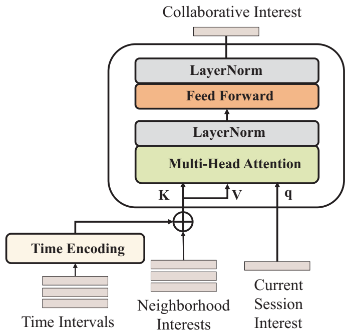


Fig. 3. Time-aware guided-attention

STAMP captures current session interest(i.e., $c_t^{interest}$) and latest interest(i.e., c_t^{latest}) simultaneously. One major difference is that STAMP abandons the RNN structure. The current session interest is generated by an attention mechanism over all items in current session, and latest interest is simply generated by the session's last clicked item.

Apparently, both the NARM and STAMP model user's interest by the current session, which is obviously insufficient. Thus, NARM and STAMP are used as a feature extractor to capture the features from sessions. We use TSCIA1 and TSCIA2 to denote TSCIA-NARM and TSCIA-STAMP respectively. In TSCIA1, we apply NARM to generate $c_t^{interest}$ and c_t^{seq} for the current session. In TSCIA2, we apply STAMP to generate $c_t^{interest}$ and c_t^{latest} for the current session. At the same time, we also apply NARM/STAMP to generate neighborhood interests from neighborhood sessions, note that we regard $c_t^{interest}$ generated from neighborhood sessions as neighborhood interests.

2.3. Collaborative Interest Aggregation Module

After generating neighborhood interests separately, in order to model general collaborative information and user's interest drifts, we propose a collaborative interest aggregation module to extract closely related interests from neighborhood sessions. First, neighborhood sessions that are less than k are padded with zeros and we use masks to denote them. Then, time-aware guided-attention mechanism is applied motivated by Transformer [8], as detailed in Figure 3.

Considering given a query $q \in \mathbb{R}^{1\times d}$, a key matrix $K \in \mathbb{R}^{n\times d}$ and a value matrix $V \in \mathbb{R}^{n\times d}$. The attended feature $f \in \mathbb{R}^{1\times d}$ is the weighted sum over value V based on the attention weights.

$$f = Attention(q, K, V) = softmax(\frac{qK^T}{\sqrt{d}})V. \tag{2}$$

We regard the current sequential behavior(for TSCIA1) or latest interest(for TSCIA2) of current session as query, and regard the neighborhood interests as key and value. Thus, the generated collaborative interest for the current session can be understood as reconstructing it by all neighborhood sessions with respect to their similarities to current session, i.e., generating similar interests in neighborhood sessions based on the sequential characteristics of current session.

In order to improve the capability of attention mechanism, multi-head attention is introduced to jointly pay attention to information from different representation subspaces.

$$\begin{aligned} &MultiHead(q,K,V) = Concat(head_1, ..., head_h)W^O \\ &where \quad head_i = Attention(qW_i^q, KW_i^K, VW_i^V). \end{aligned} \tag{3}$$

Notice that the projection matrices $W_i^q \in \mathbb{R}^{d\times d_h}, W_i^K \in \mathbb{R}^{d\times d_h}, W_i^V \in \mathbb{R}^{d\times d_h}$ and $W^O \in \mathbb{R}^{h*d_h\times d}$.

After that, layer normalization and feed-forward layer are also applied for improving the capability of attention.

Actually, temporal recency has been shown to be of great importance [9]. The occurrences of items do not obey the assumption of independent and identical distribution(iid assumption), an item will only appear in a session when it is released online [10], recent sessions contain recent popular items. Besides, user's interest often drifts with time, and is easily affected by seasonal and popular factors. As a result, some items are time-sensitive and tend to be clicked repeatedly during a certain period, such as seasonal fruits and popular products in e-commerce. Therefore, it is inappropriate for the current session to consider all neighborhood interests from different periods as equally significant.

We solve this problem by injecting the time intervals between current session and neighborhood sessions into the input embedding of guided-attention. To be specific, we encode the relative time distance into a vector of the same dimension as the neighborhood interests so that they can be added, as detailed in Figure 3. Thus, the model considers both the sequential characteristics and the temporal recency when determining the importance of each neighborhood session.

We introduce a time interval feature $\delta_i(t)$ as follow:

$$\delta_i(t) = \phi\left\{W_t log\left[t(s) - t(N_s(i)) + 1\right] + b_t\right\}, \tag{4}$$

where $t(s)$ and $t(N_s(i))$ are the occurrence time of current session s and neighborhood session $N_s(i)$ respectively.

Finally, the output of the collaborative interest aggregation module is used as the collaborative interest $c_t^{neighbor}$.

2.4. Co-Attention Module

Both current session characteristics and collaborative interest have strength and weakness. Instead of concatenating them easily, we use an adaptive method for information fusion. We apply a co-attention module for current session characteristics and collaborative interest to determine which part should play a more important role.

Taking TSCIA1 as an example, the final predicted item embedding is computed as:

$$c_t = \left[c_t^{seq}, c_t^{interest}\right] W_{lg} + \alpha c_t^{neighbor} W_n, \tag{5}$$

The coefficient α is computed as:

$$\alpha = \sigma\left(W_{\alpha l} c_t^{seq} + W_{\alpha g} c_t^{interest} + W_{\alpha n} c_t^{neighbor} + b\right), \tag{6}$$

where $W_{lg} \in \mathbb{R}^{2d\times d}, W_n \in \mathbb{R}^{d\times d}, W_{\alpha l}, W_{\alpha g}, W_{\alpha n} \in \mathbb{R}^{1\times d}$.

2.5. Prediction Layer

We generate the final prediction scores by calculating dot product of each candidate item embedding and the final predicted item embedding c_t. Let V_i be the i-th item in the item dictionary, the score of V_i is computed as $\hat{z}_i = V_i^T c_t$. The objective function is defined as the cross-entropy of the prediction and the ground truth,

$$\mathcal{L}(\hat{y}) = -\sum_{i=1}^{m} y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i), \tag{7}$$

where $\hat{y} = softmax(\hat{z})$ and y denotes the one-hot vector of the ground truth item.

3. EXPERIMENTAL SETUP

3.1. Experimental Settings

Datasets. We conduct all the experiments on two real-world datasets, i.e., *Diginetica*[1] and *Retailrocket*[2]. Diginetica comes from CIKM Cup 2016. Retailrocket comes from an e-commerce company, which contains six months of user browsing actions. Following [11], we manually divide the action history into sessions through a 30-minute interval for Retailrocket. Following [2, 3], we filter out sessions of length 1 and items that appear less than 5 times or only appear in testing set, and apply data augmentation technique for both datasets. Testing set consists of sessions from the subsequent week. Statistics of these datasets are shown in Table 1.

Baselines. We compare our model with a series of baselines, including conventional methods and recent state-of-the-art neural session-based recommendation models. These are: Pop [1], Session-Pop [1], Item-KNN [12], Session-KNN [7], GRU4Rec [1], NARM [2], STAMP [3], CSRM [6].

Evaluation Metric. We apply two evaluation metrics, i.e., Hit Rate(HR@K) and Mean Peciprocal Rank (MRR@K). HR@K indicates the proportion of test samples with the correct recommended items in the top-k position of the ranking list. MRR@K is the average of reciprocal ranks of the correct item in the top-k position of the ranking list.

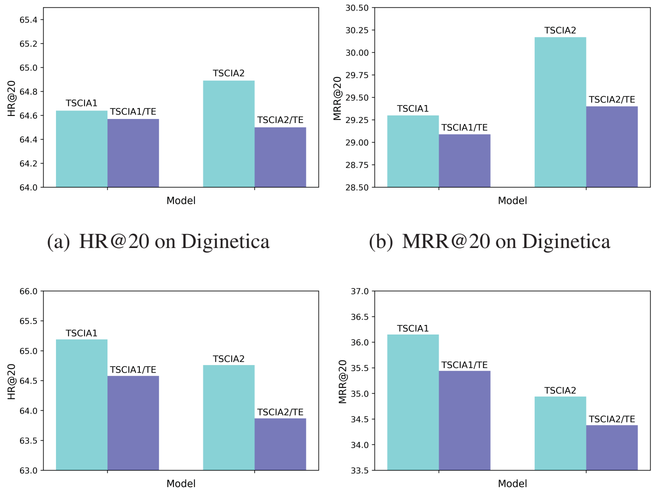
[1] http://cikm2016.cs.iupui.edu/cikm-cup
[2] https://www.kaggle.com/retailrocket/ecommerce-dataset

Table 1. Statistics of the experiment datasets

Dataset	#train	#test	clicks	items	avg.len
Diginetica	719470	60858	982961	43097	5.12
Retailrocket	264453	35762	413648	24095	6.68

Table 2. Performance comparison

Dataset	Diginetica		Retailrocket	
Measures	HR@20	MRR@20	HR@20	MRR@20
Pop	0.96	0.24	1.24	0.32
Session-Pop	21.11	14.60	40.48	32.04
IKNN	28.93	9.48	30.59	13.30
SKNN	49.79	18.59	61.78	34.39
GRU4Rec	57.95	24.93	49.65	21.60
NARM	62.58	27.35	61.79	34.07
STAMP	62.03	27.38	61.08	33.10
CSRM	63.07	27.45	63.64	34.76
TSCIA1	64.64	29.30	**65.19**	**36.15**
TSCIA2	**64.89**	**30.17**	64.76	34.94



(a) HR@20 on Diginetica (b) MRR@20 on Diginetica

(c) HR@20 on Retailrocket (d) MRR@20 on Retailrocket

Fig. 4. The performance of TSCIA and TSCIA/TE

Parameters. We use 10% of the training set as validation set for adjustment of hyperparameters for models that contain hyperparameters. We report the best models which are selected by early stopping based on the HR@20 score on the validation set. Notice that the validation set does not participate in training. According to the validation set, we use the following hyperparameters: item embedding dimension: 100, initial learning rate: 0.0005, learning rate decay: 0.9, batch-size: 256, epoch: 30. The number of neighborhood sessions is selected in $\{10, 20, 30, 40, 50\}$, and finally we set it to 30.

4. RESULTS AND ANALYSIS

We compare our TSCIA model to all baselines and the experimental results are shown in Table 2.

We have the following observations:

1) We observe that our proposed TSCIA achieves state-of-the-art performance. Our best model outperforms the second-best model by 2.89%, 2.44% on HR@20 and 9.91%, 4.00% on MRR@20 in two datasets respectively. As for TSCIA and CSRM, the improvement indicates that explicitly taking the user's interest drifts into account is valuable. Besides, our model finds neighborhood sessions from the whole training set with the help of k-Nearest-Neighbor, while CSRM exploits collaborative information from memory network that remembers a small number of recent sessions. The results also show that our model can precisely retrieve neighborhood sessions, since our TSCIA/TE (defined in the following content) model also outperforms CSRM.

2) In order to verify the performance of TSCIA in more realistic scenarios, where the recommendation system only recommends a few items at once because viewers are usually impatient. We additionally test the performance on HR@10, HR@5, MRR@10 and MRR@5, and the experimental results are summarized in Table 3. It can be observed that TSCIA still retains certain advantages, which indicates TSCIA tends to make more precise recommendations.

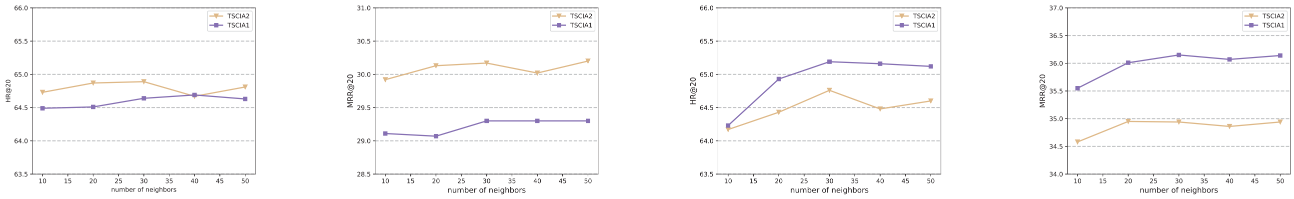
3) We observe Session-KNN outperforms Item-KNN. A possible reason is that Session-KNN makes full use of each item in the current session while Item-KNN only utilizes the last item, which is obviously insufficient. Although Session-KNN utilizes the entire session and considers collaborative filtering by nearest neighbors, it neglects the sequential order within the session, however, which is solved in our model.

Effects of collaborative interest. To illustrate the effectiveness of collaborative interest, we compare NARM with TSCIA1, as well as STAMP with TSCIA2. The main difference is that TSCIA1/TSCIA2 considers collaborative interest as complementary information in addition to current session modeling(NARM/STAMP). Finally, TSCIA1/TSCIA2 obtains obvious improvements over NARM/STAMP, as reported in Table 2. The results prove that considering current session alone is insufficient for next-item prediction. It is necessary to involve collaborative interest since similar interests tend to click on similar items.

Effects of user's interest drifts. We design two models to verify the validity of considering user's interest drifts, i.e., 1). The TSCIA model proposed in the paper, which applies time encoding by considering the time intervals when determining the importance of each neighborhood session. 2). The TSCIA/TE model, refering to TSCIA without time encoding. The experimental results are reported in Figure 4, and we use blue and purple to denote TSCIA and TSCIA/TE respectively. We observe TSCIA outperforms TSCIA/TE, proving the significance of time encoding. The primary reason is that user's interest often drifts with time, it is obvious that recent neighborhood sessions contain more similar interests than those a

Table 3. The results of HR@K,MRR@K when K=5,10

Dataset	Diginetica				Retailrocket			
Measures	HR@10	MRR@10	HR@5	MRR@5	HR@10	MRR@10	HR@5	MRR@5
STAMP	52.07	26.69	41.04	25.21	53.26	32. 57	44.82	31.44
NARM	51.91	26.53	40.67	25.02	54.28	33.54	46.09	32.44
CSRM	52.54	26.98	41.27	25.46	55.86	34.21	46.99	33.03
TSCIA1	54.14	28.57	43.07	27.08	**57.23**	**35.59**	**48.66**	**34.44**
TSCIA2	**54.67**	**29.45**	**43.85**	**28.00**	56.67	34.37	47.54	33.14



(a) HR@20 on Diginetica (b) MRR@20 on Diginetica (c) HR@20 on Retailrocket (d) MRR@20 on Retailrocket

Fig. 5. The performance with different number of neighborhood sessions

long time ago. Our model explicitly considers the user's interest drifts, which helps to refine the importance of each neighborhood session. In this way, our model can precisely infer the user's interest of current session, and furthermore relieve the uncertainty of user's next behavior prediction.

Effects of number of neighborhood sessions. In order to further observe the influence of collaborative interest, we show the performance with respect to the number of neighborhood sessions(set k to 10, 20, 30, 40, 50 respectively) in Figure 5. Overall, the performance is relatively poor when considering less neighbors, and tends to be stable but changes slightly as the number of neighborhood sessions increases, which reveals that the model cannot contain sufficient collaborative information to predict the next item when considering less neighborhood sessions. In order to balance the performance and computational efficiency, we set k to 30.

5. CONCLUSION

In this paper, a novel model named TSCIA for session-based recommendation is proposed. By incorporating a neighbor retrieval module and a collaborative interest aggregation module, general collaborative interest and user's interest drifts are both considered to complement the user's interest in the current session, which aims to help relieve the uncertainty of user's behavior prediction. Finally, the co-attention module takes both the current session and collaborative interest into account for next-item prediction. Extensive experimental results and qualitative experimental analyses have shown the effectiveness and rationality of our proposed model.

6. REFERENCES

[1] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk, "Session-based recommendations with recurrent neural networks," in *ICLR*, 2016.

[2] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma, "Neural attentive session-based recommendation," in *CIKM*, 2017, pp. 1419–1428.

[3] Qiao Liu, Yifu Zeng, et al., "Stamp: short-term attention/memory priority model for session-based recommendation," in *KDD*, 2018, pp. 1831–1839.

[4] Shu Wu, Yuyuan Tang, Yanqiao Zhu, et al., "Session-based recommendation with graph neural networks," in *AAAI*, 2019, vol. 33, pp. 346–353.

[5] Kyunghyun Cho et al., "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *EMNLP*, 2014, pp. 1724–1734.

[6] Meirui Wang, Pengjie Ren, Lei Mei, et al., "A collaborative session-based recommendation approach with parallel memory modules," in *SIGIR*, 2019.

[7] Dietmar Jannach and Malte Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *RecSys*, 2017, pp. 306–310.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Jones, et al., "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[9] Diksha Garg, Priyanka Gupta, et al., "Sequence and time aware neighborhood for session-based recommendations: Stan," in *SIGIR*, 2019, pp. 1069–1072.

[10] Yong Kiam Tan, Xinxing Xu, and Yong Liu, "Improved recurrent neural networks for session-based recommendations," in *DLRS*, 2016, pp. 17–22.

[11] Malte Ludewig and Dietmar Jannach, "Evaluation of session-based recommendation algorithms," *User Modeling and User-Adapted Interaction*, vol. 28, no. 4-5, pp. 331–390, 2018.

[12] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, et al., "The youtube video recommendation system," in *RecSys*, 2010, pp. 293–296.