An Intent-guided Collaborative Machine for Session-based Recommendation

Pan, Z.; Cai, F.; Ling, Y.; de Rijke, M.

DOI
10.1145/3397271.3401273

Publication date
2020

Document Version
Author accepted manuscript

Published in
SIGIR '20

An Intent-guided Collaborative Machine for Session-based Recommendation

Zhiqiang Pan[1] Fei Cai[1]* Yanxiang Ling[1] Maarten de Rijke[2,3]
[1]Science and Technology on Information Systems Engineering Laboratory,
National University of Defense Technology, Changsha, China
[2]University of Amsterdam, Amsterdam, The Netherlands
[3]Ahold Delhaize, Zaandam, The Netherlands
{panzhiqiang, caifei, lingyanxiang}@nudt.edu.cn, derijke@uva.nl

ABSTRACT

Session-based recommendation produces item predictions mainly based on anonymous sessions. Previous studies have leveraged collaborative information from neighbor sessions to boost the recommendation accuracy for a given ongoing session. Previous work often selects the most recent sessions as candidate neighbors, thereby failing to identify the most related neighbors to obtain an effective neighbor representation. In addition, few existing methods simultaneously consider the sequential signal and the most recent interest in an ongoing session. In this paper, we introduce an *Intent-guided Collaborative Machine for Session-based Recommendation* (ICM-SR). ICM-SR encodes an ongoing session by leveraging the prior sequential items and the last item to generate an accurate session representation, which is then used to produce initial item predictions as intent. After that, we design an intent-guided neighbor detector to locate the correct neighbor sessions. Finally, the representations of the current session and the neighbor sessions are adaptively combined by a gated fusion layer to produce the final item recommendations. Experiments conducted on two public benchmark datasets show that ICM-SR achieves a significant improvement in terms of Recall and MRR over the state-of-the-art baselines.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Session-based recommendation, Memory network, Intent-guided neighbor.

ACM Reference Format:
Zhiqiang Pan, Fei Cai, Yanxiang Ling and Maarten de Rijke. 2020. An Intent-guided Collaborative Machine for Session-based Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401273

*Corresponding author.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401273

1 INTRODUCTION

Recommender systems (RS) can help predict a user's personalized information needs according to his historical interactions [5, 12]. However, historical interactions are not always available in real-world scenarios [2], e.g., with new or anonymous users. In a session-based RS, the interactions of each user are sequentially organized into sessions according to their temporal order, and recommendations are produced solely based on the ongoing session. Thus, how to leverage an ongoing session in order to detect a user's current intent is the key to successful session-based recommendation.

Hidasi et al. [2] first apply an RNN structure to model the sequential items existing in an ongoing session. Li et al. [6] present an attention-augmented RNN framework aiming to capture a user's main purpose. Wu et al. [11] utilize a gated graph neural network to fully mine the transition relationship between items in an ongoing session and regard the last item as the user's recent interest. The work listed above generally fails to simultaneously consider the sequential signal in the session and the importance of the last item, which indicate the user's global preference and recent interest, respectively. In order to better exploit sessions, some studies have tried to extract similar sessions from other users as so-called neighbor sessions. For instance, Jannach and Ludewig [3] employ a K-nearest neighbor method to identify the neighbor sessions. Wang et al. [10] introduce a memory encoder to exploit collaborative information from neighbor sessions. A major drawback of such neighbor-based models is that the neighbor sessions are simply collected and identified by using cosine similarity without considering the user's intent.

To address the issues listed above, we propose an *intent-guided collaborative machine for session-based recommendation* (**ICM-SR**), that consists of three major components, an intent generator, an intent-guided neighbor detector, and a preference fusion layer. A session encoder is used to take the current session as input and applies a GRU to produce a representation of sequential items as global preference, where the embedding of the last item is regarded as the user's recent interest. Then, the output of the session encoder is utilized to generate an initial item predictions, where the items with the top scores are selected as potential intents of the current session. Then, an intent-guided neighbor detector is implemented following a two-stage process: first, the generated intent is used to identify the candidate neighbor sessions; after that, we generate a neighbor representation by adaptively combining the representation of neighbor sessions according to their similarity to the current session. Finally, the preference fusion layer integrates

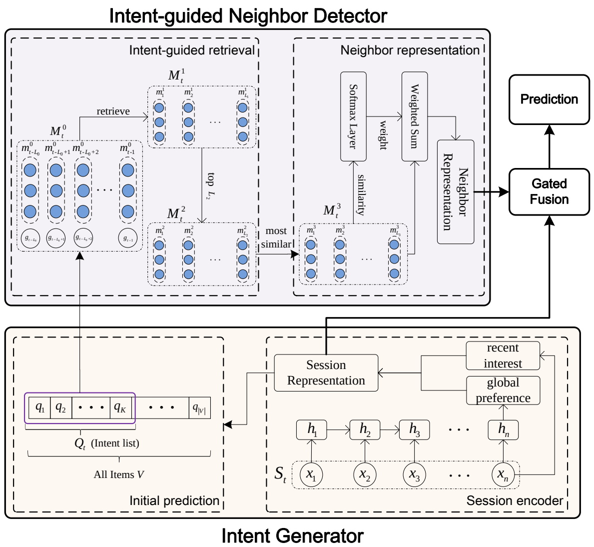


Figure 1: Overview of ICM-SR framework.

the representations of the current session as well as the neighbor sessions for producing item recommendations.

The main contributions of our work can be summarized as follows: (1) to the best of our knowledge, we are the first to consider a user's potential intent to retrieve the correct neighbor sessions for neighbor representation; (2) we propose an encoder for session representation that can simultaneously consider a user's general preference and his recent interest in the session; and (3) extensive experiments are conducted on two public benchmark datasets, validating that our proposal outperforms the state-of-the-art baselines in terms of Recall and MRR.

2 APPROACH

Given a session at timestamp t as $S_t = \{s_1, s_2, \ldots, s_n\}$ consisting of n items, where s_τ ($1 \leq \tau \leq n$) denotes an item in the session, the *session-based recommendation* aims to predict the next item s_{n+1} from an item set $V = \{v_1, v_2, \ldots, v_{|V|}\}$ that the user may interact with. Fig. 1 shows an overview of our proposed ICM-SR model, with three main components, an intent generator, an intent-guided neighbor detector, and a preference fusion layer.

2.1 Intent generator

Given a session $S_t = \{s_1, s_2, \ldots, s_n\}$, we first embed each item s_τ into a d dimensional representation $x_\tau \in \mathbb{R}^d$. Then, in order to capture the sequential signal contained in S_t, we apply a gated recurrent unit (GRU) to the item embeddings $X_t = \{x_1, x_2, \ldots, x_n\}$:

$$h_\tau = \text{GRU}(x_\tau, h_{\tau-1}), \tag{1}$$

where h_τ is the hidden state of item s_τ. We use the last hidden state h_n to represent the global preference z_t^{global} of session S_t as:

$$z_t^{global} = h_n. \tag{2}$$

Considering that the last item s_n reflects the user's latest interaction, we directly adopt its embedding x_n to represent the user's recent interest z_t^{recent} in the session, i.e., $z_t^{recent} = x_n$.

To fully consider the user's global preference as well as his recent interest, we finally model the current session $z_t^{current}$ by concatenating z_t^{global} and z_t^{recent} as:

$$z_t^{current} = W_0[z_t^{global}; z_t^{recent}], \tag{3}$$

where $[\cdot]$ denotes a concatenating operation and $W_0 \in \mathbb{R}^{d \times 2d}$ is used for the linear projection.

After that, we use $z_t^{current}$ to produce an initial prediction score corresponding to each item in V as follows:

$$\hat{y}_t^1 = \text{softmax}(z_t^{current\top} X), \tag{4}$$

where X are the embeddings of the candidate items in V and $\hat{y}_t^1 \in \mathbb{R}^{|V|}$ corresponds to the predicted scores of each item $v_i \in V$. Then we select the items with top-K scores in $\hat{y}_t^1$ as the intent Q_t of session S_t, i.e., $Q_t = \{q_1, \ldots, q_K\}$, where q_i is a selected item.

2.2 Intent-guided neighbor detector

In a given session the iterative process of presenting items for the user to interact with repeated until the user's demand is satisfied [6]. Hence, the target item in the session can implicitly represent user's intent. Hence, given a session S_t, we sequentially collect the representation of its preceding L_0 sessions and their corresponding target items to construct the session memory M_t^0, i.e., $M_t^0 = \{(m_{t-L_0}^0, g_{t-L_0}), \ldots, (m_{t-1}^0, g_{t-1})\}$, where m_i^0 and g_i (t-$L_0 \leq i \leq t$-1) denote the representation and the target item of session S_i, respectively. The session memory is updated by a first-in-first-out mechanism [10], ensuring to accommodate the latest L_0 sessions before the current session. Then, we try to select a subset from M_t^0 to construct the neighbor representation, which is implemented as a two-stage process, namely *intent-guided retrieval* and *neighbor representation*.

For intent-guided retrieval, given the session memory M_t^0, we retrieve the representations of sessions whose target item occurs in $Q_t = \{q_1, \ldots, q_K\}$. This can be formalized as:

$$M_t^1 = \{m_i^0 \mid g_i = q_k, t - L_0 \leq i \leq t-1, 1 \leq k \leq K\}. \tag{5}$$

Then we reshape M_t^1 into $\{m_1^1, m_2^1, \ldots, m_{L_1}^1\}$ according to the item order in Q_t, where L_1 is the number of retrieved session. After that, we select the top L_2 sessions from M_t^1 as the candidate neighbors of session S_t,[1] which is denoted as $M_t^2 = \{m_1^2, m_2^2, \ldots, m_{L_2}^2\}$.

After getting the candidate neighbors M_t^2, to determine the relevance of each candidate neighbor session to the current session S_t, we compute the cosine similarity of each session representation $m_j^2 \in M_t^2$ ($1 \leq j \leq L_2$) to the current session representation $z_t^{current}$:

$$\text{sim}(z_t^{current}, m_j^2) = \frac{z_t^{current} m_j^2}{\|z_t^{current}\| \times \|m_j^2\|}. \tag{6}$$

We then select the representation of the L_3 most similar sessions as the final neighbors of S_t, denoted as $M_t^3 = \{m_1^3, m_2^3, \ldots, m_{L_3}^3\}$. Then we compute a weighted sum of the neighbors:

$$z_t^{neighbor} = \sum_{r=1}^{L_3} w_r m_r^3, 1 \leq r \leq L_3, \tag{7}$$

[1] If the number of retrieved sessions L_1 is less than L_2, then the remaining part will be padded by the most recent $L_2 - L_1$ session representation in session memory M_t^0.

where m_r^3 is the representation of the r-th neighbor session in M_t^3 and $z_t^{neighbor}$ can be regarded as the neighbor representation of S_t. The weight w_r can be obtained by:

$$w_r = \text{softmax}(\text{sim}(z_t^{current}, m_r^3)). \tag{8}$$

where sim(·) is the similarity that has been calculated from Eq. (6).

2.3 Preference fusion and prediction

After generating the current session representation $z_t^{current}$ from Eq. (3) and its neighbor representation $z_t^{neighbor}$ from Eq. (7), we adopt a gated fusion layer to selectively integrate them to represent a user's preference u_t as follows:

$$u_t = f_t z_t^{current} + (1 - f_t) z_t^{neighbor}, \tag{9}$$

where f_t is calculated by:

$$f_t = \sigma(W_1 z_t^{current} + W_2 z_t^{neighbor}), \tag{10}$$

where σ is the sigmoid activation function and $W_1, W_2 \in \mathbb{R}^{d\times d}$ are trainable parameters.

The final predictions are produced in the same way as Eq. (4):

$$\hat{y}_t^2 = \text{softmax}(u_t^\top X), \tag{11}$$

where X are the embeddings of the candidate items in V.

For training the model, we adopt cross-entropy as the optimization objective to learn the parameters:

$$L(\hat{y}_t^2) = -\sum_{l=1}^{|V|} y_l \log(\hat{y}_l) + (1 - y_l)\log(1 - \hat{y}_l), \tag{12}$$

where y_l and $\hat{y}_l$ are the l-th element of the one-hot encoding vector of the ground-truth and $\hat{y}_t^2$, respectively. That is, $y_l = 1$ if item v_l is the target item of the given session; otherwise, $y_l = 0$. Finally, we apply the Back-Propagation Through Time (BPTT) algorithm to train the proposed ICM-SR model.

3 EXPERIMENTS

Research questions. (RQ1): How does the proposed model ICM-SR perform against competitive baselines? (RQ2): What is the impact of session length on the performance of ICM-SR?

Model summaries. To examine the effectiveness of our proposal, we compare it with nine competitive baselines, including: (1) Three traditional methods, i.e., S-POP [1], Item-KNN [9] and FPMC [8]. (2) Four current session based neural methods, i.e., GRU4REC [2], NARM [6], STAMP [7], SR-GNN [11]. (3) Two neighbor-enhanced neural methods, i.e., RNN-KNN [3] and CSRM [10]. The models we propose in this paper: (1) ICM-SR: the proposed Intent-guided Collaborative Machine; and its variant (2) ICM-SR-NARM: where the session encoder is replaced by NARM [6] as in CSRM [10].

Datasets. We evaluate the methods on two benchmark datasets, i.e., YOOCHOOSE[2] and DIGINETICA.[3] Following [6], we take 1/64 of the whole YOOCHOOSE dataset for experiments. For preprocessing, we follow [6, 7, 11]. For the statistics of the two datasets, please see Table 1.

Implementation details. We adopt Adam as the optimizer, where the initial learning rate is set to 0.001 and the decay is set to 0.1 after every 3 epochs. The batch size and the L2 penalty are set to 100 and 10^{-5}, respectively. The size of the session memory L_1 is set to 10000 as in CSRM [10] and the number of items K is 50 as intent. The number of candidate neighbors L_2 and the final neighbors L_3 are set to 1000 and 100, respectively.

Evaluation metrics. Following [7, 10, 11], we use Recall@N and MRR@N for evaluation; N is set to 20 in our experiments.

[2]http://2015.recsyschallenge.com/challege.html
[3]http://cikm2016.cs.iupui.edu/cikm-cup

Table 1: Statistics of the datasets used in our experiments.

Statistics	YOOCHOOSE	DIGINETICA
# clicks	557,248	982,961
# training sessions	369,859	719,470
# test sessions	55,898	60,858
# items	16,766	43,097
Average session length	6.16	5.12

Table 2: Model performance. The results of the best baseline and the best performer in each column are underlined and boldfaced, respectively. Statistical significance of pairwise differences of ICM-SR against the best baseline (▲) and of ICM-SR against ICM-SR-NARM (△) are determined by a t-test ($p < 0.05$).

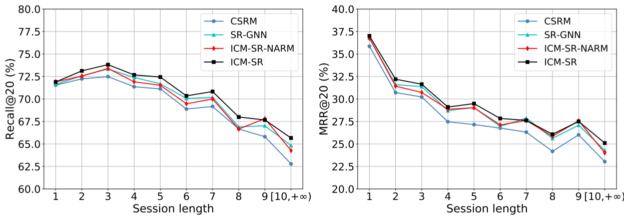
Method	YOOCHOOSE		DIGINETICA	
	Recall@20	MRR@20	Recall@20	MRR@20
S-POP	30.44	18.35	21.06	13.68
Item-KNN	51.60	21.81	35.75	11.57
FPMC	45.62	15.01	31.55	8.92
GRU4REC	60.64	22.89	29.45	8.33
NARM	68.32	28.63	49.70	16.17
STAMP	68.74	29.67	45.64	14.32
SR-GNN	<u>70.57</u>	<u>30.94</u>	50.73	<u>17.59</u>
RNN-KNN	63.77	25.22	48.06	16.95
CSRM	69.85	29.71	<u>51.69</u>	16.92
ICM-SR-NARM	70.52	30.67	52.04	17.48
ICM-SR	**71.11**▲△	**31.23**▲△	**52.32**▲△	**17.74**▲△

4 RESULTS AND DISCUSSION

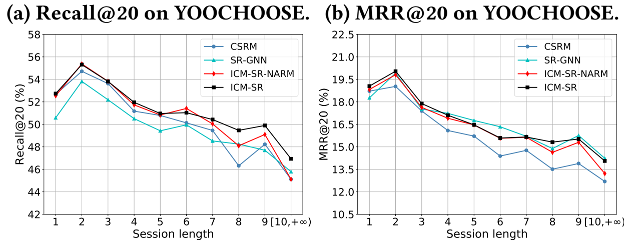
4.1 Overall performance

To answer RQ1, we compare the performance of ICM-SR and its variant ICM-SR-NARM against the baselines in terms of Recall@20 and MRR@20. The results are presented in Table 2.

First of all, we zoom in on the baselines. For the traditional methods, we see that Item-KNN with collaborative information from other items performs best. For the neural methods, we observe that the neighbor-enhanced methods consistently improve the recommendation accuracy. For instance, RNN-KNN beats GRU4REC, where RNN-KNN is the extension of GEU4REC by introducing neighbor sessions using KNN. Different from RNN-KNN, CSRM resorts to the memory to take neighbor sessions into consideration on the basis of NARM, and achieves a clearly better performance. SR-GNN achieves the best performance among all baselines in terms of Recall@20 and MRR@20 on both datasets except for one case, where CSRM performs best in terms of Recall@20 on DIGINETICA.



(a) Recall@20 on YOOCHOOSE. **(b) MRR@20 on YOOCHOOSE.**



(c) Recall@20 on DIGINETICA. **(d) MRR@20 on DIGINETICA.**

Figure 2: Model performance in terms of Recall@20 and MRR@20 with different session lengths.

It could be due to that SR-GNN models the transition relationship with a GGNN, which helps to generate an accurate representation of the current session. Hence, we use CSRM and SR-GNN as our baselines in later experiments.

Next, we move to ICM-SR and ICM-SR-NARM. In general, ICM-SR achieves the best performance in terms of Recall@20 and MRR@20 on both datasets. In addition, we see that ICM-SR outperforms ICM-SR-NARM, which indicates that, in our framework, the proposed session encoder leads to a better session representation than NARM. Compared to CSRM, ICM-SR-NARM, which employs a different neighbor selection strategy, achieves a better performance. This may be attributed to the proposed intent-guided neighbor detector, which can select closely related neighbor sessions to enhance the representation of user's preference. It is noticeable that the improvements of ICM-SR-NARM over CSRM in terms of Recall@20 are 0.96% on YOOCHOOSE and 0.68% on DIGINETICA, while the improvements in terms of MRR@20 are 3.23% and 3.31% on the two datasets, respectively. A higher improvement in terms of MRR@20 indicates that the intent-guided neighbor detector can help return the target items at an earlier position in the recommendation list.

4.2 Impact of session length

To answer RQ2, we compare the performance of ICM-SR, ICM-SR-NARM, and the best baselines, i.e., SR-GNN and CSRM, while varying the session length. The results in terms of Recall@20 and MRR@20 are plotted in Fig. 2.

On the YOOCHOOSE dataset, in general, ICM-SR consistently achieves a better performance than other models with varying session lengths in terms of Recall@20 and MRR@20. With the increase of session length, we can find that the performance of all models in terms of Recall@20 remains stable up to session length 5 and then displays a downward trend, while the performance in terms of MRR@20 shows a downward trend overall. The common downward trends of both metrics on long sessions may be due to the fact that the user will interact with more non-relevant items in longer sessions, either by curiosity or accident.

Interestingly, on the DIGINETICA dataset, all models achieve their best performance for sessions of length 2; after that, they show a downward trend in terms of both Recall@20 and MRR@20. Specifically, the gains of ICM-SR over the state-of-the-art baselines in terms of Recall@20 are more obvious than in terms of MRR@20. In some cases, ICM-SR loses against SR-GNN in terms of MRR@20. In addition, the improvements of ICM-SR over ICM-SR-NARM in terms of both Recall@20 and MRR@20 are especially noticeable for long sessions. This indicates that, by focusing on the recent interest in the current session, ICM-SR can more accurately represent long sessions than ICM-SR-NARM.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an Intent-guided Collaborative Machine for Session-based Recommendation (ICM-SR), which incorporates the current session and collaborative information from neighbor sessions for item recommendation. In particular, we have proposed a session encoder that aims to model both the sequential signal and the recent interest in the session. Moreover, with the proposed intent-guided neighbor detector, ICM-SR is able to capture the user's intent from the current session for detecting correct neighbor sessions as auxiliary information. Experimental results show that ICM-SR can achieve state-of-the-art performance in terms of Recall and MRR. As to future work, we would like to apply the proposed intent-guided detector to other tasks, e.g., conversational recommendation [4].

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under No. 61702526, the Defense Industrial Technology Development Program under No. JCKY2017204B064, and the Innovation Center for AI (ICAI). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. Knowl. Data Eng.* 17, 6 (2005), 734–749.
[2] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, et al. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR'16*.
[3] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *RecSys'17*. 306–310.
[4] Wenqiang Lei, Xiangnan He, Yisong Miao, et al. 2020. Estimation-Action-Reflection: Towards Deep Interaction Between Conversational and Recommender Systems. In *WSDM'20*. 304–312.
[5] Chenliang Li, Cong Quan, Li Peng, et al. 2019. A Capsule Network for Recommendation and Explaining What You Like and Dislike. In *SIGIR'19*. 275–284.
[6] Jing Li, Pengjie Ren, Zhumin Chen, et al. 2017. Neural Attentive Session-based Recommendation. In *CIKM'17*. 1419–1428.
[7] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, et al. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *KDD'18*. 1831–1839.
[8] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *WWW'10*. 811–820.
[9] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, et al. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *WWW'01*. 285–295.
[10] Meirui Wang, Pengjie Ren, Lei Mei, et al. 2019. A Collaborative Session-based Recommendation Approach with Parallel Memory Modules. In *SIGIR'19*. 345–354.
[11] Shu Wu, Yuyuan Tang, Yanqiao Zhu, et al. 2019. Session-Based Recommendation with Graph Neural Networks. In *AAAI'19*. 346–353.
[12] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, et al. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *WSDM'19*. 582–590.