Information Processing and Management 59 (2022) 102936

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Dynamic intent-aware iterative denoising network for session-based recommendation

Xiaokun Zhang [a], Hongfei Lin [a], Bo Xu [a,*], Chenliang Li [b], Yuan Lin [a], Haifeng Liu [a], Fenglong Ma [c]

[a] Dalian University of Technology, Dalian, China
[b] Wuhan University, Wuhan, China
[c] Pennsylvania State University, PA, USA

ARTICLE INFO

Keywords:
Session-based recommendation
Dynamic intents
Uncertain behavior
Attention mechanism

ABSTRACT

Session-based recommendation aims to predict items that a user will interact with based on historical behaviors in anonymous sessions. It has long faced two challenges: (1) the dynamic change of user intents which makes user preferences towards items change over time; (2) the uncertainty of user behaviors which adds noise to hinder precise preference learning. They jointly preclude recommender system from capturing real intents of users. Existing methods have not properly solved these problems since they either ignore many useful factors like the temporal information when building item embeddings, or do not explicitly filter out noisy clicks in sessions. To tackle above issues, we propose a novel Dynamic Intent-aware Iterative Denoising Network (DIDN) for session-based recommendation. Specifically, to model the dynamic intents of users, we present a dynamic intent-aware module that incorporates item-aware, user-aware and temporal-aware information to learn dynamic item embeddings. A novel iterative denoising module is then devised to explicitly filter out noisy clicks within a session. In addition, we mine collaborative information to further enrich the session semantics. Extensive experimental results on three real-world datasets demonstrate the effectiveness of the proposed DIDN. Specifically, DIDN obtains improvements over the best baselines by 1.66%, 1.75%, and 7.76% in terms of P@20 and 1.70%, 2.20%, and 10.48% in terms of MRR@20 on all datasets.

1. Introduction

Recommender system (RS), as an effective tool to mitigate information explosion, aims to predict user preferences and recommend what users potentially like. To improve the accuracy of recommendation, many kinds of information can be integrated in RS, such as user profiles, item descriptions and user–item interactions (Chen et al., 2019; Cheng et al., 2016; Koren et al., 2009). In most instances, however, the only available information is the previous clicked items within the current session because of privacy policy or non-logged-in users. To address this situation, session-based recommendation (SBR) is proposed to predict items which a user would interact with by considering only implicit feedbacks within the current session (Li et al., 2017).

Recently, SBR has attracted increasing attention due to its highly practical value (Wang et al., 2022). Many methods are proposed for SBR, including RNN-based models (Hidasi, Karatzoglou et al., 2016; Yu et al., 2016), attention-based models (Kang & McAuley, 2018; Liu et al., 2018) and GNN-based models (Wu et al., 2019; Xu et al., 2019). Although these approaches push the development of

* Corresponding author.
E-mail addresses: kun@mail.dlut.edu.cn (X. Zhang), hflin@dlut.edu.cn (H. Lin), xubo@dlut.edu.cn (B. Xu), cllee@whu.edu.cn (C. Li), zhlin@dlut.edu.cn (Y. Lin), liuhaifeng@mail.dlut.edu.cn (H. Liu), fenglong@psu.edu (F. Ma).

https://doi.org/10.1016/j.ipm.2022.102936
Received 17 December 2021; Received in revised form 25 February 2022; Accepted 24 March 2022
Available online 28 April 2022
0306-4573/

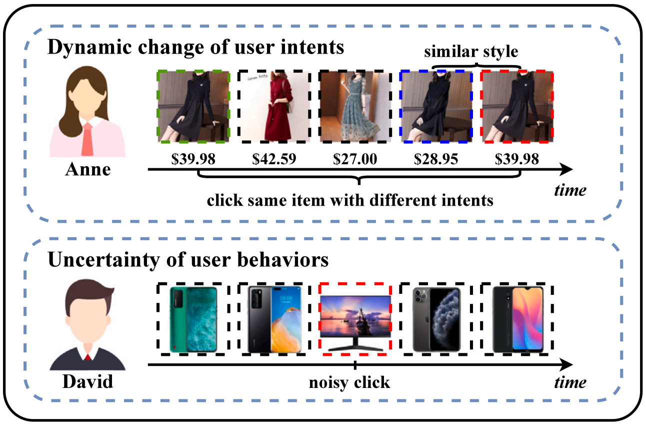


Fig. 1. Two challenges in session-based recommendation. (1) The dynamic change of user intents which makes user preferences towards items change over time. (2) The uncertainty of user behaviors which adds noise to the session.

the SBR, the following two challenges are still not thoroughly solved in SBR. The first one is *the dynamic change of user intents* (Zheng et al., 2021), which implies that the user preferences towards certain items may constantly change along the time. The first row in Fig. 1 shows the intent change of user Anne for the same kind of items. Anne viewed some dresses, among which her favorite is the first one. Before she bought the dress, she viewed another one in the similar style but with much lower price (the fourth one). Naturally, her attitude towards the first dress changed. Obviously, the dynamic change of user's intent greatly increases the difficulty of learning the user's preferences.

The second challenge is *the uncertainty of user behaviors* (Lv et al., 2020; Song et al., 2019), which refers to that users' behaviors may randomly deviate from their real intents during a session. The second row in Fig. 1 shows a session produced by another user David whose intent is to buy a cellphone. In this session, David clicked a monitor out of curiosity or by accident, which is deviated from his real intent for cellphone. This click on the monitor adds noise to the session instead of contributing to revealing his preferences for cellphone. Such noisy clicks caused by the uncertainty of user behaviors prevent recommender systems from understanding the user's need precisely.

Unfortunately, existing methods have not properly addressed these challenges. For the dynamic change of user intents, there are a few studies focusing on building dynamic item embeddings. Some recent works (Guo et al., 2019; Luo et al., 2020) represent an item differently for different users. Although they have made some progress, they ignore that the user intents change over time. We argue that for a user, *an item should be represented differently at different time*. The temporal information of items encodes the change of user intents. However, to our best knowledge, there is no existing work updating item embeddings for different users according to temporal information of items in the current session.

As to the uncertainty of user behaviors, attention mechanism is commonly employed to assign small weights on irrelevant items (Ludewig et al., 2020; Yuan et al., 2021). They usually assume that the last item works as a clue regarding the user's intent at the present time, so the last item is utilized to calculate the importance of other items. However, conventional attention mechanism (Kang & McAuley, 2018; Liu et al., 2018) used in SBR suffers from two limitations in filtering out noisy clicks: (1) The selected item (*i.e.,* the last item) used to calculate the attention weights for other items could also be a noisy click; (2) The noisy clicks with small weights could still accumulate a significant amount of noises, hindering the effective intent learning for the user.

Moreover, building dynamic item embeddings is the prerequisite for filtering out noisy items. Only when the items are comprehensively represented, is the model able to accurately distinguish the noisy items from the non-noisy items. However, as far as we know, there is none of existing methods solving these two problems in a unified way.

To tackle these challenges, we propose Dynamic Intent-aware Iterative Denoising Network (DIDN) for SBR. To model the dynamic intents, we present a dynamic intent-aware module with a dual gating mechanism to integrate various information from three perspectives: item-aware, user-aware and temporal-aware to build dynamic item embeddings. Then a novel iterative denoising module is devised to explicitly filter out noisy clicks and learn a precise embedding of the current session. We also design a collaborative session mining module that searches similar sessions to further enrich the semantics of the current session. Finally, we refine the current session embedding with these neighbor ones to enhance the intent learning. We summary the main contributions of our work as follows:

- We aim to model the dynamic user intents by exploiting personalized item embeddings. Accordingly, we present a dynamic intent-aware embedding learning strategy from item-aware, user-aware and temporal-aware perspectives simultaneously.
- We devise a novel iterative denoising module to explicitly filter out noisy clicks for the current session. Besides, we also perform collaborative session mining to further enrich the semantics for better user intent understanding.
- Extensive experiments over three public benchmarks demonstrate that our proposed DIDN achieves significant better performance consistently in terms of two metrics. To our best knowledge, we are the first to solve these two challenges in a unified way. Further analysis also confirms the validity of each design choice made in DIDN.

The remainder of this paper is organized as follows. We first briefly review the related work in Section 2, and then formulate the session-based recommendation in Section 3. In Section 4, we elaborate the proposed model DIDN. The experiment setup is described in Section 5. We present the results and analysis in Section 6. In Section 7, we discuss the theoretical and practical implications of this paper, followed by future work.

2. Related work

In this section, we briefly review the methods of session-based recommendation from following aspects, *i.e.,* traditional methods and deep-learning methods.

2.1. Traditional methods

Matrix Factorization (MF) (Koren et al., 2009; Salakhutdinov & Mnih, 2007) learns the user vectors and item vectors from user–item interaction matrix and then estimates the probability a user clicks an item by similarity between their vectors. To provide item recommendation from implicit feedback, BPR-MF (Rendle et al., 2009) optimizes personalized ranking via stochastic gradient descent. Nearest-neighbor-based approaches (Koren, 2008) is a common method in recommendations. Item-KNN (Sarwar et al., 2001) calculates the similarity between items based on their co-occurrence and recommends items which is similar to the last item in the current session. Session-KNN approaches (Garg et al., 2019; Jannach & Ludewig, 2017) determine which items should be recommended based on the session similarity. Markov chain (MC) is usually used to capture the sequential dependencies in sequences. Markov Decision Process (MDP) (Shani et al., 2005) views the problem of predicting user next behaviors as a sequential optimization problem and models the user sequential behaviors via MDP. Zimdars et al. (2001) treats the collaborative filtering problem as a time-series prediction task by using probabilistic decision-tree models. These traditional methods assume that the user intents are static and all items in the session are relevant to the user intents, which is not in line with the facts and limits their performance.

2.2. Deep-learning methods

Recurrent Neural Networks (RNN) is widely applied in SBR to model users' sequential behaviors. GRU4Rec (Hidasi, Karatzoglou et al., 2016) is the pioneering work which uses gated recurrent unit to model sequential patterns in session data. After that, more efforts are devoted to improving the performance of GRU4Rec by introducing additional features (images and text) of clicked items (Hidasi, Quadrana et al., 2016), knowledge-enhanced memory networks (Huang et al., 2018) and novel ranking loss functions (Hidasi & Karatzoglou, 2018). NARM (Li et al., 2017) enhances GRU4Rec by employing attention mechanism to capture users' main intents. CSRM (Wang, Ren et al., 2019) further improves NARM by incorporating collaborative information. RNN-based models assume that there is a rigid sequential dependencies over any successive items within a session (Wang, Hu, Wang, Cao et al., 2019), which is not always true given that there are always noisy clicks in the session.

Attention mechanism is commonly used in SBR to distinguish the importance of different items within a session. ATEM (Wang et al., 2018) applies context embedding to enrich item embeddings for predicting next items. STAMP (Liu et al., 2018) mines users' general interests and the present focus via a short-term attention/memory priority model. SASRec (Kang & McAuley, 2018) applies self-attention networks to model sequential behaviors. BERT4Rec (Sun et al., 2019) employs the deep bidirectional self-attention to mine sequential patterns. DPAN (Zhang et al., 2021) models both the collective and sequential dependencies in sessions. However, as mentioned above, these attention mechanisms have limited ability to eliminate the adverse impact of noisy clicks.

Recently, some methods (Huang et al., 2021; Pan et al., 2021, 2020; Qiu et al., 2019; Wang et al., 2021a, 2020; Xia, Yin, Yu, Wang et al., 2021) build session graph and model complex dependencies between items via graph neural networks. SR-GNN (Wu et al., 2019) applies Gated Graph Neural Networks on the session graph to learn item embeddings. LESSR (Chen & Wong, 2020) handles information loss of graph neural networks for SBR. Furthermore, some methods explore the characteristics of user behaviors (Wang et al., 2021b; Waqas et al., 2021) to improve the recommendation performance. LINet (Chen & Wong, 2019) exploits the local invariance property in sessions. RepeatNet (Ren et al., 2019) handles repeat recommendation scenarios. MCPRN (Wang, Hu, Wang, Sheng et al., 2019) models multi-interest of users. MDSR (Chen et al., 2022) aims to improve the diversification of the recommendation by exploring multi-intent of users. SLIST (Choi et al., 2021) exploits the intra- and inter-session properties in the task. CBML (Song et al., 2021) adopts a soft-clustering method and transfers shared knowledge across similar sessions. COTREC (Xia, Yin, Yu, Shao et al., 2021) combines self-supervised learning with co-training to augment session data. CAN (Yakhchi et al., 2022) unifies general and sequential recommenders to model both user's long-term and short-term interests. Although above methods have shown effective to some extent, they neither explicitly filter out noisy clicks nor model the dynamic intents of users. Other methods (Hsu & Li, 2021; Ji et al., 2020; Zhou et al., 2021) emphasize the temporal features when modeling user sequential behaviors. However, they fix the item embeddings and only use the temporal features as auxiliary information for deriving session embeddings, ignoring change of users' intents in item-level. We argue that an item in different sessions could convey different intents. Thus, it is rational to derive dynamic item embeddings for each session. Dynamic item embeddings also help model identify the noisy items, given that the temporal order affects the meaning of an item.

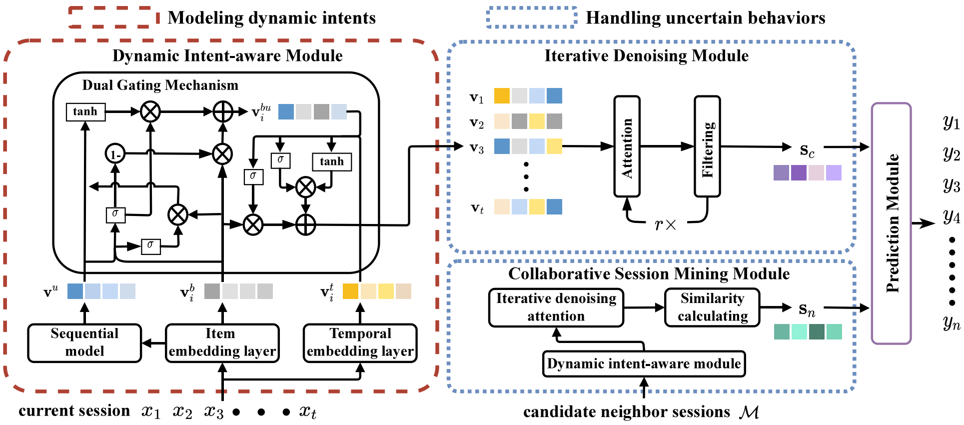


Fig. 2. The architecture of DIDN. DIDN mainly consists of four components: (1) Dynamic intent-aware module incorporates temporal and user information to construct dynamic item embeddings; (2) Iterative denoising module explicitly filters out noisy clicks and learns an accurate embedding for the current session; (3) Collaborative session mining module determines neighbor sessions and learns an auxiliary embedding for the current session; (4) Prediction module estimates the preferences of the user for different items. (*Best viewed in color*).

3. Problem formulation

The aim of session-based recommendation is to forecast next item which an user would like to click in an anonymous session. Let $\mathcal{I} = [i_1, i_2, \ldots, i_n]$ denote the set of all unique items, where $|\mathcal{I}| = n$ is the number of items. An anonymous user produces a session $\mathcal{S} = [x_1, x_2, \ldots, x_t]$, where $x_i \in \mathcal{I}$ and t is the length of session $\mathcal{S}$. The proposed model DIDN can be viewed as a function f whose input is the current session $\mathcal{S}$. The output is $\mathbf{y} = f(\mathcal{S}) = [y_1, y_2, \ldots, y_n]$, where y_j ($j \in [1, \ldots, n]$) indicates the likelihood that the user will interact with item i_j next.

4. The proposed DIDN model

The proposed DIDN is illustrated in Fig. 2. We firstly introduce a *dynamic intent-aware module* with a dual gating mechanism which constructs dynamic item embeddings by incorporating temporal and user information. We then elaborate two modules that handle the uncertainty of user behaviors: (1) *iterative denoising module* explicitly filters out noisy clicks within sessions; and (2) *collaborative session mining module* searches similar sessions to further enrich the semantics. Finally, we predict the next behavior of a user by the *prediction module*. Next, we describe each component in detail.

4.1. Dynamic intent-aware module

The goal of this module is to learn dynamic item embeddings, which are highly related to many factors: (1) every item owns its inherent characteristics which are independent of users and the temporal order it appears in the session; (2) different users may hold different preferences towards an item; (3) a user may have different preferences towards the same item at different times (ref. Anne's example in Fig. 1). Hence, we first represent an item from three aspects: item-aware, user-aware and temporal-aware perspectives, and then aggregate them via a dual gating mechanism.

Basic Item Embedding. We transform each item x_i into a dense vector $\mathbf{v}_i^b \in \mathbb{R}^d$. We call $\mathbf{v}_i^b$ the basic item embedding. Here, we share $\mathbf{v}_i^b$ for all sessions that contain item x_i, such that the inherent characteristics of the item without considering the user and temporal information are well encoded.

User Embedding. In most scenarios, user profile information is often not available due to many reasons. Hence, we view a session as a user and aim to derive user general intent by considering the previous t interacted items in the session together. For simplicity and efficiency, we choose to apply the sequential model GRU4Rec (Hidasi, Karatzoglou et al., 2016) over $\mathcal{S}$, and the last hidden state of GRU4Rec is considered as the embedding of the user, *i.e.,* $\mathbf{v}^u = GRU([\mathbf{v}_1^b, \ldots, \mathbf{v}_t^b]) \in \mathbb{R}^d$.

Temporal Embedding. One of advantages of the proposed DIDN is to model dynamic intents of a user by incorporating temporal information. In particular, we first index the temporal information associated with an item in session $\mathcal{S}$ as follows:

$$T_i^l = \kappa \times l - i \tag{1}$$

where i is the temporal order of item x_i in the session, l is the length of the session, and κ is a coefficient whose purpose is to make sure that the same temporal order with different lengths is assigned with different values. Here, we set κ to be the max length of all sessions in the dataset. As we can see in Eq. (1), both session length and the temporal order of an item x_i affect the index calculation. We treat the same temporal order of sessions with different lengths differently because that the same position has different meanings in sessions with different lengths. For example, given two sessions $S_1 = [x_1, x_2]$ and $S_2 = [x_1, x_2, x_3, x_4]$, item x_2 is at second position in both S_1 and S_2. However, given that the length of S1 and S2 is different, x_2 is the last item in S_1, while the user in session S_2 has clicked several other items after x_2. It indicates that same positions with different session lengths possess different temporal meanings. With index T_i^l, we utilize a lookup table to extract the corresponding index embedding $\mathbf{v}_i^t \in \mathbb{R}^d$.

Dual Gating Mechanism. Right now, we have learned three different embeddings (*i.e.,* $\mathbf{v}_i^b$, $\mathbf{v}^u$ and $\mathbf{v}_i^t$) from different perspectives. Inspired by the gating mechanism proposed in GRU (Cho et al., 2014) and LSTM (Hochreiter & Schmidhuber, 1997), we devise a dual gating mechanism to hierarchically integrate these three embeddings for dynamic intent modeling. Specifically, we first merge basic item embedding $\mathbf{v}_i^b$ and user embedding $\mathbf{v}^u$ as follows:

$$\mathbf{z} = \sigma(\mathbf{W}_z \mathbf{v}^u + \mathbf{U}_z \mathbf{v}_i^b) \tag{2}$$

$$\mathbf{r} = \sigma(\mathbf{W}_r \mathbf{v}^u + \mathbf{U}_r \mathbf{v}_i^b) \tag{3}$$

$$\hat{\mathbf{v}}_i^{bu} = tanh(\mathbf{W}\mathbf{v}^u + \mathbf{U}(\mathbf{r} \otimes \mathbf{v}_i^b)) \tag{4}$$

$$\mathbf{v}_i^{bu} = (\mathbf{1} - \mathbf{z}) \otimes \mathbf{v}_i^b + \mathbf{z} \otimes \hat{\mathbf{v}}_i^{bu}, \tag{5}$$

where $\otimes$ denotes element-wise product, σ is the sigmoid function and $\mathbf{W}_*, \mathbf{U}_*, \in \mathbb{R}^{d\times d}$ are learnable parameters. As shown in Eq. (5), we merge the basic item embedding and user embedding in a complementary way, which balances the contribution between item and user information. We call $\mathbf{v}_i^{bu}$ user-aware item embedding. For different users (*i.e.,* sessions), we can learn different user-aware item embeddings for an item, which is consistent with the facts that an item should be represented differently for different users. It also increases the interpretability and rationality of the proposed DIDN. And then, we integrate the temporal embedding $\mathbf{v}_i^t$ with $\mathbf{v}_i^{bu}$ as follows:

$$\mathbf{f} = \sigma(\mathbf{W}_f \mathbf{v}_i^{bu} + \mathbf{U}_f \mathbf{v}_i^t) \tag{6}$$

$$\mathbf{i} = \sigma(\mathbf{W}_i \mathbf{v}_i^{bu} + \mathbf{U}_i \mathbf{v}_i^t) \tag{7}$$

$$\hat{\mathbf{v}}_i = tanh(\mathbf{W}_t \mathbf{v}_i^{bu} + \mathbf{U}_t \mathbf{v}_i^t) \tag{8}$$

$$\mathbf{v}_i = \mathbf{f} \otimes \mathbf{v}_i^b + \mathbf{i} \otimes \hat{\mathbf{v}}_i \tag{9}$$

where $\mathbf{W}_*, \mathbf{U}_* \in \mathbb{R}^{d\times d}$ are learnable parameters, and $\mathbf{v}_i$ is final item embedding for x_i. Different from Eqs. (2)–(5), we integrate user-aware item embedding with temporal embedding by two independent gate (f, i) with no restrictions, which highlights the importance of temporal information. The $\mathbf{v}_i$ encodes the dynamic intent of a user from item-aware, user-aware and temporal-aware perspectives simultaneously. Note that the gating operation enables our model to capture high-order relations among features from different embeddings and the hierarchical structure makes our model more interpretable and rational. The dynamic item embeddings ($\mathbf{v}_i$) also endows the proposed DIDN with the ability to identify the noisy clicks.

4.2. Iterative denoising module

Recalling the example of user David illustrated in Fig. 1, it is likely to include noisy clicks in a session. Here, we devise a novel iterative denoising module which consists of multiple denoising layers. Without loss of generality, the jth denoising layer mainly contains two operations, *i.e., attention calculation* and *item filtering*. Specifically, the attention weight in jth layer is calculated via:

$$\alpha_i^j = \mathbf{u}^j \sigma(\mathbf{A}_1^j \mathbf{v}_i + \mathbf{A}_2^j \mathbf{s}^{j-1}) \tag{10}$$

where $\mathbf{u}^j$ is the attentive vector for jth layer, $\mathbf{A}_1^j, \mathbf{A}_2^j \in \mathbb{R}^{d\times d}$ are learnable parameters, and $\mathbf{s}^{j-1}$ is session representation derived in the previous layer. Prior to the first layer, we initialize $\mathbf{s}^0$ via an attention mechanism as follows:

$$\mathbf{s}^0 = \sum_{i=1}^{t} \beta_i \mathbf{v}_i \tag{11}$$

$$\beta_i = \frac{exp(\mathbf{w}^T \mathbf{u}_i)}{\sum_{j=1}^{t} \exp(\mathbf{w}^T \mathbf{u}_j)} \tag{12}$$

$$\mathbf{u}_i = \tanh(\mathbf{W}_0 \mathbf{v}_i + \mathbf{b}) \tag{13}$$

where $\mathbf{w}, \mathbf{W}_0, \mathbf{b}$ are learnable parameters. To avoid the accumulation of noisy clicks, we filter out the items with weights below a specific threshold:

$$\alpha_i^j = \begin{cases} \alpha_i^j & \alpha_i^j - \lambda \alpha_{avg}^j > 0 \\ 0 & \text{others} \end{cases} \tag{14}$$

where λ is a parameter controlling the degree of denoising, and the product of λ and the averaged attention weight $\alpha_{avg}^j = \frac{1}{t}\sum_{k=1}^{t} \alpha_k^j$ determines the threshold. The session representation in the jth layer ($j \geqslant 1$) is calculated as follows:

$$\mathbf{s}^j = \sum_{i=1}^{t} \alpha_i^j \mathbf{v}_i, \tag{15}$$

After r-layer attention calculation and item filtering, $\mathbf{s}^r$ is viewed as the embedding of the current session:

$$\mathbf{s}_c = \mathbf{s}^r \tag{16}$$

Note that we utilize this multi-layer design to refine the filtering decision iteratively, which could enhance the effectiveness of the denoising purpose. Our experimental results also validate the positive impact of this iterative denoising mechanism.

4.3. Collaborative session mining module

Intuitively, it is unlikely for users with similar intents to click same noisy items, which could guide us to extract similar sessions for semantic enrichment. We first take the recent $\mathcal{M}$ sessions containing items interacted by current session $\mathbf{s}_c$ as candidate neighbor sessions. Afterwards, for every candidate neighbor session, we feed it into the dynamic intent-aware module and the iterative denoising module to obtain the session representation. In such a way, we can get the representation of every candidate neighbor session $\mathbf{s}_n^j$. Note that the candidate sessions go through the dynamic intent-aware model and iterative denoising module, which enables them to accurately represent users' underlying intents. To obtain the relevance between current session $\mathbf{s}_c$ and every candidate session $\mathbf{s}_n^j$, we then calculate the relevance score between $\mathbf{s}_c$ and $\mathbf{s}_n^j$ via cosine similarity:

$$sim_j = \frac{\mathbf{s}_c^T \mathbf{s}_n^j}{\|\mathbf{s}_c\| \times \|\mathbf{s}_n^j\|}. \tag{17}$$

The R most relevant candidate sessions are considered as the neighbor sessions corresponding to the current session. Then, we obtain the auxiliary embedding for the current session as follows:

$$\gamma_j = \frac{exp(\tau \cdot sim_j)}{\sum_{k=1}^{R} exp(\tau \cdot sim_k)} \tag{18}$$

$$\mathbf{s}_n = \sum_{k=1}^{R} \gamma_k \mathbf{s}_n^k. \tag{19}$$

where τ is the temperature parameter to adjust the discriminative strength.

4.4. Prediction module

We have learned the current session representation $\mathbf{s}_c$ and its auxiliary representation $\mathbf{s}_n$. And then, we fuse the $\mathbf{s}_c$ and $\mathbf{s}_n$ to obtain the user intents as follows:

$$\mathbf{s} = tanh(\mathbf{W}_f[\mathbf{s}_c; \mathbf{s}_n; \mathbf{s}_c + \mathbf{s}_n; \mathbf{s}_c \otimes \mathbf{s}_n] + \mathbf{b}_f), \tag{20}$$

where $\mathbf{W}_f$ and $\mathbf{b}_f$ are learnable parameters. Note that we make the $\mathbf{s}_c$ and $\mathbf{s}_n$ interact with each other in different spaces, *i.e.,* original space, addition space and product space, which enables our model to capture the rich dependencies between $\mathbf{s}_c$ and $\mathbf{s}_n$. And then we merge the information from all spaces to enhance the representations of user intents. After that, we calculate the likelihood score y_i with respective to candidate item x_i as,

$$y_i = \frac{exp(\mathbf{s}^T \mathbf{v}_i^b)}{\sum_{j=1}^{n} exp(\mathbf{s}^T \mathbf{v}_j^b)} \tag{21}$$

At last, we train our model using cross-entropy loss as:

$$\mathcal{L}(\mathbf{p}, \mathbf{y}) = -\sum_{j=1}^{n} p_j \log(y_j) + (1 - p_j)\log(1 - y_j), \tag{22}$$

where p_j is the ground truth indicating whether the user will click on item i_j.

5. Experiment setup

We conduct experiments on three real-world benchmarks to validate the efficacy of our proposed DIDN by answering following research questions:

- **RQ1** How does DIDN perform compared with the existing SBR alternatives? (ref. Section 6.1)
- **RQ2** Will each design module contribute positively towards DIDN's performance? (ref. Sections 6.2–6.4)
- **RQ3** What is the influence of hyper-parameters on DIDN? (ref. Section 6.5)
- **RQ4** How does the session length influence the performance of the session-based recommendation? (ref. Section 6.6)

Table 1
Statistics of the datasets.

Statistics	Yoochoose 1/64	Yoochoose 1/4	Diginetica
#clicks	557,248	8,326,407	982,961
#train	332,873	532,597	647,523
#validation	36,986	59,177	71,947
#test	55,898	55,898	60,858
#items	16,776	29,618	43,097
avg.length	6.16	5.71	5.12

5.1. Datasets

We utilize following two public real-world benchmarks for performance evaluation,

- Yoochoose[1] is a public dataset released by RecSys Challenge 2015. It contains a stream of user clicks on an e-commerce website within six months.
- Diginetica[2] is an e-commerce dataset which is composed of user purchasing behaviors on the websites.

For a fair comparison, following Chen and Wong (2020), Wu et al. (2019), we delete sessions which only contain one item. The items that appear less than 5 times in all sessions are also filtered out in both datasets. The sessions of subsequent day/week consist of test datasets for Yoochoose/Diginetica. We also filter out items that do not appear in the training set from the test set for both datasets. Data augmentation is used to account for temporal shifts in sessions (Tan et al., 2016). Besides, we randomly choose 10% of the training set as the validation set. Following the common settings in Li et al. (2017), Liu et al. (2018) and Wu et al. (2019), for the Yoochoose dataset, the most recent portions 1/64 and 1/4 are used as two split datasets which are marked as "Yoochoose1/64" and "Yoochoose1/4", respectively. Table 1 summarizes the statistics for the preprocessed dataset.

5.2. Metrics

To evaluate the performance of our proposed DIDN and all baselines, we adopt two widely used metrics P@k and MRR@k. Given the predicted recommendation lists, **P@**k (Precision) measures the proportion of cases in which the ground-truth item is within the top-k list. As for **MRR@**k (Mean Reciprocal Rank), it is the average of reciprocal ranks of the ground-truth items among the recommendation lists. Note that larger metric values indicate better performances for both P@k and MRR@k.

5.3. Baselines

In order to evaluate the performance of the proposed approach DIDN, we compare it with the following competitive baselines:

- S-POP recommends the most frequent items in the current session.
- Item-KNN (Sarwar et al., 2001) recommends items which is most similar to the previous items in the session, where the cosine similarity is adopted.
- BPR-MF (Rendle et al., 2009) optimizes matrix factorization using a pairwise ranking loss. To make it suitable for SBR, we represent a session by mean latent vectors of items in it.
- GRU4Rec+ (Tan et al., 2016) enhances GRU4Rec (Hidasi, Karatzoglou et al., 2016) with a data augmentation strategy and a new ranking loss function.
- NARM (Li et al., 2017) improves GRU4Rec (Hidasi, Karatzoglou et al., 2016) by applying an attention mechanism to explore users' main purposes.
- STAMP (Liu et al., 2018) captures users' general interests from session context and the recent intents from the last-clicks via the attention mechanism.
- CoSAN (Luo et al., 2020) learns a dynamic item representation by aggregating the embeddings of neighbor sessions.
- CSRM (Wang, Ren et al., 2019) extends NARM by incorporating auxiliary information from neighbor sessions.
- SR-GNN (Wu et al., 2019) builds session graph and models the complex item transitions within sessions by graph neural networks.
- LESSR (Chen & Wong, 2020) tackles information loss of GNN-based model in SBR by introducing shortcut graph attention and edge-order preserving aggregation layers.

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

Table 2
Performance comparison of DIDN with baselines over three datasets.

Method	Yoochoose1/64		Yoochoose1/4		Diginetica	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
S-POP	30.44	18.35	27.08	17.75	21.06	13.68
Item-KNN	51.60	21.81	52.31	21.70	35.75	11.57
BPR-MF	31.31	12.08	3.40	1.57	5.24	1.98
GRU4Rec+	67.84	29.00	69.11	29.22	46.16	14.69
NARM	68.32	28.63	69.73	29.23	49.70	16.17
STAMP	68.74	29.67	70.44	30.00	45.64	14.32
CoSAN	69.93	28.74	70.75	29.16	48.59	15.69
CSRM	69.85	29.71	70.63	29.48	51.69	16.92
SR-GNN	70.57	30.94	71.36	<u>31.89</u>	50.73	17.59
LESSR	<u>70.94</u>	<u>31.16</u>	<u>71.40</u>	31.56	<u>52.17</u>	<u>18.13</u>
DIDN	**72.12***	**31.69***	**72.65***	**32.59***	**56.22***	**20.03***

The results produced by the best baseline and the best performer in each column are underlined and boldfaced respectively. * denotes the significant difference for t-test ($p < 0.01$).

5.4. Implementation detail

Hyper-parameters of DIDN and all baselines are optimized via the grid search on the validation set. Accordingly, we use the following hyper-parameters for DIDN: $d = 64$, $r = 3$, $\lambda = 0.1$, $\mathcal{M} = 512$, $R = 5$, $\tau = 1$, $\kappa = 19$. We share item embeddings in the basic item embedding layer and prediction layer. To alleviate overfitting, we employ the dropout strategy in item embedding layer and temporal embedding layer with 40% dropout. We use Adam to perform model optimization. The learning rate and mini-batch size is set to 0.001 and 512 respectively. The number of epoch is set to 100. The implementation of our proposed model is released at https://github.com/Zhang-xiaokun/DIDN.

6. Results and analysis

6.1. Overall performance (RQ1)

A summary of experimental results of all methods over the three datasets are reported in Table 2. Here, we can make the following observations.

Firstly, deep learning methods achieve large performance improvement over traditional methods in all datasets on both metrics, which demonstrates the effectiveness of deep learning methods for this task. Limited by weak ability to extract information, traditional methods can neither model the dynamic intents of users nor effectively filter out noisy clicks within the session.

Secondly, NARM outperforms GRU4Rec by applying the attention mechanism, and CSRM enhances NARM by incorporating neighbor sessions as auxiliary information. We argue that both attention mechanism and neighbor sessions used in the methods could mitigate the impact of uncertain behaviors to a certain extent, leading to better performance.

Thirdly, CoSAN improves the accuracy of modeling user behaviors by representing item differently for different users. However, its result is still inferior to the proposed DIDN which also considers temporal information besides user information. It demonstrates that updating item embeddings according to time contributes to modeling the real intents of users.

Fourthly, due to modeling the dynamic intents of users and effectively filtering out noisy clicks, the proposed DIDN achieves the best performance in all datasets on both metrics. Specifically, DIDN obtains large improvements over the best baselines by 1.66%, 1.75%, and 7.76% in terms of P@20 on the three datasets, respectively. In terms of MRR@20, the relative improvements are 1.70%, 2.20%, and 10.48%. As we can see, DIDN obtains larger improvements on Diginetica compared with other two Yoochoose datasets. As shown in Table 1, Diginetica contains more items than other datasets, suggesting there are more noisy clicks and more diverse user intents. With the help of the dynamic intent-aware module and iterative denoising module, DIDN is capable to model the dynamic intents of users and effectively filter out noisy clicks. The consistent improvements of the proposed DIDN over all baselines verifies the effectiveness of our DIDN.

6.2. Impact of dynamic intent-aware module (RQ2)

To model the dynamic intents of users, we utilize temporal and user information (*i.e.,* $\mathbf{v}^t$ and $\mathbf{v}^u$) in DIDN and propose a dual gating mechanism to integrate various information from multiple perspectives. We demonstrate the effectiveness of these design choices with the following variants:

- DIDN$_{-u}$ is similar as DIDN but without incorporating user information when representing items. It only integrates item information and temporal information into final item embeddings by Eqs. (6)–(9).
- DIDN$_{-t}$ is similar as DIDN but without incorporating temporal information when representing items.
- DIDN$_{-ut}$ is similar as DIDN but without incorporating user or temporal information when representing items. In other words, We replace dynamic intent-aware module with simple item embedding layer in this variant.

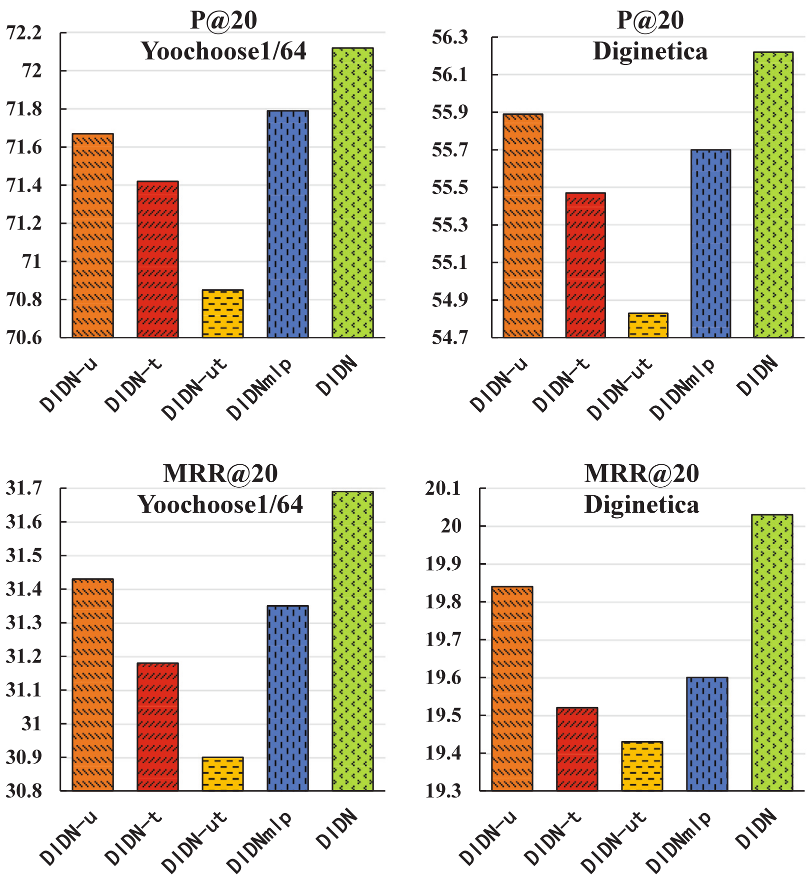


Fig. 3. Impact of dynamic intent-aware module.

Table 3
Impact of different temporal embedding.

Method	Yoochoose1/64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
DIDN_{for}	71.52	31.23	55.13	19.21
DIDN_{rev}	71.83	31.36	55.87	19.75
DIDN	**72.12***	**31.69***	**56.22***	**20.03***

- DIDN_{mlp} replaces the dual gating unit with MLP to merge all information. According to the performance, we apply MLP containing two hidden layers in here.

According to Fig. 3, we have some insights: (1) DIDN achieves better performance over DIDN_{-ut}, which shows the superiority of dynamic intent-aware module in representing items in SBR. (2) DIDN defeats DIDN_{-t} and DIDN_{-u} in both datasets on both metrics. It indicates that the temporal information and user information is complementary for modeling dynamic intents of users. (3) DIDN_{-u} outperforms DIDN_{-t} to some extent, indicating that the temporal information is relatively an important feature. (4) DIDN achieves better improvement over DIDN_{mlp}, which demonstrates that the proposed dual gating mechanism is better in integrating temporal and user information.

Moreover, as discussed in Section 4.1, both the temporal order of an item and the length of the session affect the item's temporal semantics simultaneously. Therefore, we design a new method, as formulated in Eq. (1), to encode the temporal information of items. To demonstrate the effectiveness of the proposed temporal encoding, the following variants of DIDN are designed: DIDN_{for} with commonly used forward position encoding as in Kang and McAuley (2018), Pan et al. (2020) and Sun et al. (2019); DIDN_{rev} applies the reversed position encoding as in Wang et al. (2020) and Xia, Yin, Yu, Wang et al. (2021). As we can observe in the Table 3, DIDN achieves best performance in terms of P@20 and MRR@20 in Yoochoose1/64 and Diginetica. It indicates that our proposed method of temporal encoding can represent temporal information more accurately and contribute revealing the dynamic nature of user intents.

Table 4
Impact of iterative denoising.

Method	Yoochoose1/64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
DIDN$_{-id}$	71.38	31.19	55.07	19.26
DIDN	**72.12***	**31.69***	**56.22***	**20.03***

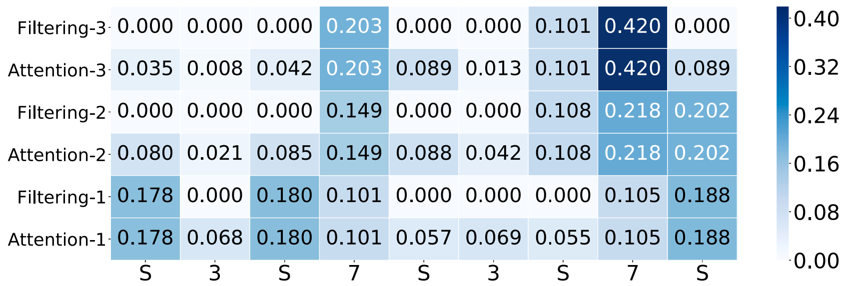


Fig. 4. Attention visualization. The depth of the color indicates the importance of an item.

Table 5
Impact of collaborative session.

Method	Yoochoose1/64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
DIDN$_{-n}$	71.08	31.01	54.86	19.04
DIDN	**72.12***	**31.69***	**56.22***	**20.03***

6.3. Impact of iterative denoising module(RQ2)

In order to make up for the flaws of conventional attention mechanism as mentioned in previous sections, we propose a novel iterative denoising module to denoise the sessions. DIDN$_{-id}$ utilizes conventional attention layer used in Liu et al. (2018) and Wu et al. (2019) to represent current session. As shown in Table 4, DIDN outperforms DIDN$_{-id}$ in both metrics on Yoochoose1/64 and Diginetica. It demonstrates that the proposed iterative denoising module is effective to filter out the noisy user behaviors.

Moreover, we randomly choose a session from Yoochoose 1/64. The attention weights α_i^j of items are displayed in Fig. 4. The *Attention-k* indicates the attention weights derived by kth layer. In the absence of item specific information, we explain denoising process with the category of an item. Besides the category id, there is also a special category named "S" to indicate something special by the dataset, including the items on sale, and so on. Note that most items in the dataset belong to this category, indicating that this category covers little semantics.

For the example in Fig. 4, the target item belongs to category "7". Here, we make the following observations: (1) There indeed exists noisy clicks in sessions. Except for items with the same category as that of the target item, other items could be considered as noisy clicks within the session. (2) The last item, which is usually selected as the criterion to calculate importance of the other items, cloud also be a noisy click. Different from previous methods (Kang & McAuley, 2018; Liu et al., 2018) that utilize some specific items as criterion, the proposed iterative denoising module utilizes a representative vector $\mathbf{s}^j$ to calculate the importance. We believe that this is one of the reasons why our model works well. (3) The filtering operation enables the model to understand the actual intents of users. As we can see from Attention-3, although the weights of noisy clicks are relatively small (0.035, 0.008, 0.042, 0.089, 0.013, 0.089), they can accumulate to a large value (0.276) which is even larger than the weight of the relevant item (0.203). Therefore, it is difficult for the model to capture the user's actual intent if small weights of noisy clicks remain in the session. (4) The iterative denoising process can correct the wrong denoising decision made in lower layers. As shown by Attention-1, the relevant items get small weights (0.105) because there are several noisy clicks in the session. After Filtering-1, some noisy clicks are removed, which could help the denoising process made by Attention-2. The same thing happens in the third layer. The iterative structure extracts users' actual intents by multiple attention calculation and item filtering.

6.4. Impact of collaborative session mining module (RQ2)

Besides the iterative denoising module, we further examine the impact of the collaborative session mining module by removing it from DIDN (DIDN$_{-n}$). As shown in Table 5, DIDN perform much better than DIDN$_{-n}$ across the two datasets. It demonstrates that the collaborative session mining module is able to extract useful information from other sessions and improve the accuracy of modeling user behavior. We believe that the collaborative information mined from neighbor sessions could filter out noisy clicks to a certain extent. The probability of similar sessions containing the same noise is small, because it is unlikely for different users with

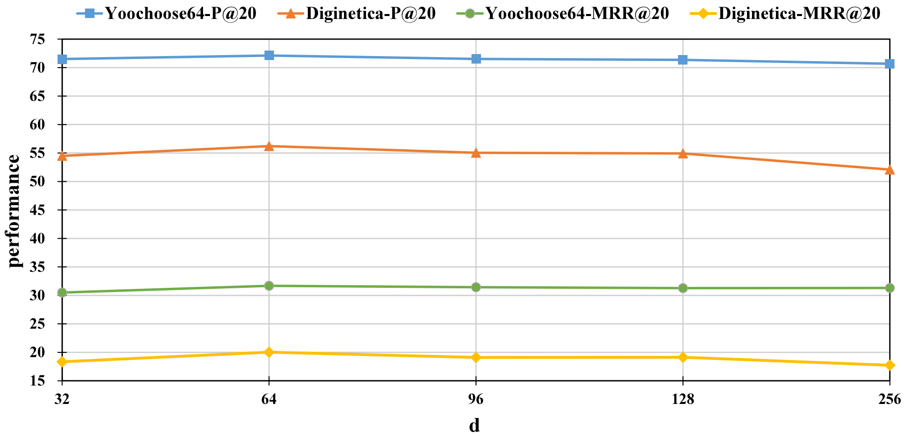


Fig. 5. Impact of the embedding dimension.

Table 6
Impact of the iteration number.

Method	Yoochoose1/64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
$DIDN_1$	71.64	31.29	55.39	19.61
$DIDN_2$	71.85	31.43	55.74	19.76
$DIDN_3$	**72.12***	**31.69***	**56.22***	**20.03***
$DIDN_4$	71.71	31.24	55.82	19.87

similar interest to inadvertently click same noisy items. As a result, we can utilize collaborative information to mitigate the impact of individual uncertainty.

6.5. Influence of hyper-parameters (RQ3)

We evaluate the influence of embedding dimension d and iteration number r in terms of both metrics in Yoochoose1/64 and Diginetica. The effects of these two hyper-parameters are shown in Fig. 5 and Table 6. Firstly, we provide the experimental results under different dimension d which ranges in $\{32, 64, 96, 128, 256\}$ in Fig. 5. Generally speaking, as we increase the dimension, the representation ability of the model becomes more expressive. In DIDN, we represent items from different perspectives and merge the different embeddings by dual gating mechanism. Therefore, our DIDN can have strong representation ability by using relative small dimension (*i.e.,* $d = 64$) in each perspective. And then, we investigate the impact of the iteration number. We increase the number of iteration from 1 to 4 (i.e., $DIDN_i$, $i \in [1, 2, 3, 4]$) to examine the performance pattern of DIDN. As shown in Table 6, too much or too little iterations degrade the performance of the model. Hence, we choose $r = 3$ in our experiments. We believe that less iterations cannot effectively denoise the sessions, while more iterations lead to overfitting.

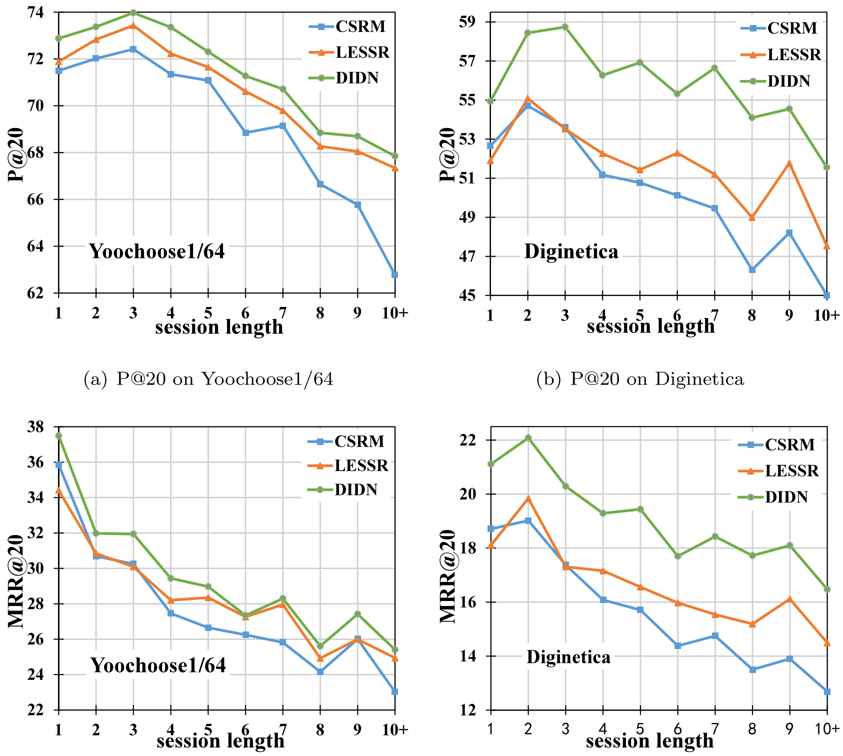
6.6. Impact of session length (RQ4)

In this section, we present the performance of DIDN and two representative baselines CSRM and LESSR under various session lengths (*i.e.,* t). The performance patterns of above three models are shown in Fig. 6. Please see the curve of P@20 in Yoochoose1/64 as an example (ref. Fig. 6(a)). With the increase of session length, all models' performance increases first and then shows a continuous downward trend. Specifically, when t is small ($1 \leqslant t \leqslant 3$), with the increase of t, users expose more information to models, which helps the models to capture users' intents. However, longer sessions ($t > 3$) may contain more noisy clicks due to the uncertainty of user behaviors, which degrades the performance of all models. DIDN outperforms CSRM and LESSR at all lengths on both datasets in both metrics, which validates the superiority of DIDN for SBR again. We believe the performance gain comes from the following reasons: (1) DIDN considers the dynamic change of user intents; (2) For short sessions, DIDN incorporates similar sessions which introduce more useful information to the model; (3) For long sessions, the built-in iterative denoising module further alleviates the impact of uncertain behaviors.

7. Conclusion

7.1. Theoretical and practical implications

In this paper, we propose Dynamic Intent-aware Iterative Denoising Network (DIDN) to address two longstanding challenges in session-based recommendation, *i.e.,* the dynamic change of user intents and the uncertainty of user behaviors. Different from



(a) P@20 on Yoochoose1/64 (b) P@20 on Diginetica

(c) MRR@20 on Yoochoose1/64 (d) MRR@20 on Diginetica

Fig. 6. Impact of session length.

previous studies, DIDN builds dynamic item embeddings for different users according to different time and explicitly filters out noisy items within sessions. Specifically, we present a dynamic intent-aware module with a dual gating mechanism incorporating various information from different perspectives to build dynamic item embeddings. And then, we devise a novel iterative denoising module to explicitly filter out noisy clicks within sessions. Besides, we also harness collaborative information to further enrich the semantics of the current session.

Extensive experiments conducted on three real-world datasets demonstrate the effectiveness of our proposed DIDN. Further ablation studies show that it is meaningful to represent items differently for different users at different time. And explicitly filtering out noisy items within sessions is helpful to mine real intents of users. Moreover, collaborative information contributes to enhancing session semantics. All of above insights broaden our understanding for the task of session-based recommendation and endow recommender system the potential to accurately predict user behavior.

As for application, our work is also of practical significance. In this paper, we study the two longstanding challenges in session-based recommendation and propose a solution to the problems. This can help e-commerce platforms to understand real needs of users, thus improving the accuracy of user behavior prediction, providing more user-friendly experience and increasing revenue. Additionally, the proposed dual gating mechanism can be applied to merge different features in other tasks, *e.g.,* merging representations of different features in multimodal tasks. And, the iterative denoising module can be used as a fundamental module in tasks of Natural Language Processing, such as sentiment analysis, question answering and machine translation, to filter out irrelevant words within a sentence or document for improving the performance of corresponding models.

7.2. Future work

In the future, we plan to incorporate more auxiliary information into DIDN, *e.g.,* image, category, and title of item, to enrich the semantics of item embeddings (Li et al., 2019; Zheng et al., 2017). And, it is also an interesting direction to construct an explainable recommender system which helps users understand the reasons for recommendations, thus improving users' trust in the system.

CRediT authorship contribution statement

Xiaokun Zhang: Conceptualization, Methodology, Data curation, Formal analysis, Experiments, Visualization, Writing – original draft, Writing – review & editing. **Hongfei Lin:** Supervision, Funding acquisition, Project administration. **Bo Xu:** Writing – review

& editing, Investigation, Funding acquisition. **Chenliang Li:** Methodology, Investigation, Writing – review & editing. **Yuan Lin:** Writing – review & editing. **Haifeng Liu:** Writing – review & editing. **Fenglong Ma:** Writing – review & editing.

Acknowledgments

We thank the handling editors and reviewers for their effort and constructive expert comments. This work has been supported by the Natural Science Foundation of China (No. 62076046, No. 61976036, No. 62006034) and Major science and technology projects of Yunnan Province (202002ab080001-1).

References

Chen, W., Cai, F., Chen, H., & de Rijke, M. (2019). Joint neural collaborative filtering for recommender systems. *ACM Transactions on Information Systems*, *37*(4), 39:1–39:30.

Chen, W., Ren, P., Cai, F., Sun, F., & de Rijke, M. (2022). Multi-interest diversification for end-to-end sequential recommendation. *ACM Transactions on Information Systems*, *40*(1), 20:1–20:30.

Chen, T., & Wong, R. C. (2019). Session-based recommendation with local invariance. In *Proceedings of 2019 IEEE International conference on data mining* (pp. 994–999). IEEE.

Chen, T., & Wong, R. C. (2020). Handling information loss of graph neural networks for session-based recommendation. In *Proceedings of the 26th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1172–1180). ACM.

Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., & Shah, H. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* DLRS@RecSys, (pp. 7–10).

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP 2014, Eighth workshop on syntax, semantics and structure in statistical translation* (pp. 103–111). Association for Computational Linguistics.

Choi, M., Kim, J., Lee, J., Shim, H., & Lee, J. (2021). Session-aware linear item-item models for session-based recommendation. In *Proceedings of the web conference 2020* (pp. 1604–1614). ACM / IW3C2.

Garg, D., Gupta, P., Malhotra, P., Vig, L., & Shroff, G. M. (2019). Sequence and time aware neighborhood for session-based recommendations: STAN. In *Proceedings of the 42nd International ACM SIGIR conference on research and development in information retrieval* (pp. 1069–1072). ACM.

Guo, G., Ouyang, S., He, X., Yuan, F., & Liu, X. (2019). Dynamic item block and prediction enhancing block for sequential recommendation. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 1373–1379). ijcai.org.

Hidasi, B., & Karatzoglou, A. (2018). Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International conference on information and knowledge management* (pp. 843–852). ACM.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In *Proceedings of 4th International conference on learning representations*.

Hidasi, B., Quadrana, M., Karatzoglou, A., & Tikk, D. (2016). Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on recommender systems* (pp. 241–248). ACM.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hsu, C., & Li, C. (2021). RetaGNN: Relational temporal attentive graph neural networks for holistic sequential recommendation. In *Proceedings of the web conference 2021* (pp. 2968–2979). ACM / IW3C2.

Huang, C., Chen, J., Xia, L., Xu, Y., Dai, P., Chen, Y., Bo, L., Zhao, J., & Huang, J. X. (2021). Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In *Proceedings of Thirty-Fifth AAAI conference on artificial intelligence* (pp. 4123–4130). AAAI Press.

Huang, J., Zhao, W. X., Dou, H., Wen, J., & Chang, E. Y. (2018). Improving sequential recommendation with knowledge-enhanced memory networks. In *Proceedings of the 41st International ACM SIGIR conference on research & development in information retrieval* (pp. 505–514). ACM.

Jannach, D., & Ludewig, M. (2017). When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems* (pp. 306–310). ACM.

Ji, W., Wang, K., Wang, X., Chen, T., & Cristea, A. (2020). Sequential recommender via time-aware attentive memory network. In *Proceedings of the 29th ACM international conference on information and knowledge management* (pp. 565–574). ACM.

Kang, W., & McAuley, J. J. (2018). Self-attentive sequential recommendation. In *Proceedings of IEEE international conference on data mining* (pp. 197–206). IEEE Computer Society.

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 426–434). ACM.

Koren, Y., Bell, R. M., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *IEEE Computer*, 30–37.

Li, C., Niu, X., Luo, X., Chen, Z., & Quan, C. (2019). A review-driven neural model for sequential recommendation. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 2866–2872). ijcai.org.

Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., & Ma, J. (2017). Neural attentive session-based recommendation. In *Proceedings of the 26th ACM on conference on information and knowledge management* (pp. 1419–1428). ACM.

Liu, Q., Zeng, Y., Mokhosi, R., & Zhang, H. (2018). STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1831–1839). ACM.

Ludewig, M., Mauro, N., Latifi, S., & Jannach, D. (2020). Empirical analysis of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction*, 1–33.

Luo, A., Zhao, P., Liu, Y., Zhuang, F., Wang, D., Xu, J., Fang, J., & Sheng, V. S. (2020). Collaborative self-attention network for session-based recommendation. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence* (pp. 2591–2597). ijcai.org.

Lv, Y., Zhuang, L., Luo, P., Li, H., & Zha, Z. (2020). Time-sensitive collaborative interest aware model for session-based recommendation. In *Proceedings of IEEE international conference on multimedia and expo* (pp. 1–6).

Pan, Z., Cai, F., Chen, W., & Chen, H. (2021). Graph co-attentive session-based recommendation. *ACM Transactions on Information Systems*, *40*(4), 1–31.

Pan, Z., Cai, F., Chen, W., Chen, H., & de Rijke, M. (2020). Star graph neural networks for session-based recommendation. In *Proceedings of the 29th ACM international conference on information and knowledge management* (pp. 1195–1204). ACM.

Qiu, R., Li, J., Huang, Z., & Yin, H. (2019). Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 579–588). ACM.

Ren, P., Chen, Z., Li, J., Ren, Z., Ma, J., & de Rijke, M. (2019). Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the thirty-third AAAI conference on artificial intelligence* (pp. 4806–4813). AAAI Press.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence* (pp. 452–461). AUAI Press.

Salakhutdinov, R., & Mnih, A. (2007). Probabilistic matrix factorization. In *Proceedings of the twenty-first annual conference on neural information processing systems* (pp. 1257–1264). Curran Associates, Inc..

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the tenth international world wide web conference* (pp. 285–295). ACM.

Shani, G., Heckerman, D., & Brafman, R. I. (2005). An MDP-based recommender system. *Journal of Machine Learning Research*, 1265–1295.

Song, J., Shen, H., Ou, Z., Zhang, J., Xiao, T., & Liang, S. (2019). ISLF: interest shift and latent factors combination model for session-based recommendation. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 5765–5771). ijcai.org.

Song, J., Xu, J., Zhou, R., Chen, L., Li, J., & Liu, C. (2021). CBML: A cluster-based meta-learning model for session-based recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 1713–1722). ACM.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1441–1450). ACM.

Tan, Y. K., Xu, X., & Liu, Y. (2016). Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 17–22). ACM.

Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., & Lian, D. (2022). A survey on session-based recommender systems. *ACM Computing Surveys*, *54*(7), 154:1–154:38.

Wang, J., Ding, K., Zhu, Z., & Caverlee, J. (2021). Session-based recommendation with hypergraph attention networks. In *Proceedings of the 2021 SIAM international conference on data mining* (pp. 82–90). SIAM.

Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., & Liu, W. (2018). Attention-based transactional context embedding for next-item recommendation. In *Proceedings of the thirty-second AAAI conference on artificial intelligence* (pp. 2532–2539). AAAI Press.

Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., & Orgun, M. A. (2019). Sequential recommender systems: Challenges, progress and prospects. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 6332–6338). ijcai.org.

Wang, S., Hu, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., & Cao, L. (2019). Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 3771–3777). ijcai.org.

Wang, Y., Liu, X., Ju, Y., Börner, K., Lin, J., Sun, C., & Si, L. (2021). Chinese e-romance: analyzing and visualizing 7.92 million alibaba valentine's day purchases. *Data and Information Management*, *5*(4), 363–371.

Wang, M., Ren, P., Mei, L., Chen, Z., Ma, J., & de Rijke, M. (2019). A collaborative session-based recommendation approach with parallel memory modules. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 345–354). ACM.

Wang, Z., Wei, W., Cong, G., Li, X., Mao, X., & Qiu, M. (2020). Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 169–178).

Waqas, M., Ahmad, N., & Wu, J. (2021). Adoption of e-services and quality of life among older consumers in china. *Data and Information Management*, *5*(1), 125–130.

Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., & Tan, T. (2019). Session-based recommendation with graph neural networks. In *Proceedings of the thirty-third AAAI conference on artificial intelligence* (pp. 346–353). AAAI Press.

Xia, X., Yin, H., Yu, J., Shao, Y., & Cui, L. (2021). Self-supervised graph co-training for session-based recommendation. In *Proceedings of the 30th ACM international conference on information and knowledge management* (pp. 2180–2190). ACM.

Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., & Zhang, X. (2021). Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of thirty-fifth AAAI conference on artificial intelligence* (pp. 4503–4511). AAAI Press.

Xu, C., Zhao, P., Liu, Y., Sheng, V. S., Xu, J., Zhuang, F., Fang, J., & Zhou, X. (2019). Graph contextualized self-attention network for session-based recommendation. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 3940–3946). ijcai.org.

Yakhchi, S., Behehsti, A., Ghafari, S.-m., Razzak, I., Orgun, M., & Elahi, M. (2022). A convolutional attention network for unifying general and sequential recommenders. *Information Processing & Management*, *59*(1), Article 102755.

Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on research and development in information retrieval* (pp. 729–732). ACM.

Yuan, J., Song, Z., Sun, M., Wang, X., & Zhao, W. X. (2021). Dual sparse attention network for session-based recommendation. In *Proceedings of thirty-fifth AAAI conference on artificial intelligence* (pp. 4635–4643). AAAI Press.

Zhang, X., Lin, H., Yang, L., Xu, B., Diao, Y., & Ren, L. (2021). Dual part-pooling attentive networks for session-based recommendation. *Neurocomputing*, *440*, 89–100.

Zheng, Y., Liu, S., Li, Z., & Wu, S. (2021). Cold-start sequential recommendation via meta learner. In *Proceedings of thirty-fifth AAAI conference on artificial intelligence* (pp. 1172–1180). AAAI Press.

Zheng, L., Noroozi, V., & Yu, P. S. (2017). Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the tenth ACM international conference on web search and data mining* (pp. 425–434). ACM.

Zhou, H., Tan, Q., Huang, X., Zhou, K., & Wang, X. (2021). Temporal augmented graph neural networks for session-based recommendations. In *Proceedings of the 44th International ACM SIGIR conference on research and development in information retrieval* (pp. 1798–1802). ACM.

Zimdars, A., Chickering, D. M., & Meek, C. (2001). Using temporal data for making recommendations. In *Proceedings of the 17th conference in uncertainty in artificial intelligence* (pp. 580–588). Morgan Kaufmann.