Deep User Multi-interest Network for Click-Through Rate Prediction

Ming Wu[1], Junqian Xing[2], and Shanxiong Chen[1(✉)]

[1] College of Computer and Information Science, Southwest University, Chongqing, China
csxpml@163.com
[2] Faculty of Engineering, University of Sydney, Sydney, NSW, Australia

Abstract. Click-through rate (CTR) prediction is widely used in recommendation systems. Accurately modeling user interest is the key to improve the performance of CTR prediction task. Existing methods pay attention to model user interest from a single perspective to reflect user preferences, ignoring user different interests in different aspects, thus limiting the expressive ability of user interest. In this paper, we propose a novel Deep User Multi-Interest Network (DUMIN) which designs Self-Interest Extraction Network (SIEN) and User-User Interest Extraction Network (UIEN) to capture user different interests. First, SIEN uses attention mechanism and sequential network to focus on different parts in self-interest. Meanwhile, an auxiliary loss network is used to bring extra supervision for model training. Next, UIEN adopts multi-headed self-attention mechanism to learn a unified interest representation for each user who interacted with the candidate item. Then, attention mechanism is introduced to adaptively aggregate these interest representations to obtain user-user interest, which reflects the collaborative filtering information among users. Extensive experimental results on public real-world datasets show that proposed DUMIN outperforms various state-of-the-art methods.

Keywords: Recommender system · CTR prediction · Multi-interest learning · Multi-headed attention mechanism · Deep learning · Auxiliary loss

1 Introduction

In modern recommendation systems, users behave diversely, including collecting and purchasing, which are subsequent behaviors preceded by the basic behavior: click. In cost per click (CPC) advertising systems, effective cost per mille (eCPM) is calculated by production of advertisement bid price and click-through rate (CTR), which is used to rank advertisements. The CTR requires model prediction, whose performance has direct influence on user experience and corporate profit. Hence, CTR prediction has attracted extensive research in industry and academia [4, 10, 17].

In recommended scene, users have a variety of click interests. For instance, users may click on completely unrelated items such as clothes and electronic devices at a same time in E-commerce. Therefore, aiming at CTR prediction task, accurately capturing user interests in extenso is the key to improve model performance. With the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
G. Memmi et al. (Eds.): KSEM 2022, LNAI 13369, pp. 57–69, 2022.
https://doi.org/10.1007/978-3-031-10986-7_5

development of deep learning, some models based on DNN have been proposed to capture user interest. For instance, DIN [19] believes user behaviors contain a variety of interests, utilizing an attention mechanism to adaptively learn user interest in candidate item. However, it does not address the dependence of interests and lacks the ability of capturing interest transfer with time shift. DIEN [18] utilizes GRU [3] and attention mechanism respectively to model the representation and evolution of interest. While DIEN neglects to capture similarities between users to reflect user preferences. Since the similarity among users can reflect the target user preferences [1], DUMN [8] first learns a unified interest representation for target user and relevant users (**that is, users who have interacted with the candidate item**), and then aggregates these interests according to their similarities. DUMN has further enriched user interest by incorporating relevant information among users. However, it independently learns the interest of each user, and has not established a similar mapping between target user and relevant users, thus failing to fully exploit the collaborative filtering information among users. Most existing models simply put a single perspective into consideration of user interest, while user interests are diverse. Capturing multiple interests in different aspects is of significance to user interest representation.

Based on the observations above, this paper proposes a novel Deep User Multi-Interest Network (DUMIN) that designing Self-Interest Extraction Network (SIEN) and User-User Interest Extraction Network (UIEN) to process multiple different interests to predict CTR. Firstly, in SIEN, Direct Interest Extraction Layer adaptively extracts direct interest by using attention mechanism to measure the correlation between user behavior and the candidate item. Meanwhile, Evolutionary Interest Extraction Layer explicitly extracts the potential interest at each moment from user behavior and regards the last potential interest as evolutionary interest. Next, in UIEN, User Interest Representation Layer uses a multi-head self-attention mechanism to establish a similar interest mapping between target user and relevant users, thus amplifying the collaborative filtering signals among users. User Interest Match Layer adaptly matches interests between target user and each relevant user to aggregate similar interests from User Interest Representation Layer. Finally, a variety of different interests extracted by SIEN and UIEN, candidate item and context are concatenated and fed into Multilayer Perceptron (MLP) to predict CTR. The main contribution of this paper are summarized as follows:

- We point out the importance of multi-interest modeling user interest representation, and propose a novel model called DUMIN that extracts multiple user interests modeling CTR prediction task.
- We utilize a multi-head self-attention mechanism to learn the similar interest between the target user and relevant users in different representation subspaces, amplifying the collaborative filtering singals among users.
- Extensive experiments on public real-world datasets prove that the proposed DUMIN has significant advantages over several state-of-the-art baselines, and our code is publicly available for reproducible[1].

In the following part of this paper, we first review the related work in Sect. 2. Then, we introduce our model in detail in Sect. 3. Next, we conduct extensive experiments to verify the effectiveness of our model in Sect. 4. Finally, the conclusions and future outlooks are presented in Sect. 5.

[1] https://github.com/MrnotRight/DUMIN.

2 Related Works

With the widespread application of deep learning [4,20,21], deep learning models have been proved to possess great advantages in feature interaction and combination. Traditional linear models, such as LR [12], FM [14], etc., use linear combination and matrix factorization to model CTR prediction task, which pay little attention to capture high-order feature interactions and nonlinear features, and limit the expression ability of model. Wide&Deep [2] combines linear combination and neural network cross features to enhance model expression ability, while the wide part still needs manual designed. DeepFM [5] supersedes the wide part with FM on the basis of Wide&Deep, which avoids feature engineering and improves the ability of capturing second-order features. Limited by combinatorial explosion, FM is difficult to extend to high-order forms. NFM [6] combines FM with DNN to model high-order feature. Besides, PNN [13] introduces outer product and inner product operations to specifically enhance the interaction of different features, making it easier for the model to capture cross-information. However, these methods directly model feature interactions, and rarely pay attention to the abundant interest patterns implied in user's historical behavior data.

In order to dig out the rich information in user's historical behavior data, GRU4REC [7] applies GRU to model the evolution of items in user behavior, while it not pay attention to learn the user interest representation. The attention mechanism is introduced in DIN to learn interest representation from user's historical data, and it's application adequately captures the diversity of user interest. DIEN believes that the user interest migrate with temporal variation. Therefore, DIEN chooses GRU to extract the interests in user behaviors, and adaptively model the evolution of user interest in the candidate item by the attention mechanism. To model user interest representation in multiple subspaces, DMIN [16] introduces a multi-head self-attention mechanism to model the interests in different subspaces. DMR [11] designs user-item network and item-item network to employ different relevances to derive user interests in candidate item. The relevance among users can strengthen the collaborative filtering signals, which are able to learn accurate personalized preferences for users [1]. DUMN employs the correlation between users to improve the accuracy of interest representation learning, thereby improving the performance of the model. Although these methods fully exploit the potential interests of user historical behavior data, they rarely focus on enriching user interest representation modeling from multiple perspectives.

The works mentioned above improve the CTR prediction task through different modeling approaches. However, none of them attempted to learn user multiple interests from different aspects. In a real recommendation scenario, users often have a variety of different interests. Motivated by this, we refer to the interests learned from the user historical behavior data as self-interest, and those from relevant users as user-user interest. Moreover, the self-interest is subdivided into direct interest in the candidate item and evolutionary interest in user behavior. In DUMIN, on the one hand, we extract direct interest and evolutionary interest separately to form self-interest. On the other hand, we use self-interest as query to model the interest similarities between the target user and relevant users in different representation subspaces through the multi-head self-attention mechanism. In this way, we learn a variety of different interests for users, and capture the similarity relationship between self-interest and user-user interest, thereby

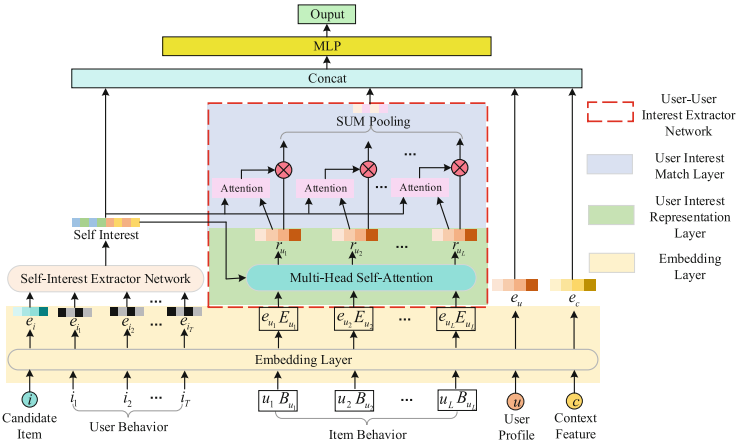

Fig. 1. DUMIN framework. Fed the candidate item and the target user behavior into SIEN to extract self-interest, and then fed self-interest and the candidate item behavior into UIEN to extract user-user interest. These two interests are concatenated to form the user interest representation for subsequent prediction of CTR via MLP.

amplifying the collaborative filtering signals among users, making the user interest representation more abundant and more accurate.

3 The Proposed Model

3.1 Preliminaries

There are five categories of features in DUMIN: *User Profile*, *Candidate Item*, *Context*, *User Behavior* and *Item Behavior*. Among them, *User Profile* contains *user ID*; *Candidate Item* contains *item ID*,*category ID*, etc.; Features in *Context* are *time*, *location* and so on. *User Behavior* and *Item Behavior* are defined as follows:

User Behavior. *Given a user* u*, the user behavior* $\mathbf{B}_u$ *is a time-sorted list of items that user* u *has interacted with. Each item has features such as item ID, category ID, etc.* $\mathbf{B}_u$ *is formalized as* $\mathbf{B}_u = [i_1, i_2, ..., i_{T_u}]$*, in which* i_t *is the t-th interacted item, and* T_u *is the length of* $\mathbf{B}_u$.

Item Behavior. *Given an item* i*, the item behavior* $\mathbf{N}_i$ *is a time-sorted list of users who has interacted with item* i*. Each user contains features such as user ID, user behavior, etc.* $\mathbf{N}_i$ *is formalized as* $\mathbf{N}_i = [(u_1, \mathbf{B}_{u_1}), (u_2, \mathbf{B}_{u_2}), ..., (u_{L_i}, \mathbf{B}_{u_{L_i}})]$*, in which* u_t is the t-th interacted user, $\mathbf{B}_{u_t}$ is the user behavior of u_t, and L_i *is the length of* $\mathbf{N}_i$.

3.2 Embedding

The category features used in DUMIN need to be encoded to low-dimensional dense features that facilitate deep neural network learning. This is widely implemented in

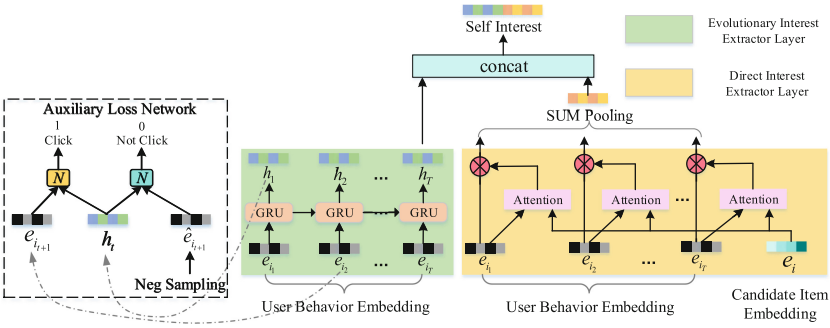

Fig. 2. The architecture of Self-Interest Extractor Network. The green part employs GRU and the auxiliary loss network to extract evolutionary interest from user behavior, and the yellow part employs the attention mechanism to extract user direct interest in candidate item. Concatenate these two interests to form self-interest representation. (Color figure online)

CTR prediction models based on deep learning [11,19]. Target user u, candidate item i, context c, user behavior $\mathbf{B}_u$ and item behavior $\mathbf{N}_i$ go through the embedding layer to obtain embedding vectors $\mathbf{e}_u$, $\mathbf{e}_i$, $\mathbf{e}_c$, $\mathbf{E}_u$ and $\mathbf{X}_i$, where $\mathbf{E}_u = [\mathbf{e}_{i_1}, \mathbf{e}_{i_2}, ..., \mathbf{e}_{i_{T_u}}]$, $\mathbf{X}_i = [(\mathbf{e}_{u_1}, \mathbf{E}_{u_1}), (\mathbf{e}_{u_2}, \mathbf{E}_{u_2}), ..., (\mathbf{e}_{u_{L_i}}, \mathbf{E}_{u_{L_i}})]$.

3.3 Self-Interest Extractor Network

In this subsection, we will introduce the details of SIEN in DUMIN. As shown in Fig. 2, SIEN captures the user self-interest from two different aspects. Direct Interest Extractor Layer extracts interest based on the correlation between user behavior and the candidate item; Evolutionary Interest Extractor Layer only focus on interest evolution in user behavior. Concatenate the two interests to obtain the self-interest for subsequent usage.

Direct Interest Extractor Layer. Direct Interest Extractor Layer measures the correlation between user behavior and the candidate item through the attention mechanism, reflecting the user preference in the candidate item. In this paper, we adopts the interest extraction method that is used in DIN [19], and the formulas are as follows:

$$\hat{\alpha_t} = \mathbf{Z}_d^T \sigma(\mathbf{W}_{d_1}\mathbf{e}_{i_t} + \mathbf{W}_{d_2}\mathbf{e}_i + \mathbf{b}_d) \tag{1}$$

$$\alpha_t = \frac{exp(\hat{\alpha_t})}{\Sigma_{j=1}^{T_u} exp(\hat{\alpha_j})}, \quad \mathbf{s}_d = \sum_{j=1}^{T_u} \alpha_j \mathbf{e}_{i_j} \tag{2}$$

where $\mathbf{e}_{i_t}, \mathbf{e}_i \in \mathbb{R}^D$ are the embedding vectors of the t-th interacted item in the target user behavior and the candidate item, respectively. $\mathbf{W}_{d_1}, \mathbf{W}_{d_2} \in \mathbb{R}^{H_d \times D}$, and $\mathbf{Z}_d, \mathbf{b}_d \in \mathbb{R}^{H_d}$ are network learning parameters, α_t is the normalized attention weight for the t-th interacted item, T_u is the length of target user behavior, σ is sigmoid activation function. $\mathbf{s}_d$ is the target user direct interest in the candidate item, formed by sum pooling the embedding vectors of items in user behavior via the attention weight.

Evolution Interest Extractor Layer. It has been proposed in DIEN that user interest evolution over time [18]. Inspired by this, Evolutionary Interest Extractor Layer also utilizes GRU to extract the interest state at each moment in user behavior. Unlike in DIEN, we do not pay attention to the correlation between the interest state and the candidate item. We merely care about capturing an evolutionary interest completely independent of the candidate item, which directly reflects the user preference when the user behavior has evolved to the final moment. The GRU models evolutionary interest can be formulated as:

$$\mathbf{u}_t = \sigma(\mathbf{W}_u \mathbf{e}_{i_t} + \mathbf{V}_u \mathbf{h}_{t-1} + \mathbf{b}_u) \tag{3}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{e}_{i_t} + \mathbf{V}_r \mathbf{h}_{t-1} + \mathbf{b}_r) \tag{4}$$

$$\hat{\mathbf{h}}_t = tanh(\mathbf{W}_h \mathbf{e}_{i_t} + \mathbf{r}_t \circ \mathbf{V}_h \mathbf{h}_{t-1} + \mathbf{b}_h) \tag{5}$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{u}_t) \circ \mathbf{h}_{t-1} + \mathbf{u}_t \circ \hat{\mathbf{h}}_t \tag{6}$$

where $\circ$ is element-wise product, $\mathbf{W}_u, \mathbf{W}_r, \mathbf{W}_h \in \mathbb{R}^{E \times D}$, $\mathbf{V}_u, \mathbf{V}_r, \mathbf{V}_h \in \mathbb{R}^{E \times E}$, $\mathbf{b}_u, \mathbf{b}_r, \mathbf{b}_h \in \mathbb{R}^E$ are learning parameters in GRU, $\mathbf{h}_t \in \mathbb{R}^E$ is t-th hidden states, E is the hidden dimension. For maximize the correlation between evolutionary interest and item, this paper introduces auxiliary loss network to supervise the learning of them. To construct the auxiliary loss network input samples, for each hidden state in the GRU, use the next clicked item in user behavior as a positive example, and randomly sample one item from all items as a negative example. Auxiliary loss can be formulated as:

$$\begin{aligned} L_{aux} = -\frac{1}{N}(\sum_{i=1}^{N} \sum_{t} & log\varphi(concat(\mathbf{h}_t, \mathbf{e}_{i_{t+1}})) \\ & + log(1 - \varphi(concat(\mathbf{h}_t, \hat{\mathbf{e}}_{i_{t+1}})))) \end{aligned} \tag{7}$$

where N is size of the training set, $\hat{\mathbf{e}}_{i_t}$ represents the embedding of t-th unclicked item generated by random negative sampling. φ is the auxiliary loss network whose output layer activation function is sigmoid. Regard the final hidden state in the GRU as the evolutionary interest, and concatenate it with direct interest to form self-interest of the target user. The formulation is listed as follows:

$$\mathbf{s}_u = concat(\mathbf{s}_d, \mathbf{h}_{T_u}) \tag{8}$$

where $\mathbf{s}_d, \mathbf{h}_{T_u}$ and $\mathbf{s}_u$ are the direct interest, evolutionary interest and self-interest of the target user u, respectively.

3.4 User-User Interest Extractor Network

The architecture of User-User Interest Extractor Network (UIEN) is shown in Fig. 1. First, the self-interest extracted from the SIEN is fed into User Interest Representation Layer to learn the unified similar interests between the target user and relevant users. Then, in User Interest Match Layer, all similar interests are aggregated by user-to-user relevance. In the next two subsections, we will introduce UIEN in detail.

User Interest Representation Layer. In User Interest Representation Layer, the objective is to learn a unified interest representation for each relevant user in the candidate item behavior. Existing methods directly measure the item-item correlation between user behavior and the candidate item to extract interest representation, which focus on the correlation of user and item. However, they are not suitable reflections of the correlation among users. In this paper, we utilize a multi-head self-attention mechanism, employing the self-interest as the query to capture the similarities between the target user and each relevant user. Note that the query is only generated by the self-interest. For the relevant user u_m, we can formalize the calculation in the User Interest Representation Layer as follows:

$$\mathbf{H}_{u_m} = concat(\mathbf{s}_u, \mathbf{E}_{u_m}) \tag{9}$$

$$\mathbf{Q} = \mathbf{W}^Q \mathbf{s}_u, \quad \mathbf{K} = \mathbf{W}^K \mathbf{H}_{u_m}, \quad \mathbf{V} = \mathbf{W}^V \mathbf{H}_{u_m} \tag{10}$$

where $\mathbf{E}_{u_m}$ is the user behavior embedding of u_m, $\mathbf{W}^Q$, $\mathbf{W}^K$ and $\mathbf{W}^V$ are projection matrices. $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are query, key and value, respectively. Self-attention is calculated as:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \tag{11}$$

d_k is the dimension of query, key, and value. The similar interest representation in j-th subspaces is calculated as:

$$\mathbf{head}_j = Attention(\mathbf{W}_j^Q \mathbf{s}_u, \mathbf{W}_j^K \mathbf{H}_{u_m}, \mathbf{W}_j^V \mathbf{H}_{u_m}) \tag{12}$$

$\mathbf{W}_j^Q \in \mathbb{R}^{d_k \times (E+D)}$, $\mathbf{W}_j^K, \mathbf{W}_j^V \in \mathbb{R}^{d_k \times (E+(T_u+1)*D)}$ are weighting matrices for the j-th head. For capturing the similarities in different representation subspaces [15], we concatenate the multi-head calculation results as a unified interest representation of each relevant user, which is formalized as:

$$\mathbf{r}_{u_m} = concat(\mathbf{e}_{u_m}, \mathbf{head}_1, \mathbf{head}_2, ..., \mathbf{head}_N) \tag{13}$$

$\mathbf{e}_{u_m}$ is the embedding of user u_m, N is the number of heads.

User Interest Match Layer. In User Interest Match Layer, the target is to learn adaptive weights for each similar interest of relevant users, so as to aggregate these similar interests by learned weights to obtain the final user-user interest. Thus, the attention mechanism is implemented to calculate the similarity weights as follows:

$$\hat{\eta}_m = \mathbf{V}_a^T \sigma(\mathbf{W}_{a_1} \mathbf{s}_u + \mathbf{W}_{a_2} \mathbf{r}_{u_m} + \mathbf{b}_a) \tag{14}$$

$$\eta_m = \frac{exp(\hat{\eta}_m)}{\sum_{j=1}^{L_i} exp(\hat{\eta}_j)}, \quad \mathbf{r}_u = \sum_{j=1}^{L_i} \eta_j \mathbf{r}_{u_j} \tag{15}$$

where $\mathbf{W}_{a_1} \in \mathbb{R}^{H \times (E+D)}$, $\mathbf{W}_{a_2} \in \mathbb{R}^{H \times (D+N*d_k)}$, $\mathbf{V}_a, \mathbf{b}_a \in \mathbb{R}^H$ are learning parameters. η_m represents the similarity weight between the target user and relevant user. L_i is item behavior length of the candidate item i. $\mathbf{r}_u$ represents the user-user interest of target user u, which is derived by sum pooling the similarity weight between each relevant user and u.

3.5 Prediction and Optimization Objective

Prediction. The vector representation of self-interest, user-user interest, candidate item, user profile, context are concatenated. Then fed them into MLP for predicting the click probability of target user on candidate item. Formally:

$$\mathbf{input} = concat(\mathbf{s}_u, \mathbf{r}_u, \mathbf{e}_u, \mathbf{e}_i, \mathbf{e}_c) \tag{16}$$

$$p = MLP(\mathbf{input}) \tag{17}$$

The activation function used in the hidden layer of MLP is PReLU, and the output layer of that is sigmoid activation function for normalizing the click probability from 0 to 1.
Optimization Objective. We adopt the most commonly used negative log-likelihood target loss for CTR model training, which is formalized as follows:

$$L_{target} = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(p_i) + (1 - y_i) log(1 - p_i)) \tag{18}$$

where N is the size of the training set, p_i represents the predicted CTR of the i-th sample, $y_i \in \{0, 1\}$ represents the click label. Considering with the auxiliary loss mentioned above, the final optimization objective of our model can be represent as:

$$Loss = L_{target} + \beta \cdot L_{aux} \tag{19}$$

where β is a hyperparameter, which is to balance the weight of the auxiliary loss and the target loss.

4 Experiments

In this section, firstly, we will compare DUMIN with several state-of-the-art methods on public real-world datasets to verify the effectiveness of our model. Then, an ablation study is designed to explore the influence of each part in DUMIN. Finally, the effects of some hyperparameters is analyzed.

Table 1. The statistics of the three datasets

Dataset	#Users	#Items	#Categories	#Reviews	#Samples
Beauty	22363	12101	221	198502	352278
Sports	35598	18357	1073	296337	521478
Grocery	14681	8713	129	151254	273146

Table 2. The bolded result is the best of all methods, and the underlined result is the best result of baselines.

Model	Beauty		Sports		Grocery	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
SVD++	0.6867	0.6831	0.7070	0.6347	0.6385	0.8306
Wide&Deep	0.8064	0.5516	0.7926	0.5488	0.6823	0.6634
PNN	0.8081	0.5509	0.8012	0.5408	0.7033	0.6324
DIN	0.8178	0.5375	0.8074	0.5334	0.7053	0.6284
GRU4Rec	0.8416	0.4923	0.8136	0.5263	0.7949	0.5360
DIEN	0.8530	0.4811	0.8225	0.5167	0.7875	0.5472
DUMN	0.8555	0.4796	0.8173	0.5227	0.8107	0.5159
DUMIN-AN[a]	0.8617	0.4603	0.8244	0.5132	0.8053	0.5216
DUMIN	**0.8721**	**0.4429**	**0.8325**	**0.5041**	**0.8225**	**0.5035**
Improvement	**+1.94%**	**−7.65%**	**+1.22%**	**−2.44%**	**+1.46%**	**−2.40%**

[a] DUMIN without auxiliary loss network.

4.1 Datasets

We conduct experiments on three public real-word subsets of Beauty, Sports, and Grocery in the Amazon dataset[2]. Each dataset contains product reviews and metadata. For the CTR prediction task, we regard all product reviews as positive samples of click. First, sort the product reviews in ascending order according to the timestamp to construct user behaviors and item behaviors. Then, randomly select another item from the unclicked items to replace the item in each review to construct the negative samples. Finally, according to the timestamp, split the former 85% part of the entire dataset as the training set, and the remaining 15% as the testing set. The statistics of datasets are summarized in Table 1.

4.2 Competitors and Parameter Settings

Competitors. We compared DUMIN with the following state-of-the-art methods to evaluate the effectiveness of it:

- **SVD++** [9]. It is a matrix factorization method that combines domain information. In our experiments, we use item behavior as domain information.
- **Wide&Deep** [2]. Wide&Deep combines wide and deep parts for linear combination features and cross features, respectively.
- **PNN** [13]. PNN introduces outer product and inner product in the product layer to learn abundant feature interactions.
- **DIN** [19]. DIN implements the attention mechanism to adaptively learn diverse interest representations in user behavior.

[2] http://jmcauley.ucsd.edu/data/amazon/.

- **GRU4Rec** [7]. GRU4Rec utilizes GRU to model user behavior. We develop it to model item behavior as well.
- **DIEN** [18]. DIEN uses a two-layer GRU and attention mechanism to model the extraction and evolution of user interests.
- **DUMN** [8]. DUMN first learns unified representations for users, then measures the user-to-user relevance among users.

The public codes[3] for these baselines are provided in the previous work [8]. What should be noted is that DIEN implemented in it does not use the auxiliary loss network. For fairness, we implement DIEN with the auxiliary loss network.

Parameter Settings. In the experiment, we follow the parameter settings in [8]. We set the embedding vector dimensions of category features as 18. The maximum length of user behavior and item behavior are set as 20 and 25, respectively. Employ Adam optimizer and set the batch size to 128 and the learning rate to 0.001. Furthermore, we set auxiliary loss coefficient and margin to 1, and the number of heads in multi-headed self-attention to 6.

4.3 Experimental Results

Area Under ROC Curve (AUC) and Logloss are utilized as evaluation indicators, which are widely used to evaluate the performance of the CTR prediction models [5,8].

We repeat all experiments 5 times and record the average results. The comparison results on public real-world datasets are shown in Table 2. Compared with the best baseline, the average relative improvement achieved by DUMIN in AUC and Logloss is 1.54% and 4.16% respectively, which is particularly significant in the CTR prediction task. Observing the experimental results, first of all, SVD++ has achieved the worst performance due to its inability to capture nonlinear and high-order features. Secondly, Wide&Deep and PNN introduce a neural network, which is the reason of a huge improvement compared with SVD++, while PNN designs a product layer that enriches the interaction of features and achieved better performance than Wide&Deep. Thirdly, compared to Wide&Deep and PNN, the introduction of the attention mechanism allows DIN to model the CTR prediction task more accurately. Fourthly, GRU4Rec and DIEN are superior to DIN because the former focus on both user behavior and item behavior, while the latter captures the interests evolution in user behavior. The reason for the different outperformances between GRU4Rec and DIEN on the different datasets is that the time-dependent method of DIEN modeling interest representation is more advanced, while GRU4Rec introduces item behavior and captures more useful information. Fifthly, DUMN has achieved the best performance on the Beauty and Grocery datasets compared to other baselines, which reflects that the relevant users interests are particularly effective for CTR prediction. Finally, DUMIN achieves the best performance on all datasets compared with all baselines, which indicates the effectiveness of multi-interest modeling user interests. It is worth mentioning that, compared with DUMN, DUMIN not only extracts self-interest in user behavior, but also adopts

[3] https://github.com/hzzai/DUMN.

Table 3. Results of ablation study on the public real-word datasets. The bolded scores are the original model performance. ↓ indicates the most conspicuously declined score in each dataset.

Model	Beauty		Sports		Grocery	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
DUMIN-AN[a]	0.8617	0.4603	0.8244	0.5132	0.8053	0.5216
DUMIN-DI[b]	0.8700	0.4458	0.8269	0.5092	0.8226	0.5046
DUMIN-EI[c]	0.8585	0.4667	0.8204↓	0.5169↓	0.8012	0.5248
DUMIN-UI[d]	0.8557↓	0.4704↓	0.8213	0.5167	0.7592↓	0.5705↓
DUMIN	**0.8721**	**0.4429**	**0.8325**	**0.5041**	**0.8225**	**0.5035**

[a] DUMIN without auxiliary loss network.
[b] DUMIN without direct interest.
[c] DUMIN without evolutionary interest.
[d] DUMIN without user-user interest.

self-interest as a query to establish a similarity mapping between self-interest and user-user interest in item behavior, enhancing collaborative filtering signals between interest representations, which has resulted in huge progress. Moreover, we dropped auxiliary loss network to train DUMIN-AN and got a worse performance compared with DUMIN, which proves the superiority of the auxiliary loss network to enhance correlation between interest and item.

4.4 Ablation Study

In this section, we conducted an ablation study to explore the effectiveness of the various components in DUMIN. The experimental results are shown in Table 3. The following facts can be observed: First of all, DUMIN outperforms DUMIN-AN, which verifies the importance of the auxiliary loss network. Next, the performance of DUMIN-EI is worse than DUMIN-AN because after removing the evolutionary interest, the extra supervision provided by the auxiliary loss network is meaningless. Finally, the performance of DUMIN-DI, DUMIN-EI and DUMIN-UI are all worse than DUMIN, which reflects the effectiveness of our designed different components to capture multiple user interests in different aspects to accurately model the interest representation. Moreover, the significant drop in the performance of DUMIN-UI also verifies the importance of similar interests among users to the CTR prediction task.

4.5 Parameter Analysis

As some hyperparameters in DUMIN have impact on the experimental results, we conducted extensive experiments to explore the effects of these hyperparameters. The experimental results are shown above in Fig. 3, in which we discover: (1) When the maximum length of the item behavior L_{max} is between 25 and 30, the DUMIN performance is the best. When L_{max} increases in the range of 5 to 25, the performance becomes better accordingly. When L_{max} increases in the range of 35 to 50, however, the performance gets worse gradually. It is obvious that when L_{max} is set too low or

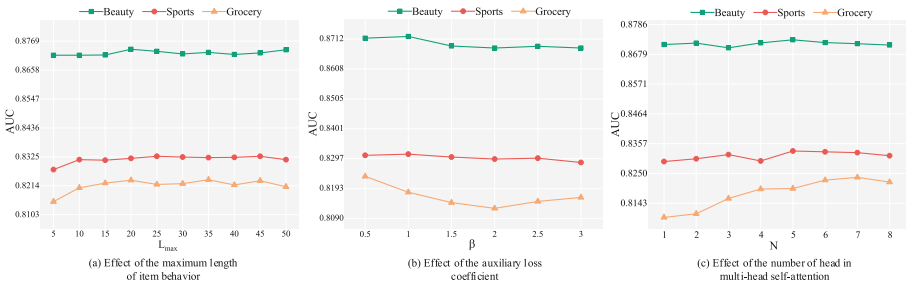

Fig. 3. Parameter analysis. The effect of different hyperparameters in DUMIN on Beauty dataset

too high, the performance will deteriorate, which indicates that a suitable number of relevant users is conducive to learn user-user interest, while too many or few relevant users could affect the learning of accurate representation of user-user interest. (2) When the auxiliary loss coefficient β in the range of 0.5 to 1.0, DUMIN performs best. When β grows bigger, however, the performance gradually decreases. This suggests that it is of positive significance to increase the proportion of the auxiliary loss in Eq.(19), while too high proportion will be detrimental to network parameter optimization. (3) DUMIN achieves the best performance when N is 5 or 6. From an overall point of view, the performance of DUMIN keeps the same trend with N, which indicates that increasing the number of heads in the multi-head self-attention helps to utilize the properties of similar abilities in different subspaces.

5 Conclusions

This paper proposed a novel Deep User Multi-Interest Network (DUMIN) from a multi-interest perspective to accurately model diverse user interest representations. DUMIN not only focuses on different interests in users' historical behavioral data, but also captures the similar interest among users. Moreover, the introduction of the auxiliary loss network enhances the correlation between interest and item, and makes a better interest representation be learned. In the future, we will explore more effective interest extraction methods to improve the accuracy of CTR prediction task.

References

1. Bellogin, A., Castells, P., Cantador, I.: Neighbor selection and weighting in user-based collaborative filtering: a performance prediction approach. ACM Trans. Web **8**(2), 1–30 (2014)
2. Cheng, H.T., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10 (2016)
3. Chung, J., Gulcehre, C., Cho, K.H., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
4. Covington, P., Adams, J., Sargin, E.: Deep neural networks for Youtube recommendations. In: Proceedings of the 10th Conference on Recommender Systems, pp. 191–198 (2016)

5. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: DeepFM: a factorization-machine based neural network for CTR prediction. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 1725–1731 (2017)
6. He, X., Chua, T.S.: Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International Conference on Research on Development in Information Retrieval, pp. 355–364 (2017)
7. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: Proceedings of the 4th International Conference on Learning Representations (2016)
8. Huang, Z., Tao, M., Zhang, B.: Deep user match network for click-through rate prediction. In: Proceedings of the 44th International Conference on Research and Development in Information Retrieval, pp. 1890–1894 (2021)
9. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining, pp. 426–434 (2008)
10. Li, X., Wang, C., Tong, B., Tan, J., Zeng, X., Zhuang, T.: Deep time-aware item evolution network for click-through rate prediction. In: Proceedings of the 29th International Conference on Information and Knowledge Management, pp. 785–794 (2020)
11. Lyu, Z., Dong, Y., Huo, C., Ren, W.: Deep match to rank model for personalized click-through rate prediction. In: Proceedings of the 34th Conference on Artificial Intelligence, pp. 156–163 (2020)
12. McMahan, H.B., et al.: Ad click prediction: a view from the trenches. In: Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining, pp. 1222–1230 (2013)
13. Qu, Y., et al.: Product based neural networks for user response prediction. In: Proceedings of the 16th International Conference on Data Mining, pp. 1149–1154 (2016)
14. Rendle, S.: Factorization machines. In: Proceedings of the 10th International Conference on Data Mining, pp. 995–1000 (2010)
15. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
16. Xiao, Z., Yang, L., Jiang, W., Wei, Y., Hu, Y., Wang, H.: Deep multi-interest network for click-through rate prediction. In: Proceedings of the 29th International Conference on Information and Knowledge Management, pp. 2265–2268 (2020)
17. Xu, Z., et al.: Agile and accurate CTR prediction model training for massive-scale online advertising systems. In: Proceedings of the 2021 International Conference on Management of Data, pp. 2404–2409 (2021)
18. Zhou, G., et al.: Deep interest evolution network for click-through rate prediction. In: Proceedings of the 33th Conference on Artificial Intelligence, pp. 5941–5948 (2019)
19. Zhou, G., et al.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th International Conference on Knowledge Discovery and Data Mining, pp. 1059–1068 (2018)
20. Qiu, H., Zheng, Q., Msahli, M., Memmi, G., Qiu, M., Lu, J.: Topological graph convolutional network-based urban traffic flow and density prediction. IEEE Trans. Intell. Transp. Syst. **22**(7), 4560–4569 (2020)
21. Cao, W., Yang, P., Ming, Z., Cai, S., Zhang, J.: An improved fuzziness based random vector functional link network for liver disease detection. In: Proceedings of the 6th International Conference on Big Data Security on Cloud, pp. 42–48 (2020)