



# Graph-aware collaborative reasoning for click-through rate prediction

Xin Zhang<sup>1</sup> · Zengmao Wang<sup>1</sup> · Bo Du<sup>1</sup>

Received: 4 March 2022 / Revised: 27 March 2022 / Accepted: 4 April 2022 /  
Published online: 8 June 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Click-through rate prediction(CTR) is a critical task in an online advertising system. Recently, deep learning based architectures have brought great attention in Click-through rate prediction by learning the nonlinear interaction between feature embedding of users and items. However, these methods have the following issues: (1) The collaborative information between users and items could not be fully explored due to the static embedding with lookup-table technique. (2) The learning procedure lacks cognitive reasoning about what the users want to do and what they may need. To address the above challenges, we propose a graph aware collaborative reasoning method for CTR prediction which explores the collaborative information with graph and then predicts the users' behaviors with logical reasoning. Specifically, the graph is built by the common behaviors between users, and the embedding of users and items can be learned by propagating the collaborative information in the graph. Then with the collaborative embedding of users and items, two logical operations NOT and OR are adopted to integrate the embedding for logical reasoning with the neural networks. By learning the proposed architecture in an end-to-end manner, the logical behaviors of users in the behavior sequences can be learned efficiently. Extensive experiments on five real-world datasets show that the proposed method outperforms several state-of-the-art methods in CTR prediction.

**Keywords** click-through rate prediction · collaborative filtering · logical reasoning

---

Zengmao Wang and Bo Du contributed equally to this work.

---

This article belongs to the Topical Collection: *Special Issue on Spatiotemporal Data Management and Analytics for Recommend*

Guest Editors: Shuo Shang, Xiangliang Zhang and Panos Kalnis

---

✉ Zengmao Wang  
wangzengmao@whu.edu.cn

Xin Zhang  
zhangxin@whu.edu.cn

Bo Du  
dubo@whu.edu.cn

<sup>1</sup> School of Computer Science, Wuhan University, Wuhan 430000, China

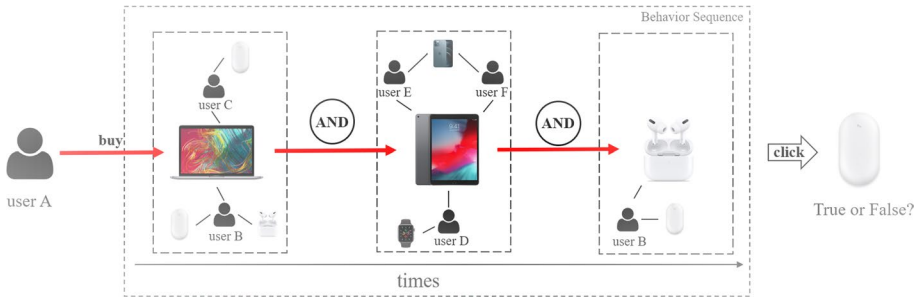
## 1 Introduction

Online advertising system has attracted considerable attention in industry and academia due to the high yield. A good online advertising system can generate tens of billions of revenue yearly in some companies, such as Google and Alibaba. Click-through rate (CTR) prediction, which is defined as predicting the probability of a user clicking at the target item, is an essential task in online advertising systems. Hence, CTR prediction has been a popular topic in data mining and machine learning fields. Moreover, numerous methods have been developed for CTR prediction.

One popular approach for CTR prediction is to learn the feature combinations. Factorization Machine(FM) [1], which is one of the most popular models, attempts to learn the embedding of each feature and predict CTR by linear combination of the first and second-order feature interaction. In recent years, deep learning has shown powerful capability to learn the high-order feature interaction. The methods with deep learning for CTR prediction, such as Wide&Deep [2] and DeepFM [3], are widely developed. Wide&Deep jointly trains logistic regression and deep neural networks to combine the strength of memorization and generalization for prediction. DeepFM replaces the linear model in Wide&Deep with FM for learning second-order feature interaction and the feature engineering can also be simplified. However, even though the rich information can be learned by informative feature interaction, it only focuses on user preference, because features usually imply the prior information regarding whether the user is interested in a target item. The historical behaviors of users, which not only reflects user preference but also their requirements [4], are not fully explored in these methods. Hence, the behaviors are important resources to improve the CTR prediction.

With the increasing availability and quality of users' behavior data, the use of this data for prediction becomes more necessary [5, 6]. Collaborative filtering (CF) approaches are adopted by assuming that the users with similar behaviors share similar preferences on items to explore the information in behaviors completely. In most collaborative filtering models, CF learns the embedding of users and items via historical interactions. One prominent model is Matrix Factorization(MF), which uses the inner product between the embedding of users and items to model user-item interaction. The embedding is represented with a dense vector. Many methods are developed on the basis of MF for CTR prediction. In [7], the CTR prediction is regarded as a special matrix completion problem. This study learns embedding of both users and items with logistic loss for binary data instead of square loss in MF and predicts the probability via inner product of embedding. The method in [8] proposed to use attention mechanism to learn the embedding of the users, and then the embeddings are incorporated into MF to capture the relevance between the user and the target item.

Although these methods are successfully applied in the advertising system, they mainly attempt to predict the items that may match the historical preference of users and user requirements are ineffectively learned. Hence, discovering the logical behaviors in the sequential behaviors is necessary for the inference of users intention. Figure 1 shows the logical behaviors. In Figure 1, when user A bought a macbook online, instead of browsing another laptop, he/she may prefer to click accessories such as air pods. This scenario indicates that the current decision of a user does not heavily depend on the similarity to his/her previous behaviors. Meanwhile, the collaborative information is not fully explored in the CF methods due to the lack of explicit connections, which may result in the matching of irrelevant items. For a good recommendation, the item that is recommended to the



**Figure 1** The illustration of logical behaviors. User A has bought macbook, ipad and airpods in the behavior sequence while user B, C, D, E and F have bought at least one of these three items. There exist many items that user A may need according to the user B, C, D, E and F. To infer which one is the requirement for A, the items that user A has bought are connected with logical operation AND( $\wedge$ ). With the logical inference, the prediction of that whether user A will click an item (such as mouse) can be learned

users should generally satisfy the preferences and the requirements of the users simultaneously. Hence, the collaborative capability via the user-item interaction also should be further improved.

A graph-aware collaborative reasoning for CTR prediction, termed as GACR, is proposed in this paper to address the above challenges. This method learns deep collaborative embedding based on user-item interaction graph and obtains the logical reasoning by integrating the logical operators into neural networks based on the embedding of users and items. In the proposed method, the collaborative information is explicitly presented by message passing and message propagation between the nodes in the graph. The information is learned recursively via the paths in the graph to guarantee that deep collaborative information can be endowed into the embedding of users and items. The interaction information between users and items is then learned on the basis of embedding with an interaction layer. The interaction information is transformed into logical representation based on the logic laws to capture the logical behaviors in the behavior sequence. A logical neural network is adopted to learn the logical behaviors with the logical representations. Therefore, the proposed method can learn the collaborative information and logical behaviors simultaneously with an end-to-end manner, and the preference and the requirements of the users can be captured effectively in CTR prediction.

The main contributions of the paper can be summarized as follows:

- The proposed GACR is the first work that attempts to learn the collaborative information and logical reasoning jointly in one network architecture for cognitive learning in CTR prediction.
- This paper highlights that the preferences and the requirements are both essential for CTR prediction, and the proposed method can capture both of them effectively.
- The results of extensive experiments on several real-world datasets demonstrate that our GACR outperforms several state-of-the-art models.

The following sections of this paper is organized as follows. Section 2 provides some recent studies that are related to the current work. In Section 3, the methodology of the proposed method is introduced in detail. Extensive experiments are reported in Section 4. The conclusion of the paper is shown in Section 5.

## 2 Related work

This section focuses on introducing the related works on both graph-based recommendation and CTR prediction.

### 2.1 Graph-based recommendation

Graph is a data structure with rich information and it had been applied in multiple areas [9]. ItemRank [10] designs a random-walk algorithm based on user-item interactions to extract user preferences in the user-item interaction graph. However, the algorithm is a model-based collaborative filtering(CF) method and lacks optimization capability. HOP-Rec [11] presents a unified and efficient framework that incorporates graph- and embedding-based methods. RecWalk [12], which is also based on random walk, utilizes the spectral properties in Markov chains to explore interaction graph effectively. Graph neural network is proven to be efficient in various task [13]. On the basis of paths in graph, RKGE [14] designs a recurrent network architecture for capturing user preference and then learns additional informative embedding to recommendation. MCRec [15] defines meta-path and leveraging co-attention mechanism to improve the representations of both entities and meta-paths. These path-based models are limited due to their sensitive effect on path selection despite their promising performance. The heterogeneous information network(HIN) has recently become a hot topic in graph-based recommendation. NIRec [16] emphasizes the interaction in HIN and proposes meta-path based module to learn interactive patterns. Finding appropriate paths requires large experiences and time, NGCF [17] focus on collaborative signal in graph structure and utilizes a propagation layer for extraction to alleviate the aforementioned problems. Similarly, KGAT [18] combines recursive embedding propagation and attention mechanism to extract high-order connectivity. An innovative model equipped with embedding propagation and logical reasoning is introduced in this paper to learn the comprehensive interest of users.

### 2.2 CTR prediction

CTR prediction is the core technology in recommendation, searching and advertising systems. Historically, logistic regression(LR) had been widespread in CTR prediction. However, using LR to predict CTR requires a huge workload for manual feature engineering. Facebook proposed [19] using gradient boost decision tree(GBDT) and LR to explore the combination of features automatically to reduce human consumption feature engineering. Despite the strengths in auto feature engineering, this approach did not consider the presence of many high dimensional sparse features in CTR prediction. Using decision tree to handle highly sparse data or online scenery is difficult.

FM [1] uses inner product between latent vectors to learn the weight of each second-order feature combination and provides predictions via linear aggregation of all first and second-order features. Some feature combinations have minimal influence on target task. AFM [20] introduced attention network to learn the influence of each second-order feature combination to enhance the expressiveness of the model. Similarly, FFM [21] referred to the concept of field to discriminate different importance of diverse feature cross.

FNN[22] was proposed on the basis of FM. FNN adopts FM to train the embedding in advance and then feeds the embedding into multi layer perceptron(MLP) to learn the high-order feature correlation. This two-phase structure with Embedding and MLP can be

summarized as Embedding&MLP model. PNN [23] introduced the product layer to extract additional complex feature interaction. Instead, NFM [24] proposed bi-interaction pooling layer to learn the feature interaction and enhance the expressiveness of the model.

Many recent studies [2, 3, 25] have shown that deep learning based CTR prediction models have achieved remarkable effectiveness. Wide&Deep Learning(WDL) [2] was first proposed by Google and employed in Google play application. WDL comprises a generalized linear module (wide part) and an MLP (deep part). WDL also combines the memorization of the wide part and the generalization of the deep part to model the user behavior. However, this approach is limited to designing feature cross manually. DeepFM [3] introduced the Deep Neural Network (DNN) based on FM model to improve the capability of information extraction and Deep Crossing [26] applies a deep residual network to learn cross features. Since DNN can only conduct implicit feature cross, Deep&Cross [27] designed Cross Net to replace FM. Thus the model can also learn bound-degree feature cross explicitly.

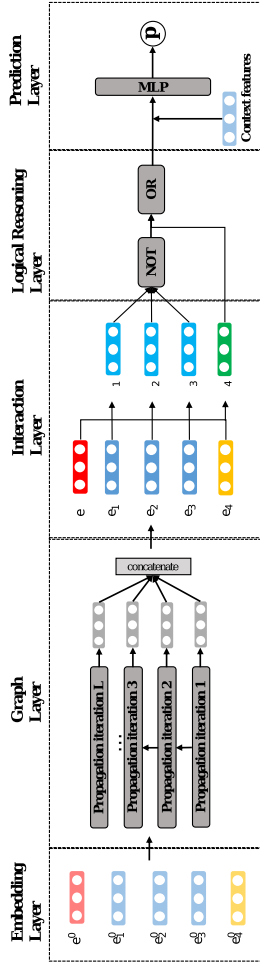
The attention mechanism learns a function to assign large weights to closely correlated items. This mechanism is originally proposed for the neural machine translation (NMT) [28], but has been widely used in diverse domains. Considering CTR prediction, DeepIntent [29] applies attention in the context and utilizes RNN to model text, and then learns the global hidden vector to allocate the weights of keys in each query. DIN [30] uses attention mechanism to learn the representation of historical behaviors of users. DIEN [31] adopts the interest extractor layer and interest evolving layer to learn the representation of user behaviors and capture the dynamic changes in user interests respectively. DSIN [32] leverages multiple historical sessions of users in behavior sequences with the attention mechanism to extract accurate interest representations of each session. In DMR [8], the model calculates the item-to-item similarities between user-target item interaction using the attention mechanism.

### 3 Methodology

The proposed GACR model will be comprehensively introduced in this section. First, an embedding layer is adopted to initialize the embedding of users and items with the lookup-table technique. Then a graph layer is adopted to propagate the collaborative information with the interaction graph and enrich the information in the embedding of users and items. Following the graph layer, an interaction layer is used to learn the interaction information between a user and an item from the embedding. A logical reasoning layer is used by propagating the interaction information into logical representation to learn the logical behaviors in the sequential behaviors. Finally, a prediction layer is adopted to predict the probability that user clicks target item. The architecture of GACR is shown in Figure 2.

#### 3.1 Embedding layer

One-hot encoding is a popular technique used to generate continuous feature representation for discrete data. This technique has been widely used in recommendation and CTR prediction. The user ID, item ID and context features are usually transformed into a sparse vector with one-hot encoding technique to learn the embedding of users and items in recommendation and CTR prediction, and then the sparse vector is transformed into a low dense vector via the lookup-table technique. Without loss of generality, the IDs of users



**Figure 2** Illustration of the GACR architecture. GACR takes the embedding of users, their historical items and the target item as inputs, and outputs the probability that the user will click the target item

and items are similarly transformed into dense vectors, which are then treated as the initialized embedding of users and items.

Mathematically,  $e_u^0 \in \mathbb{R}^d$  and  $e_i^0 \in \mathbb{R}^d$  are defined as the initialized embedding of user  $u$  and item  $i$  respectively, where  $d$  represents the embedding dimension. All the embeddings of users and items are organized as the embedding lookup table  $\mathbf{E}_u \in \mathbb{R}^{n_u \times d}$ ,  $\mathbf{E}_i \in \mathbb{R}^{n_i \times d}$  and  $\mathbf{E}_f \in \mathbb{R}^{n_f \times d}$ , where  $n_u$ ,  $n_i$  and  $n_f$  denotes the number of users, items and context features respectively. Thus, the embedding for each user, item and feature can be represented as follows:

$$\begin{aligned} e_u^0 &= t_u \times \mathbf{E}_u \\ e_i^0 &= t_i \times \mathbf{E}_i \\ e_f &= t_f \times \mathbf{E}_f \\ Cf &= \text{concat}(e_f), \quad \forall f \in F \end{aligned} \tag{1}$$

where  $t_u \in \mathbb{R}^{n_u}$  and  $t_i \in \mathbb{R}^{n_i}$  represent the one-hot encoding of user and item respectively.  $t_f \in \mathbb{R}^{n_u}$  represents the one-hot encoding of one field feature and  $Cf$  is the concatenated vector of context features set  $F$ . The user-item interaction graph is constructed on the basis of these initialized embedding to endow them with collaborative information where  $\text{concat}$  indicates the concatenate operation.

### 3.2 Graph layer

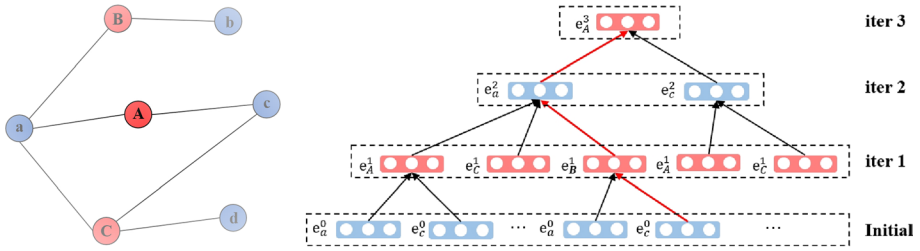
The collaborative information with graph is explicitly explored by following to capture the collaborative information between the users and items effectively [17]. The endowment of collaborative information to the embedding of users and items by the graph layer will be subsequently described. This approach mainly contains message passing and message propagation.

#### 3.2.1 Message passing

The message passing is achieved via the paths in the graph. For example, in a user-item interaction graph, as shown in Figure 1, if user A has purchased item a and this item is also purchased by user B, then a path  $A \rightarrow a \rightarrow B$  can be obtained. This condition indicates potential similarity between A and B. If B has purchased c, then the path  $A \rightarrow a \rightarrow B \rightarrow c$  implies that A may be interested in c. The message passing between one user-item pair can be then represented as:

$$m_{h \rightarrow t} = \frac{1}{\sqrt{D_h D_t}} (W_s e_h + b_s + W_i (e_h \odot e_t) + b_i) \tag{2}$$

where  $h$  and  $t$  in (2) respectively represent the head and tail nodes in an interaction path.  $m_{h \rightarrow t} \in \mathbb{R}^d$  represents the message passing from head node to tail node.  $D_h$  and  $D_t$  are the degrees of head and tail nodes respectively, and the degree of a node is the number of nodes that are connected with it in the graph.  $W_s, W_i \in \mathbb{R}^{d \times d}$  and  $b_s, b_i \in \mathbb{R}^d$  are trainable parameters for core message extraction. Notably, active users or popular items usually connect with an excessive number of nodes and may propagate numerous messages to neighbors. While the information from those cold users or items may be missed. Therefore, bias learning may appear.  $1/\sqrt{D_h D_t}$  in (2) is served as decay factor to solve this problem. The Element-wise product  $\odot$  in (2) facilitates the dependence of message passing on affinity between nodes according to [17].



**Figure 3** Overview of the graph layer. (left) is a sample user-item interaction graph and (right) is the illustration of message propagation with (a)

### 3.2.2 Message propagation

A node  $t$  in this graph can obtain various messages from its neighbor nodes. These messages are integrated together recursively for collaborative information learning, and the embedding of node  $t$  at each iteration can be calculated as:

$$\begin{aligned}
 e_t^l &= \sigma_1(W_s^l e_t^{l-1} + b_s^l + \sum_{h \in N_t} m_{h \rightarrow t}^l) \\
 m_{h \rightarrow t}^l &= \frac{1}{\sqrt{D_h D_t}} (W_h^l e_h^{l-1} + b_h^l + W_i^l (e_h^{l-1} \odot e_t^{l-1}) + b_i^l)
 \end{aligned}
 \tag{3}$$

where  $e_t^l$  represents the embedding of node  $t$  at  $l$ -th iteration, and  $N_t$  represents the set of nodes that are connected to  $t$  in graph.  $W_s^l, W_i^l \in \mathbb{R}^{d \times d}$  and  $b_s^l, b_i^l \in \mathbb{R}^d$  are trainable parameters at  $l$ -th iteration. LeakyReLU[33] is chosen as activation function  $\sigma_1$  in (3).

A simple example is shown in Figure 3 to understand the graph layer. Figure 3(a) shows a user-item interaction graph, and Figure 3(b) shows the message propagation based on the graph. Figure 3(a) reveals a path from the graph, such as  $b \rightarrow B \rightarrow a \rightarrow A$ . The collaborative information in this path is then learned With three iterations in Figure 3(b) to obtain the embedding of user  $A$ , that is,  $e_A^3$ . In recursive learning procedure,  $e_A^3$  mainly depends on the  $e_a^2$  and  $e_c^2$  in the second iteration. While  $e_a^2$  contains the messages from  $e_b^1$  and  $e_c^1$ .

With the message propagation in the graph, many embedding can be obtained from  $e^0$  to  $e^l$  for each user and item. The embedding at different iterations may contain different level collaborative information. Therefore, the final embedding for each node can be calculated as:

$$e = \text{concat}(e^0, e^1, \dots, e^l)
 \tag{4}$$

### 3.3 Interaction layer

The interaction layer is adopted on the basis of collaborative embedding of users and items to learn the interaction information between the users and items. Specifically, suppose a user  $u$  has interacted with several items  $i_1, i_2, \dots, i_n, i_{tar}$ , where  $n$  is the number of items which the user has clicked and  $i_{tar}$  is the latest item that the user has clicked. An MLP is used to obtain the interaction embedding between user  $u$  and the items that



the user has clicked. The concatenation of user embedding and the corresponding item embedding can be learned as follows:

$$\begin{aligned} I_i^u &= \text{concat}(e_u, e_i) \\ E_i^u &= W_2\sigma_2(W_1I_i^u + b_1) + b_2 \end{aligned} \tag{5}$$

where  $e_u, e_i$  are the embeddings of user  $u$  and item  $i$  respectively,  $I_i^u$  represents the interaction information between user  $u$  and item  $i$ .  $W_1, b_1, W_2, b_2$  are trainable parameters in MLP. Meanwhile, the ReLU is set as the activation function in MLP.

### 3.4 Logical reasoning layer

The behaviors can imply the potential and dynamic preference of the user and can be treated as an important feature to describe the user. With this feature, the use of logical learning is attempted to learn the user requirements. Through the following logical operation, a logical conditional statement can be transferred into logical representation with only disjunction  $\vee$  and negation  $\neg$ , that is,  $a \rightarrow b \iff \neg a \vee b$ . The problem can be transformed as follows:  $(a \wedge b \wedge c) \rightarrow t$  to  $\neg a \vee \neg b \vee \neg c \vee t$ .

Specifically, the logical reasoning layer comprises a recurrent neural module based on idea of distributed representation and neural symbolic framework [34, 35], to learn logical operation ‘‘OR’’ and a module to learn the logical operation ‘‘NOT’’. The logical representation for inferring can be easily defined with the interaction information learned from interaction layer. For example, if user  $u$  has bought items  $i_1, i_2, \dots, i_n$ , then the logical representation to predict whether  $u$  will click the item  $i_{tar}$  can be represented as:

$$\neg E_1^u \vee \neg E_2^u \vee \dots \vee \neg E_n^u \vee E_{tar}^u \tag{6}$$

The operation ‘‘NOT’’ is represented as a two-layer MLP, and can be formulated as follows:

$$\begin{aligned} \neg E_i^u &= \text{NOT}(E_i^u) \\ \text{NOT}(x) &= W_2\text{ReLU}(W_1x + b_1') + b_2' \end{aligned} \tag{7}$$

where  $W_1^n, W_2^n, b_1^n, b_2^n$  are trainable parameters in NOT module. An MLP module simulates ‘‘NOT’’. Therefore, the embedding of the interaction at negation, that is,  $\neg E_i^u$ , can be easily obtained. Equation (7) shows that logical operation  $\vee$  is continuous and requires two behaviors as input simultaneously. The OR network is designed as a recurrent MLP network. Mathematically, this network can be defined similarly to Recurrent Neural Network(RNN):

$$\begin{aligned} x_t &= E_t^u \\ h_t &= W_2^o\text{ReLU}(W_1^o(h_{t-1} + \neg x_t) + b_1^o) + b_2^o \end{aligned} \tag{8}$$

where  $W_1^o, W_2^o, b_1^o, b_2^o$  are trainable parameters and  $h_t$  represents a temporary state after  $t$  behaviors of a user. The initial hidden state  $h_1$  as  $\neg E_1^1$  is set and the negation of each interaction embedding is fed into the OR module recursively until the entire OR operation is learned. Thus,  $E_1^u \vee \neg E_2^u \vee \dots \vee \neg E_n^u$  can be finally transformed to  $h_n$ . The embedding of the logical behaviors in behavior sequence can then be calculated by:

$$\mathbf{O} = W_2^o\text{ReLU}(W_1^o(h_n + E_{tar}^u) + b_1^o) + b_2^o \tag{9}$$

Notably, the input is the embedding of the latest interaction  $E_{tar}^u$  instead of its negation  $\neg E_{tar}^u$ .  $\mathbf{O}$  represents the embedding of the logical behaviors. Similar to [35], an extra regularizer is adopted to learn the modules with logical functions.

### 3.5 Prediction layer

A considerable amount of collaborative information can be endowed to the embedding of users and items with the graph layer. The interaction layer can learn the interaction information effectively with the collaborative embedding. Since the goal is to calculate the occurrence probability of the behavior. Thus, a constant vector  $\mathbf{T} \in \mathbb{R}^d$  is defined to represent “True behavior”, which indicates the occurrence of the behavior. The calculation of similarity between  $\mathbf{O}$  and  $\mathbf{T}$  can directly imply the prediction of probability. The cosine similarity is chosen in this paper to measure the proximity between decision vector and True vector. Additionally, since contextual information assists models with stronger expressive [36], we incorporate the embedding of context features  $Cf$  and combine it with the cosine similarity for prediction via MLP. Cosine similarity between the two vectors and the prediction of CTR can be calculated as:

$$\begin{aligned} sim(\mathbf{O}, \mathbf{T}) &= \frac{\mathbf{O}\mathbf{T}}{\|\mathbf{O}\|\|\mathbf{T}\|} \\ p &= MLP(concat(sim(\mathbf{O}, \mathbf{T}), \mathbf{O}, \mathbf{T}, Cf)) \end{aligned} \tag{10}$$

Notably,  $p$  is the predicted probability of user clicks target item and  $MLP$  represents a two-layer fully connected network with ReLU activation.

### 3.6 Learning algorithm

The loss function has two components. The first one is the log-likelihood function which is widely applied in CTR models [30, 32]. This function considers both the label of a sample and the corresponding prediction score. Specifically, an ideal model should assign positive and negative samples with higher and lower scores respectively. The loss function is presented as:

$$L_{log} = -\frac{1}{N} \sum_{(x,y) \in \mathbb{D}} (y \log p + (1 - y) \log(1 - p)). \tag{11}$$

where  $\mathbb{D}$  is the training dataset and  $x, y$  and  $p$  are the inputs, labels and prediction probabilities respectively.

Second, logical equation is introduced on the basis of the idea w.r.t logical regularizers [35] to guarantee that NOT and OR modules can be learned with the property of propositional logic. The motivation of logical regularizer is that the modules, which have the capability of logical operation, should satisfy the basic logical operation laws. Thus, the interactions are transformed into logical representation and added to the loss function. The logical laws and the corresponding logical representation are listed in Table 1.

The loss function is organized by combining the log likelihood function and the logical regularizers together which can be represented as:

$$L = L_{log} + \frac{\lambda_1}{\|E^*\|} \sum_{i=1}^6 l_i + \lambda_2 \|\Theta\|_2^2 \tag{12}$$

**Table 1** Correspondence between logical laws and expressions applied in loss function

name	Law	Expression
$l_1$	$\neg T = F$	$\sum(1 + sim(NOT(E^*), E^*))$
$l_2$	$\neg(\neg x) = x$	$\sum(1 - sim(NOT(NOT(E^*)), E^*))$
$l_3$	$x \vee F = x$	$\sum(1 - sim(E^*, OR(E^*, NOT(T))))$
$l_4$	$x \vee T = T$	$\sum(1 - sim(T, OR(E^*, T)))$
$l_5$	$x \vee x = x$	$\sum(1 - sim(E^*, OR(E^*, E^*)))$
$l_6$	$x \vee \neg x = T$	$\sum(1 - sim(T, OR(E^*, NOT(E^*))))$

$E^*$  represents the representation of interaction information in the interaction matrix

where  $\|E^*\|$  represents the total number of interactions.  $\lambda_1, \lambda_2$  are the penalty coefficients of logical regularizers and Frobenius norm respectively.

### 4 Experiments

Several popular real-world datasets are used in the experiments to demonstrate the effectiveness of the proposed method, and several state-of-the-art and baseline methods are compared.

#### 4.1 Datasets

Five large-scale datasets are adopted, and these datasets are collected by Amazon from their websites [37], including *Video Games, Digital Music, Cell Phones and Accessories, Toys and Games and CDs and Vinyl*. These datasets are all from Amazon 5-core.<sup>1</sup>

The rating scores in these datasets are from a candidate {1,2, 3, 4, 5}. The data are transformed into binary classification data to apply data in CTR prediction task by labeling samples with ratings of 4 and 5 to be positive and the rest to be negative. CTR prediction is a binary problem, which indicates whether the user has clicked on the item; thus the highest score is set as 1. Meanwhile, each dataset is divided into training data and testing data based on the time stamp. For each user, The latest behavior of each user in the sequential behaviors according to his/her historical behaviors is used for testing. The latest behavior in the training sequential behaviors is used during the training procedure as the target item for training. For example, if the behaviors of a user are represented as  $B = [b_1, b_2, \dots, b_n]$ , then the latest behavior  $b_n$  is used for testing while the behavior  $b_{n-1}$  is used to train the model for prediction. A total of 10% of users in the training set with their historical behaviors are randomly selected as validation set.

#### 4.2 Experimental settings

The proposed GACR is compared with several popular CTR models, including graph-based CF models and logical reasoning models.

<sup>1</sup> <http://jmcauley.ucsd.edu/data/amazon/>

- **Wide&Deep** [2]: Wide & Deep Learning framework is widely used in modern industrial applications. This framework combines two technological parts: the wide part used the linear model with the capability of memorization, and the deep part can extract non-linear correlations among features via deep neural networks.
- **DeepFM** [3]: DeepFM is derived from Wide & Deep and utilizes an FM to replace the linear model for learning two-order feature interaction.
- **xDeepFM** [25]: A combination of the deep neural network and a novel compress information module to extract additional implicit information behind features for CTR prediction.
- **NGCF** [17]: A graph-based collaborative filtering model that exploits the graph structure by propagating embedding and generates expressive representations with high-order connectivity.
- **NLR** [35]: A cognition recommendation focused model, which integrates embedding learning technology and logical reasoning.
- **DIFM** [38]: A FM based model that combine the self-attention and deep neural network to learn both bit-wise and vector-wise level feature interaction simultaneously.

The learning rate is set as 0.001 and the optimizer with Adam, which is widely used in deep learning models. Meanwhile, the batch size for training with 1024 is chosen. For the proposed method, the weight of logical regularizer  $\lambda_1$  is set to 0.4 (except 0.1 in **Video Games**), the penalty coefficient of Frobenius norm regularizer  $\lambda_2$  is set to  $1 \times 10^{-5}$  and the embedding size is set to 16 (except 8 in **Digital Music** and 32 in **Toys and games**). Moreover, the number of iterations in graph layer is set to 4.

These models are assessed with two popular evaluations, namely, Area Under ROC Curve (AUC) to show the superiority of the proposed method. AUC, which reflects the ranking capability is suitable for imbalanced data [39]. A high AUC indicates ranking of additional positive samples before negative samples. AUC and logloss are defined as follows:

$$AUC = \frac{\sum I(f(x^+), f(x^-))}{M \times N} \quad (13)$$

where  $M, N$  indicates the total number of positive samples  $x^+$  and negative samples  $x^-$  respectively, and  $f(x)$  is the prediction results of  $x$ .  $I(x, y)$  are equal to 0, 0.5, 1 for  $x < y$ ,  $x = y$  and  $x > y$  respectively.

In addition, RelImpr metric [40] is introduced for measuring relative improvement of the proposed model. RelImpr is defined as follows:

$$RelImpr = \left( \frac{AUC_t - 0.5}{AUC_c - 0.5} - 1 \right) \times 100\% \quad (14)$$

where  $AUC_t, AUC_c$  represent the AUC achieved by the proposed model and the best result in the compared methods respectively.

### 4.3 Results

The results of the methods in the experiments are reported in Table 3. All the results are obtained with the average results in five runs. In the compared methods, WDL, DeepFM and xDeepFM are three methods that attempt to extract high-order feature

**Table 2** Statistics of five Amazon datasets

dataset	# users	# items	# interaction	# density
Video Games	55,222	71,976	231,780	0.005%
Digital Music	16,565	77,625	169,781	0.013%
Cell Phones	27,878	284,012	194,437	0.002%
Toys and Games	208,180	78,772	1,828,971	0.011%
CDs and Vinyl	112,394	418,435	1,097,592	0.004%

**Table 3** The AUC results of the comparison methods on five Amazon datasets

	Video Games	Digital Music	Cell Phones	Toys and Games	CDs and Vinyl
WDL	0.6721	<u>0.7378</u>	<u>0.6187</u>	0.6685	<u>0.6419</u>
DeepFM	<u>0.6882</u>	0.7370	0.6176	<u>0.6696</u>	0.6404
xDeepFM	0.6814	0.7307	0.6179	0.6677	0.6402
NGCF	0.6326	0.6997	0.5609	0.6125	0.5763
NLR	0.6706	0.7357	0.6057	0.6533	0.6328
DIFM	0.6835	0.7303	0.5981	0.6123	0.6329
GACR	<b>0.6945*</b>	<b>0.7641*</b>	<b>0.6237*</b>	<b>0.6769*</b>	<b>0.6507*</b>
Improv	0.91%	3.56%	0.80%	1.09%	1.37%
RelaImpr	3.34%	11.05%	4.21%	4.30%	6.20%

The best performance on each dataset is highlighted with bold font while the second best performance is marked with underline. Results with “\*” indicate the statistically improvements between the proposed method and the best result in the compared methods based on paired t-test with  $p$ -value at 0.05

interaction. These methods achieve the second-best performance of AUC on most datasets. Therefore, learning interaction information from embedding is crucial.

NGCF and NLR are two state-of-the-art methods for graph learning and logical reasoning respectively in recommendation. Notably, the performance of the two methods is worse than that of other feature based models on AUC. This finding may be due to the excessive number of similar items, which makes it difficult for NGCF and NLR to capture the specific item. DIFM is the state-of-the-art method designed based on FM architecture. As shown in Table 3, DIFM achieves worse performance at most cases. The explanation for this phenomenon may be that though DIFM utilizes complex network to learn the weight for feature interaction, it needs more data of feature interaction for training.

Meanwhile, the proposed method achieves the best performance on all datasets on AUC value. In particular, the proposed model is superior to the second-best results and RelaImpr reaches 11.05% and 6.2% on **Digital Music** and **CDs and Vinyl** respectively. This finding demonstrates that the proposed method can effectively capture the preference and the requirement of the users, and recommend additional specific items for the users. Hence, the combination of collaborative information and the logical behaviors is crucial to designing ideal advertising systems, and the proposed method can effectively achieve this goal.

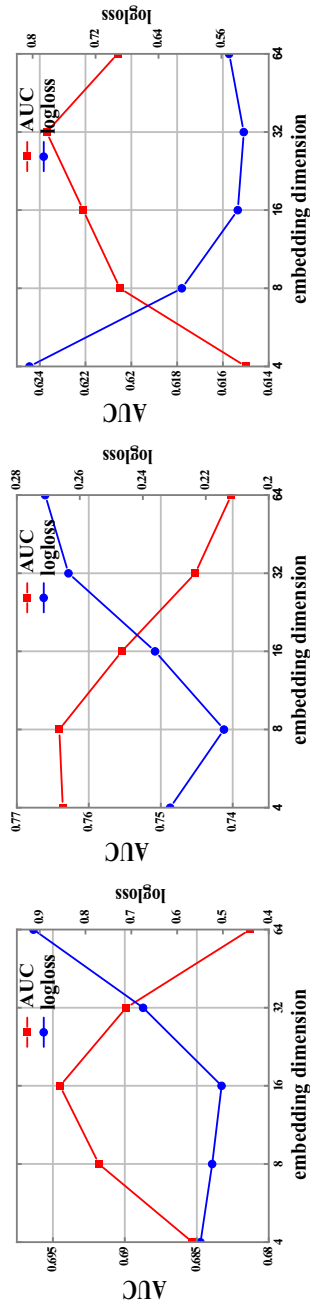


Figure 4 Effect of embedding dimension on (left) Video Games; (middle) Digital Music; (right) Cell Phones

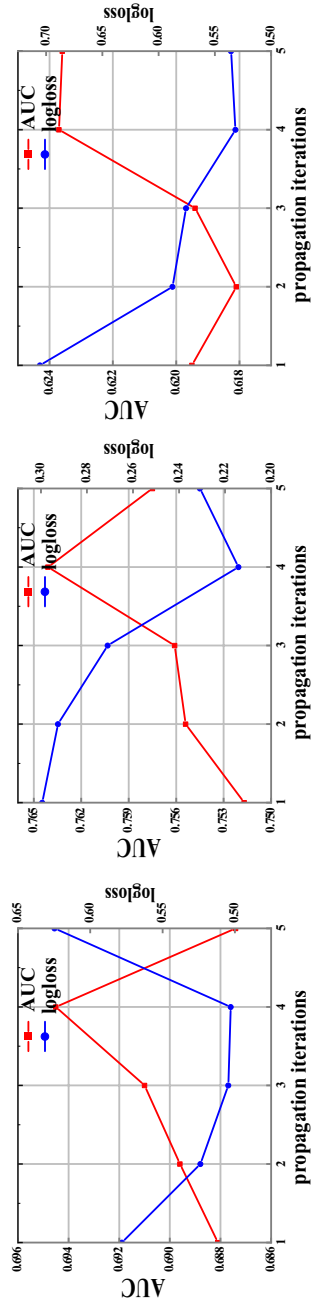


Figure 5 Effect of the iterations in graph layer on (left) Video Games; (middle) Digital Music; (right) Cell Phones

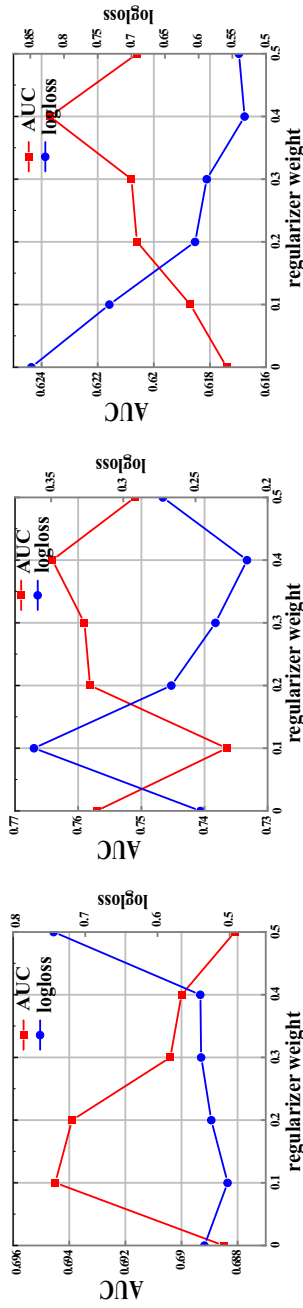
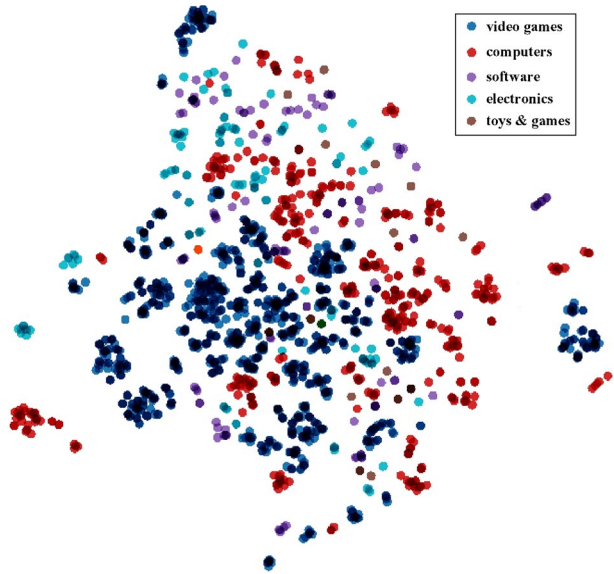


Figure 6 Effect of the weight  $\lambda_1$  about logical regularizer on (left) Video Games; (middle) Digital Music; (right) Cell Phones



**Figure 7** Visualization of the item embeddings in Video Games dataset via t-SNE. Color of points corresponds to the different categories



#### 4.4 Parameters analysis

The following effects of parameters in GACR are studied in this section: (1) the dimension of embedding; (2) the number of propagation iterations in the graph layer; (3) the weight of logical loss  $\lambda_1$  in (13).

##### 4.4.1 Effect of embedding dimension

The embedding dimension from the candidate set (4,8,16,32,64) to study the influence of embedding dimension, and results on **video game, digital music, and Cell Phones** are reported in Figure 4. In Figure 4, the red lines represent the AUC values while the blue lines represent the results of logloss. The results indicate that the proposed method is sensitive to the embedding dimension. The performance decreases when the embedding dimension is larger than 16 on video game, 8 on digital music and 32 on Cell Phones dataset. At Video Games and Cell Phones dataset, 8 or 16 is a small dimension for the optimal settings compared to former work. It is due to the message propagation mechanism in Graph layer. It could learn the collaborative information on different level in forms of embedding vectors and concatenate them together according to (4). Thus a small embedding dimension can lead to expressive embedding vectors. With the increase of embedding dimension, the overfitting problem leads to worse performance. Notably, at Cell Phones dataset with the most sparse interactions according to Table 2, the collaborative information is less and thus the optimal dimension increases to 32.

#### 4.4.2 Effect of propagation iterations $l$ in the graph layer

The collaborative information in the graph layer is learned by message propagation on graph. The message propagates recursively to learn the propagation layer effectively. Hence, the propagation iteration  $l$  is an important parameter in the proposed method. The propagation results on **video game**, **digital music**, and **Cell Phones** are presented by changing  $l$  from 1 to 5 in Figure 5. This figure shows that the results are sensitive to the  $l$ . AUC and logloss can obtain improved results when  $l$  is increased. However, the results decrease quickly on all datasets when  $l$  is larger than 4. This phenomenon may result in the overfitting problem due to the presence of many iterations. However, the best performance is achieved on all datasets when  $l$  is equal to 4. Hence,  $l$  can be set as 4 in real applications.

#### 4.4.3 Effect of $\lambda_1$

The performance with different weights on three datasets, including **video game**, **digital music**, and **Cell Phones**, is presented to evaluate the importance of logical reasoning and the results are reported in Figure 6. This Figure reveals that the results are sensitive to weight. The best performance is achieved on **Video Games** when  $\lambda_1$  is equal to 0.1 while the two other datasets achieve the best performance when  $\lambda_1$  is equal to 0.4. Compared without logical reasoning, the improvements are significant. Hence, the logical reasoning is essential for CTR prediction. Meanwhile, although the results begin to descend when  $\lambda_1$  is larger than 0.1 on **Video Games**, the descending is almost can be ignored. Hence,  $\lambda_1$  can be set as 0.4 for practical application.

### 4.5 Visualization of embedding

Five main categories of items in **Video Games** dataset are selected to show the effective learning of embedding, and several items from these categories are randomly sampled. Figure 7 shows the visualization of embedding vectors of items with t-SNE [41]. This figure indicates that although all the items that belong to the same category are not closed, several items in the same categories are always closed with each other because several subcategories may exist in one category. Hence, the consistency and the difference among the items can be learned effectively. Moreover, considering the embeddings cross categories, the items in different categories also have significant differences. These results confirm that the proposed method can learn cognitive embedding for the items, which is favorable to logical reasoning.

## 5 Conclusion

A graph-aware collaborative reasoning method for CTR prediction is proposed in this paper. The proposed method uses the graph to propagate messages and endow deep collaborative information for the embedding of users and items. The preference of the user can be effectively learned. A logical reasoning network is then adopted to learn

the logical behaviors in the sequential behaviors from the interaction information. The intention of the user can be inferred from the logical behaviors. The proposed method can simultaneously learn user preference and user requirements by the proposed architecture. Extensive experiments on five large-scale datasets demonstrate that the proposed method outperforms the state-of-the-art methods for CTR prediction.

**Author contributions** Three authors contributed equally to the data analysis, model construction, experiments and manuscript in this work.

**Funding** This work was supported in part by the National Natural Science Foundation of China under Grants 62006176, 62141112, 41871243, the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170, and the Natural Science Foundation of Hubei Province under Grants 2020CFB241. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

**Data availability** The datasets analyses during the current study are available in the “<http://jmcauley.ucsd.edu/data/amazon/>”.

**Code availability** Not applicable.

## Declarations

**Ethics approval** The work was established, according to the ethical guidelines.

**Consent to participate** Not applicable.

**Consent for publication** Not applicable.

**Conflict of interests** We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of the manuscript.

## References

1. Rendle, S.: Factorization machines. In: 2010 IEEE International Conference on Data Mining, pp 995–1000. IEEE (2010)
2. Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., 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. 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)
4. Shang, S., Ding, R., Yuan, B., Xie, K., Zheng, K., Kalnis, P.: User oriented trajectory search for trip recommendation. In: Proceedings of the 15th International Conference on Extending Database Technology, pp 156–167 (2012)
5. Chen, L., Shang, S., Jensen, C.S., Yao, B., Zhang, Z., Shao, L.: Effective and efficient reuse of past travel behavior for route recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 488–498 (2019)
6. Chen, L., Shang, S.: Region-based message exploration over spatio-temporal data streams. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp 873–880 (2019)
7. Menon, A.K., Chitrapura, K.-P., Garg, S., Agarwal, D., Kota, N.: Response prediction using collaborative filtering with hierarchies and side-information. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 141–149 (2011)

8. Lyu, Z., Dong, Y., Huo, C., Ren, W.: Deep match to rank model for personalized click-through rate prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp 156–163 (2020)
9. Yang, C., Chen, L., Wang, H., Shang, S.: Towards efficient selection of activity trajectories based on diversity and coverage. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp 689–696 (2021)
10. Gori, M., Pucci, A.: Itemrank: a random-walk based scoring algorithm for recommender engines. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp 2766–2771 (2007)
11. Yang, J.-H., Chen, C.-M., Wang, C.-J., Tsai, M.-F.: Hop-rec: high-order proximity for implicit recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp 140–144 (2018)
12. Nikolakopoulos, A.N., Karypis, G.: Recwalk: Nearly uncoupled random walks for top-n recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp 150–158 (2019)
13. Han, P., Wang, J., Yao, D., Shang, S., Zhang, X.: A graph-based approach for trajectory similarity computation in spatial networks. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp 556–564 (2021)
14. Sun, Z., Yang, J., Zhang, J., Bozzon, A., Huang, L.-K., Xu, C.: Recurrent knowledge graph embedding for effective recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp 297–305 (2018)
15. Hu, B., Shi, C., Zhao, W.X., Yu, P.S.: Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1531–1540 (2018)
16. Jin, J., Qin, J., Fang, Y., Du, K., Zhang, W., Yu, Y., Zhang, Z., Smola, A.J.: An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 75–84 (2020)
17. Wang, X., He, X., Wang, M., Feng, F., Chua, T.-S.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 165–174 (2019)
18. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.-S.: Kgat: Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 950–958 (2019)
19. He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., et al: Practical lessons from predicting clicks on ads at facebook. In: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, pp 1–9 (2014)
20. Xiao, J., Ye, H., He, X., Zhang, H., Wu, F., Chua, T.: Attentional factorization machines: learning the weight of feature interactions via attention networks (2017)
21. Juan, Y., Zhuang, Y., Chin, W.-S., Lin, C.-J.: Field-aware factorization machines for ctr prediction. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp 43–50 (2016)
22. Zhang, W., Du, T., Wang, J.: Deep learning over multi-field categorical data. In: European Conference on Information Retrieval, pp 45–57. Springer (2016)
23. Qu, Y., Cai, H., Ren, K., Zhang, W., Yu, Y., Wen, Y., Wang, J.: Product-based neural networks for user response prediction. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp 1149–1154. IEEE (2016)
24. He, X., Chua, T.-S.: Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 355–364 (2017)
25. Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., Sun, G.: xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1754–1763 (2018)
26. Shan, Y., Hoens, T.R., Jiao, J., Wang, H., Yu, D., Mao, J.: Deep crossing: Web-scale modeling without manually crafted combinatorial features. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 255–262 (2016)
27. Wang, R., Fu, B., Fu, G., Wang, M.: Deep & cross network for ad click predictions. In: Proceedings of the ADKDD'17, pp 1–7 (2017)
28. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015 (2015)

29. Zhai, S., Chang, K.-H., Zhang, R., Zhang, Z.M.: Deepintent: Learning attentions for online advertising with recurrent neural networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1295–1304 (2016)
30. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., Gai, K.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1059–1068 (2018)
31. Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., Gai, K.: Deep interest evolution network for click-through rate prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp 5941–5948 (2019)
32. Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., Yang, K.: Deep session interest network for click-through rate prediction. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp 2301–2307. AAAI Press (2019)
33. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: ICML Workshop on Deep Learning for Audio, Speech and Language Processing. Citeseer (2013)
34. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2, pp 3111–3119 (2013)
35. Shi, S., Chen, H., Ma, W., Mao, J., Zhang, M., Zhang, Y.: Neural logic reasoning. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp 1365–1374 (2020)
36. Han, P., Li, Z., Liu, Y., Zhao, P., Li, J., Wang, H., Shang, S.: Contextualized point-of-interest recommendation. In: International Joint Conferences on Artificial Intelligence Organization (2020)
37. McAuley, J., Targett, C., Shi, Q., Van Den Hengel, A.: Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 43–52 (2015)
38. Lu, W., Yu, Y., Chang, Y., Wang, Z., Li, C., Yuan, B.: A dual input-aware factorization machine for ctr prediction. In: Proceedings of the 31st International Joint Conference on Artificial, pp 3139–3145 (2020)
39. Han, P., Shang, S., Sun, A., Zhao, P., Zheng, K., Kalnis, P.: Auc-mf: point of interest recommendation with auc maximization. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp 1558–1561. IEEE (2019)
40. Yan, L., Li, W.-J., Xue, G.-R., Han, D.: Coupled group lasso for web-scale ctr prediction in display advertising. In: International Conference on Machine Learning, pp 802–810 (2014)
41. Maaten, L.V.D., Hinton, G.: Visualizing data using t-sne. *J. Mach. Learn. Res.* **9**(Nov), 2579–2605 (2008)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.