Knowledge and Information Systems (2022) 64:2543–2564
https://doi.org/10.1007/s10115-022-01714-4

REGULAR PAPER

Graph relation embedding network for click-through rate prediction

Yixuan Wu[1] · Youpeng Hu[2] · Xin Xiong[3] · Xunkai Li[2] · Ronghui Guo[2] · Shuiguang Deng[1]

Received: 24 September 2021 / Revised: 19 June 2022 / Accepted: 26 June 2022 /
Published online: 4 August 2022

Abstract
Most deep click-through rate (CTR) prediction models utilize a mainstream framework, which consists of the embedding layer and the feature interaction layer. Embeddings rich in semantic information directly benefit the downstream frameworks to mine potential information and achieve better performance. However, the embedding layer is rarely optimized in the CTR field. Although mapped into a low-dimensional embedding space, discrete features are still sparse. To solve this problem, we build graph structures to mine the similar interest of users and the co-occurrence relationship of items from click behavior sequences, and regard them as prior information for embedding optimization. For interpretable graph structures, we further propose graph relation embedding networks (GREENs), which utilize adapted order-wise graph convolution to alleviate the problems of data sparsity and over-smoothing. Moreover, we also propose a graph contrastive regularization module, which further normalizes graph embedding by maintaining certain graph structure information. Extensive experiments have proved that by introducing our embedding optimization methods, significant performance improvement is achieved.

Keywords Click-through rate · Graph embedding · Recommender system · Graph neural network

1 Introduction

Whether in online advertising, search engines, or recommender systems [37], human–computer interaction [21], movies [17], robot service [24], and intelligent control [23], click-through rate (CTR) prediction tasks are of great research and commercial value, whose result can rank the items returned to a user to maximize the number of clicks.

✉ Shuiguang Deng
dengsg@zju.edu.cn

1 Zhejiang University, Hangzhou, China

2 Shandong University, Weihai, China

3 Nanjing University, Nanjing, China

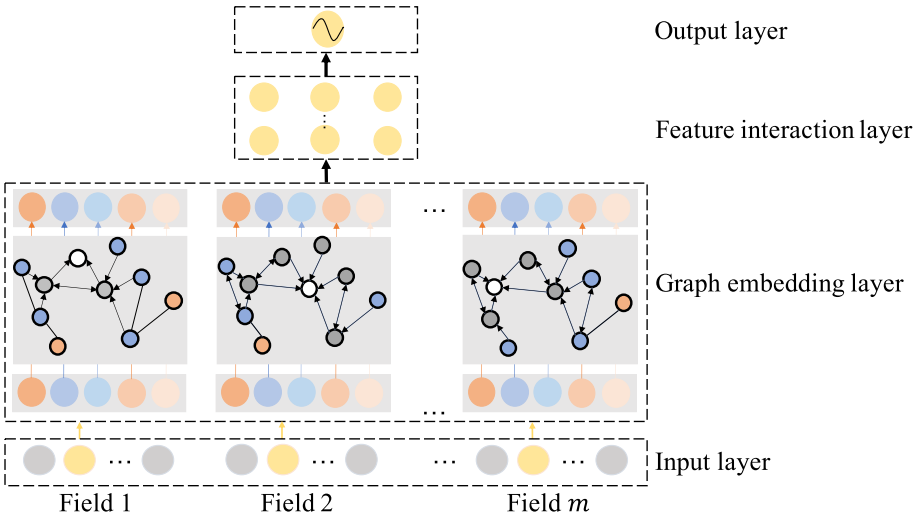


Fig. 1 The deep CTR prediction architecture with the graph embedding layers

Deep learning methods have stronger expressive ability and more flexible structures, which can better handle classification tasks. Instead of traditional methods, a series of representative deep CTR models have been developed by introducing neural networks, such as Wide&Deep [3] and PNN [27], etc. In recent years, researchers have made various meaningful attempts. DeepFM [7] is an end-to-end model with stronger generalization ability and memory ability, which extracts both low- and high-order feature interactions by introducing factorization machine (FM). DIN [44] and DIEN [43], which achieve considerable performance improvements, utilize users' historical click behaviors to mine the distribution and transfer of users' interest, respectively, as the prior information to provide an estimate of current interest. It can be seen that the inherent prior information of features can effectively improve prediction accuracy.

The above deep CTR models are mainly composed of an input layer, an embedding layer, a feature interaction layer, and an output layer, which can be regarded as a joint optimization of representation learning and task-oriented learning. The embedding layer is responsible for densifying the discrete features as their corresponding representations and then passing them to the downstream modules for feature interaction. In practice, densification requires a large number of data to support, while a user's click behaviors or a item's clicked behaviors are too sparse compared to numerous items and users, which brings challenges to learn representations with rich semantic information.

Graph embedding has achieved an excellent effect in the field of embedding representation [4, 16, 20, 41]. By constructing interpretable graph structures, graph learning methods can be applied to the CTR prediction for a more reasonable feature space, as shown in Fig. 1. Based on click behavior sequences, graph intention network (GIN) [18] constructs the co-occurrence graph of items and aggregates neighbor nodes by attention mechanism to solve the problems of over-sparse and weak generalization. However, GIN leaves a lot to be desired, such as underutilized relationship information and slow convergence speed.

In addition, the CTR model is prone to overfitting due to a huge amount of parameters for feature extraction and interaction, resulting in its poor generalization ability. To solve it, there is a mainstream solution to introduce regularization terms to enhance generalization

ability. Inspired by graph contrastive learning methods such as deep graph infomax (DGI) [32], we can maintain certain graph structure information in a self-supervised way to prevent the overfitting phenomenon. With fewer parameters and no additional labels required, it is considered as an efficient regularization strategy of the graph embedding layer.

Based on previous works and the pain points of CTR prediction, we propose our novel embedding optimization method, namely graph relation embedding network (GREEN), on the item co-occurrence graph and the user co-interest graph constructed through click behavior sequences, as shown in Fig. 3. An adapted order-wise graph convolution is designed in GREEN to aggregate information and provide rich prior information for a more reasonable feature space. Moreover, we propose a graph contrastive regularization (GCR) method to suppress the overfitting phenomenon. It is emphasized that our method is applicable to any deep CTR models for embedding optimization.

In summary, the main contributions of this paper are as follows:

- We mine potential relations based on click behaviors and propose a reasonable and interpretable construction strategy of the item co-occurrence graph and the user co-interest graph.
- We propose the graph relation embedding network. Through the relations among multi-hop neighbors, it can effectively alleviate the data sparsity and learn better representation.
- As a novel regularization method, graph contrastive regularization is proposed to relieve the problem of overfitting.
- Our method is implemented on the public datasets and achieved a considerable performance improvement compared to the baseline.

2 Preliminaries

The datasets for CTR prediction consist of n samples, each of which is represented as $(\mathbf{x}, y)$, where $y \in \{0, 1\}$ represents whether the user clicks the target item in the specific context, $\mathbf{x} = [\mathbf{x}_{field_1}, \mathbf{x}_{field_2}, ..., \mathbf{x}_{field_m}]$, and $\mathbf{x}_{field_i}$ is to describe the feature of user, item, context, or others.

Since discrete features are often encoded to sparse one-hot vectors, they are densified through an embedding layer in deep CTR models:

$$(\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m) = f(\mathbf{x}_{field_1}, \mathbf{x}_{field_2}, ..., \mathbf{x}_{field_m}), \tag{1}$$

where $f(\cdot)$ represents the embedding function which follows the table lookup mechanism, and $\mathbf{e}$ represents the corresponding embeddings. Further, the CTR prediction result is obtained through a feature interaction layer and an output layer:

$$\hat{y} = \sigma(g(\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m)), \tag{2}$$

where $g(\cdot)$ represents the feature interaction function, such as multilayer perceptrons, $\hat{y}$ is the prediction result of the current data $\mathbf{x}$, and $\sigma(t) = \frac{1}{1+e^{-t}}$. In the training process, the binary cross entropy function is utilized to calculate prediction loss:

$$\mathcal{L}_{BCE} = -\frac{1}{n}\sum_{j}^{n}\left[y_j \ \log \ \hat{y}_j + (1 - y_j) \ \log \ (1 - \hat{y}_j)\right]. \tag{3}$$

Finally, the end-to-end joint optimization process is carried out by the back propagation.

3 Proposed approach

3.1 Graph construction

Compared with the specific framework of Graph Neural Networks (GNNs), a reasonable and interpretable graph structure determines the upper limit of model accuracy to a greater degree. We construct graph structures for users and items, respectively, to enrich their embeddings for both attribute information and the corresponding topology. Embeddings richer in semantic information directly benefit the downstream frameworks for feature interaction to achieve better performance. Graph construction is based on the following accepted assumptions:

- Users who click on the same item have a degree of interest similarity. The degree of interest similarity of users is related to the number of the same items they clicked in general.
- Items which are continuously clicked by the same user have a certain co-occurrence relation. The degree of co-occurrence relation of items depends on the times of being clicked by the same users continuously.

Given the item set $\mathbf{I} = \{i_1, i_2, ..., i_N\}$, the user set $\mathbf{U} = \{u_1, u_2, ..., u_M\}$, and the click behaviors $\mathbf{b}_j = [i_{j_1}, i_{j_2}, ..., i_{j_k}]$ of each user u_j, where N and M represent the total number of items and users, respectively, and the length of the click behaviors k is different for different users, we construct a user co-interest graph $\mathbf{G}_u = (\mathbf{U}, \mathcal{E}_u)$ and an item co-occurrence graph $\mathbf{G}_i = (\mathbf{I}, \mathcal{E}_i)$, where $\mathcal{E}_u$ and $\mathcal{E}_i$ are the weighted edge sets of the user co-interest graph and the item co-occurrence graph, respectively.

The construction of the item co-occurrence graph is shown in Fig. 2a. Referring to Li et al. [18], we iterate through each user's click behavior sequences in chronological order to connect items that have been continuously clicked. If the two items are connected for the first time, their weight is set to one, otherwise, their weight is increased by one. A bigger weight between any two items illustrates that it is more possible for them to be continuously clicked again.

On the other side, the user co-interest graph is shown in Fig. 2b, where s_j represents the user set clicked the same item i_j. We connect all users who have clicked the same non-popular item, whose clicked number is smaller than maximum length L_u, for popular items lead to considerable meaningless relationships of users constructed and increase the complexity of the graph. If they have been connected, the weight is increased by one. In this way, a bigger weight between users means more similar click interests.

Based on this, we obtain the weighted adjacency matrices $\mathbf{A}_i \in \mathbb{R}^{N \times N}$ and $\mathbf{A}_u \in \mathbb{R}^{M \times M}$ to describe the connection relationship and strength among items or users. It is emphasized that not only can the user embedding and the item embedding be optimized through our model, but also others with latent graph structures are well applied.

3.2 Graph relation embedding network

In the item co-occurrence graph and the user co-interest graph, connected nodes have similar clicked intentions or click interests. For example, when the user u_j clicks on the item i_m, it is extremely possible that u_j will click on another item that has a co-occurrence relationship with i_m, and i_m will be clicked by another user that has a co-interest relationship with u_j. It will be reflected in the relative position in the feature space, where the embedding representations of co-occurrence items or co-interest users are more closer through the neighbor aggregation of nodes on the graph.

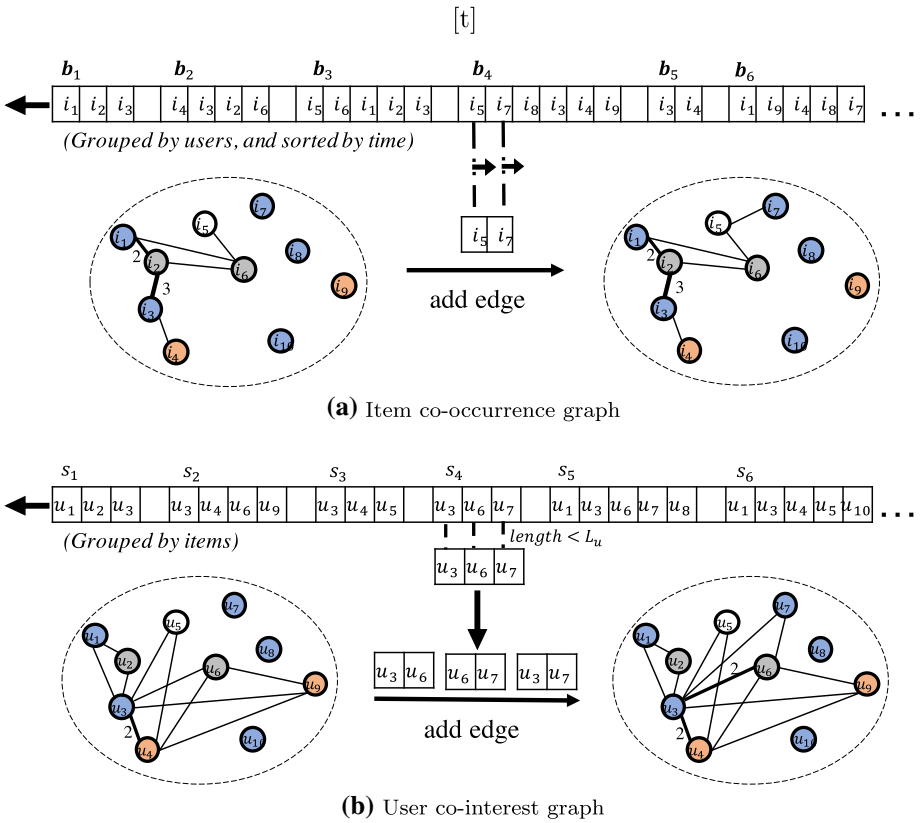


Fig. 2 An illustration of graph construction, where b_i represents the sorted click behavior sequences of user i, s_j represents the user set clicked item j, and j will be regarded as a popular item if the length of s_j is longer than L_u

Graph relation embedding network architecture is shown in Fig. 3. Taking item feature as an example, we define the initial embedding matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, the weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ of the graph structure, and the edges' degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$, where d is the dimension of embeddings and $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{i,j}$. The graph convolutional network (GCN) was proposed in Kipf and Welling [15]:

$$\mathbf{H} = \mathbf{GX2} = \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}\mathbf{X2}, \tag{4}$$

where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$, $\widetilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}_N$, $\mathbf{I}_N$ is an identity matrix whose dimension is the number of nodes N, and $\mathbf{2}$ is a learnable right multiplication matrix, which is used to map the feature space of $\mathbf{X}$ to a new feature space. In Eq. (4), the graph convolution kernel $\mathbf{G} = \widetilde{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}$ is used for feature aggregation of adjacent nodes.

For the CTR task, the embedding representations of users and items have exclusive feature space. Therefore, it is unnecessary to perform redundant and repeated feature space mapping, or introduce considerable optional learnable parameters resulting in the inference slowdown, which is also verified in Wu et al. [34]. From the formula analysis, in our work k order graph convolution is defined as:

$$\mathbf{X}^{(k)} = \mathbf{GX}^{(k-1)}, \tag{5}$$

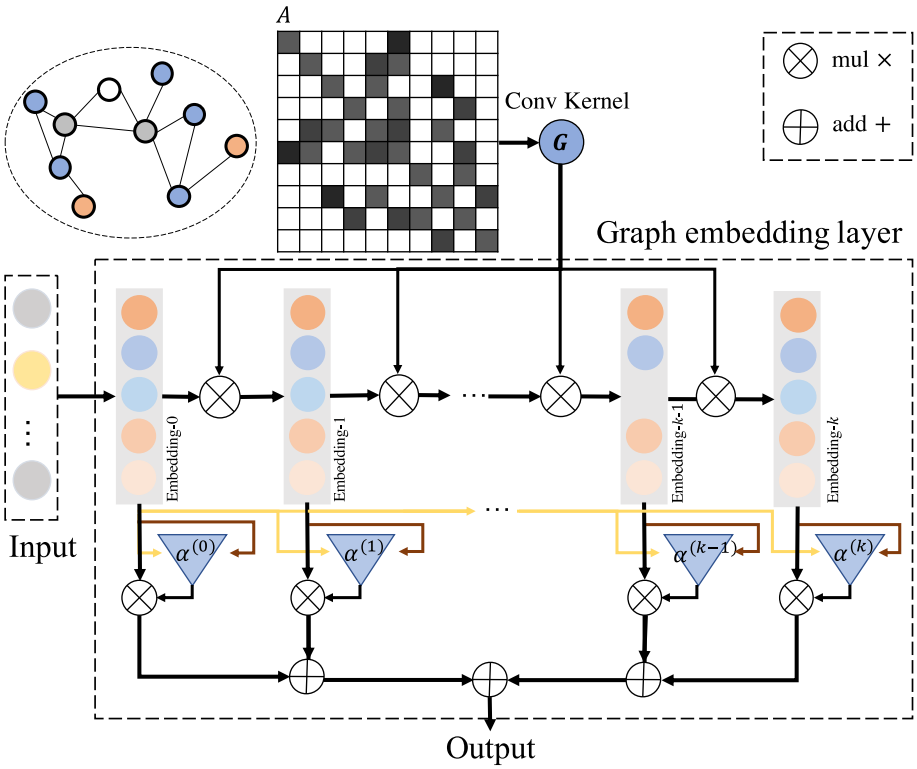


Fig. 3 Specific implementation of the graph relation embedding network

where $\mathbf{X}^{(0)} = \mathbf{X}$ which is the initial embedding matrix, and $\mathbf{X}^{(k)}$ is the feature aggregation of $\mathbf{X}$ through k times.

When we use multi-order graph convolution to perform the aggregation representation of embedding, we will inevitably encounter over-smoothing problems, *i.e.,* the phenomenon that all node embeddings tend to be consistent, making the CTR prediction inaccurate. In order to solve the problem, we introduce adapted order-wise weights. Inspired by the weight of attention defined in Li et al. [18], the weight calculation corresponding to the k order graph convolution is:

$$\alpha^{(k)} = \frac{1}{N}\sum_{j=1}^{N}\sigma\left[(\mathbf{X}^{(k)}\|\mathbf{X}\|(\mathbf{X}^{(k)} - \mathbf{X})\|(\mathbf{X}^{(k)} \odot \mathbf{X}))\mathbf{W}\right], \tag{6}$$

where $\cdot\|\cdot$ represents matrix concatenation, $\sigma(\cdot)$ is the sigmoid function, $\odot$ denotes the element-wise product, and $\mathbf{W} \in \mathbb{R}^{4d\times 1}$ is the trainable parameter. Then, the final output of the GREEN is:

$$\mathbf{X}_{out} = \sum_{j=0}^{k}\alpha^{(j)}\mathbf{G}\mathbf{X}^{(j)}. \tag{7}$$

Equation (7) brings additional time complexity $O(kdM)$ to base model by sparse computing, where M represents the number of edges, and $M >> k, d$. Reasonable trimming for edges can effectively accelerate the inference.

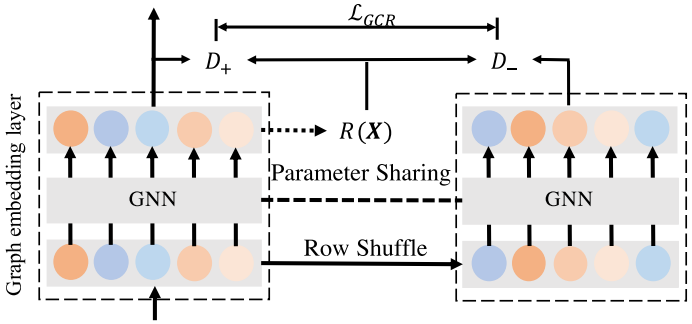


Fig. 4 An illustration of the graph contrastive regularization

Applying the architecture of GREEN, each embedding is learned by more sufficient data through the graph structure, which greatly alleviates the problem of data sparsity. Therefore, by truncating outdated historical behaviors, the graph is adapted to the real-time relationship changed rapidly, and the model can even obtain higher accuracy with less data.

3.3 Graph contrastive regularization

Inspired by the deep graph infomax (DGI) [32] model based on the contrastive paradigm, a regularization method based on graph contrastive learning is proposed to further suppress overfitting, namely graph contrastive regularization (GCR), as shown in Fig. 4.

The core idea of contrastive learning is to find three components from the original data: positive sample, negative sample, and anchor. We randomly shuffle the initial embedding matrix $\mathbf{X}$ of nodes to generate fake features $\widetilde{\mathbf{X}}$, and set $\mathbf{x}_i$ and $\widetilde{\mathbf{x}}_i$ as the real and fake feature of i-th node. We input $\widetilde{\mathbf{X}}$ into the same GREEN model to obtain the output representation of the embedding layer:

$$\widetilde{\mathbf{X}}_{out} = \sum_{j=0}^{k} \alpha^{(j)} \mathbf{G} \widetilde{\mathbf{X}}^{(j)}, \tag{8}$$

where $\widetilde{\mathbf{X}}_{out}$ represents the fake feature matrix. In addition, we use the mean function as the readout step to extract the graph embedding representation as the anchor:

$$R(\mathbf{X}) = \sigma\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\right), \tag{9}$$

where $\sigma(\cdot)$ is the sigmoid function. Furthermore, a bilinear scoring function is utilized as the discriminator:

$$D(\mathbf{x}_i, \mathbf{r}) = \sigma\left(\mathbf{x}_i{}^T \mathbf{W} \mathbf{r}\right), \tag{10}$$

where $\mathbf{r}$ represents the anchor introduced in Eq. (9), $\sigma(\cdot)$ is the sigmoid function, and $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a learnable scoring matrix to generate the sample's score for $(\mathbf{x}_i, \mathbf{r})$. Finally, we use noise contrastive estimation(NCE) loss [25]:

$$\mathcal{L}_{GCR} = -\frac{1}{2M}\left(\sum_{v_i \in S} \mathbb{E}_{(\mathbf{X},\mathbf{A})}\left[\log\ D(\mathbf{x}_{out\ i}, R(\mathbf{X}_{out}))\right] + \sum_{v_j \in S} \mathbb{E}_{(\widetilde{\mathbf{X}},\mathbf{A})}\left[1 - log\ D(\widetilde{\mathbf{x}}_{out\ j}, R(\mathbf{X}_{out}))\right]\right), \tag{11}$$

where $S \subset \mathcal{V}$ is a randomly selected subset of the node set $\mathcal{V}$. To balance the consumption of additional resources and the degree of regularization, the size $|S|$ is adjusted depending on datasets characteristics. By Eq. (11), we effectively maximizes the mutual information between $x_{out\ i}$ and $R(X_{out})$, *i.e.,* the JS divergence between the joint distribution and the product of marginal distribution.

$\mathcal{L}_{GCR}$ can be directly added to the base model loss Eq. (3) for integrated end-to-end learning:

$$\mathcal{L} = \mathcal{L}_{BCE} + w\mathcal{L}_{GCR}, \tag{12}$$

where w is the weight of the GCR module as regularization term, which balances the graph contrastive learning and CTR prediction. By sharing parameters of GREEN, the two tasks complement information and promote learning together, improving the generalization ability of the model. Specifically, in the training process, all trainable parameters in the model are optimized by minimizing $\mathcal{L}$. In the test process, the prediction result is obtained through the main task without GCR.

Overall, the presence of ancillary loss $\mathcal{L}_{GCR}$ has several advantages. First, the introduction of GCR will maintain certain graph structure information to a certain extent. Secondly, the multi-task learning paradigm can relieve the overfitting phenomenon because it can improve the model's robustness to unseen data [26]. Finally, as an ancillary task, it does not impose any computational burden on the model application.

4 Experiments

4.1 Experimental settings

4.1.1 Datasets

The statistical information of the datasets is shown in Table 1, and the description is as follows:

Amazon[1] [10]: is used as the benchmark dataset with pretty rich click behaviors for CTR prediction, which contains product reviews and metadata. We use two subsets: **Electronics** and **Movies and TV**, to verify the effect of our embedding optimization method. We group the samples by users, and each user's click behaviors can be described as $(b_1, b_2, ..., b_n)$. Our goal is to predict each user's n-th click behavior based on the past $n-1$ behaviors.

MovieLens[2] [9]: is a dataset used to describe users' ratings ranging from 0 to 5, which is treated as a binary classification problem here, where the click data with the rating no less than 4 are regarded as positive samples, and others are regarded as negative. We group the samples by each user to predict his n-th click behavior following the Amazon dataset.

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

[2] https://grouplens.org/datasets/movielens/20m/.

Table 1 Statistics of the datasets

Dataset	Users	Items	*Categories*	Samples
Amazon (Electro)	192,403	63,001	801	1,689,188
Amazon (Movies and TV)	123,960	50,052	29	1,697,533
MovieLens	138,493	27,278	21	20,000,263

In order to prevent the over-rich historical information, we take the latest 10 historical click behaviors as users' latent interests in our experiments to enhance the prediction difficulty.

4.1.2 Baselines

We introduce GREEN and GCR successively on five basic models to verify our methods.

Wide & Deep [3]: combined by a wide component and a deep component is proposed to capture both low-order and high-order feature interactions, which takes both memory ability and generalization ability of the model into account.

PNN [27]: introduces a product layer after the embedding layer to better extract high-order feature interactions.

Deep Crossing [30]: uses multiple residual units to mine the relationship between features, instead of explicitly interacting features.

DeepFM [7]: combines the advantages of factorization machines (FMs) and deep learning networks (DNNs) to shorten the convergence time while ensuring accuracy, where FM extracts low-order feature interactions through multiple inner product units and a linear unit, and DNN extracts high-order interactions among features through MLP layers.

DIN [18]: introduces the attention mechanism to extract the users' latent interests from their own historical behaviors. We combine the feature of historical behaviors with user profile feature, item feature, context feature, etc, and then input them to MLP for end-to-end learning.

4.1.3 Matrics

In the field of CTR prediction, AUC is used to evaluate the effectiveness of models [6]. Since the goal of CTR prediction is to sort the candidate items of each user, there are differences among different users, such as some users have a higher click rate or often give higher scores to items. Therefore, we use GAUC proposed by Zhu et al. [46] as our AUC matric:

$$AUC = \frac{\sum_{i=1}^{n} \#impression_i \times AUC_i}{\sum_{i=1}^{n} \#impression_i}. \tag{13}$$

Furthermore, RelaImpr [36] is used to measure relative improvement:

$$RelaImpr = \left(\frac{AUC(objective\ model) - 0.5}{AUC(base\ model) - 0.5} - 1 \right) \times 100\%. \tag{14}$$

4.1.4 Implementation

To verify the validity, the hyperparameters for all models are consistent, followed by Zhou et al. [44]. The models are learned by the Adam optimizer, the learning rate is set to 0.001, and

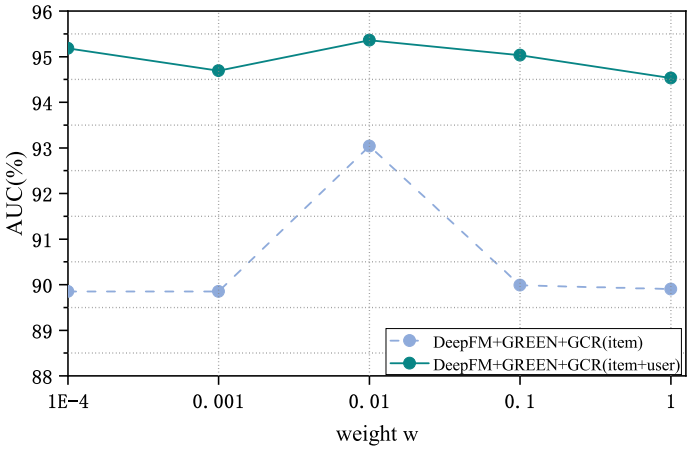


Fig. 5 The relation curve between the weight w of the GCR module mentioned in Eq. (12) and AUC on Amazon (Electro)

its decay rate is 0.9 per 336,000 samples. For all datasets, the training batch size is set to 32, the testing batch size is set to 512, the embedding dimension d is set to 128, the maximum length L_u is set to 40, the order k of GREEN is set to 4, and the sigmoid function is used as the activation function.

For the GCR module, the contrastive size $|S|$ is relevant to the scale of the dataset, and the weight w is relevant to the model. For Amazon (Electro), $|S|$ and w are set to 3000 and 0.01, respectively. For MovieLens, $|S|$ is set to 1000, and w is set to 2 for all except 0.001 for Deep Crossing. And for Amazon (Movies and TV), $|S|$ is set to 25,000, and w is set to 0.01, 0.001, 0.1, 0.001, 0.1 for Wide & Deep, PNN, Deep Crossing, DeepFM, and DIN, respectively. In order to study the performance impact of the two parameters in the GCR module, we conduct experiments by introducing GREEN and GCR in DeepFM on the Amazon (Electro). As can be seen from Fig. 5, by fixing $|S|$ to 3000, AUC varies with w, and the model performs best when w is set to 0.01. Similarly, by fixing w to 0.01, the model performs best when $|S|$ is set to 3000 as shown in Fig. 6.

4.2 Ablation study

Based on Wide & Deep, PNN, Deep Crossing, DeepFM, and DIN, we extract item features and user features of click samples in turn to verify the validity of the item co-occurrence graph and the user co-interest graph applied GREEN architecture and GCR successively, and the experimental results are shown in Tables 2, 3, and 4. On all base models and datasets, GREEN can provide consistent and significant performance improvement, even $relaImpr$ is up to 27.97%. Compared to MovieLens with richer samples, GREEN has more impressive improvements on Amazon with sparse samples, which proves that our model can relieve the problem of data sparsity. Moreover, the introduction of GCR can achieve further performance improvements.

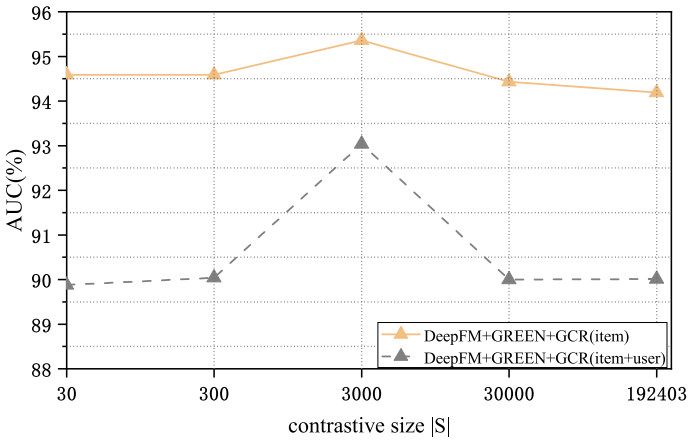


Fig. 6 The relation curve between the contrastive size $|S|$ of the GCR module mentioned in Eq. (11) and AUC on Amazon (Electro), where the total number of nodes is 192403

4.3 Comparative study

To further verify the effectiveness of our methods, we compare with the existing CTR models for feature optimization. DUSIN [14] and DDIL [40] are both sequential recommendation models for CTR prediction, which model behavior sequences to optimize the feature of user interest. DUSIN extracts and segments users' dynamic interests by considering the user's own historical sequence and potential interests of similar users. DDIL divides user interests into local sessions and global sessions, which are used to capture users' short-term dynamic interests and long-term interests, respectively. In addition, DDIL learns the heterogeneous behaviors within the sessions with consistency learning. The two models both concatenate user feature, item feature, and the optimized feature of user interest into a multiple layer perception (MLP).

The experimental results in Table 5 show that, in most cases, the model performs best by introducing GREEN rather than DUSIN or DDIL. Moreover, it can bring further performance improvement when utilizing our proposed embedding optimization method on the item feature and user feature of DUSIN and DDIL.

4.4 Historical behavior truncation study

We restrict the number of click behaviors of each user as l for sparse data on Amazon (Electro), *i.e.,* retain the latest l historical behaviors $(b_{n-l}, b_{n-l+1}, ..., b_{n-1})$ in the behavior sequences $(b_1, b_2, ..., b_{n-1})$. The truncation parameters and experimental results are shown in Table 6, where m represents the maximum number of behaviors, and $r\%$ represents the proportion of samples retained in the dataset. It is obvious that GREEN achieves a greater performance improvement for more sparse click behaviors. Through the experimental results, we observe that for lower m, the accuracy of the base model decreases due to data sparsity, but the GREEN-based model shows a remarkable upward trend. It shows that the GREEN framework is less sensitive to data sparsity, because relatively complete graph structures can be constructed with limited historical behaviors, and even the graphs constructed by more real-time data can achieve more excellent performance.

Table 2 Prediction results on the Amazon (Electro) dataset

	Item		Item&User	
	AUC	RelaImpr (%)	AUC	RelaImpr (%)
Wide & Deep	86.09	0.00	86.09	0.00
+GREEN	89.30	8.89	93.19	19.67
+GCR	89.30	8.89	93.24	19.81
PNN	86.54	0.00	85.83	0.00
+GREEN	90.18	9.96	95.85	27.97
+GCR	90.30	10.29	95.88	28.05
Deep Crossing	86.72	0.00	85.85	0.00
+GREEN	89.33	7.11	93.43	21.14
+GCR	89.43	7.38	93.47	21.26
DeepFM	86.93	0.00	86.38	0.00
+GREEN	89.87	7.96	94.94	23.52
+GCR	89.95	8.17	95.36	24.68
DIN	87.15	0.00	87.27	0.00
+GREEN	91.37	11.35	94.16	18.48
+GCR	91.45	11.57	94.34	18.96

Table 3 Prediction results on the MovieLens dataset

	Item		Item&User	
	AUC	RelaImpr (%)	AUC	RelaImpr (%)
Wide & Deep	71.73	0.00	71.00	0.00
+GREEN	72.83	5.06	71.59	2.81
+GCR	72.83	5.06	72.21	5.76
PNN	73.50	0.00	73.92	0.00
+GREEN	74.45	4.04	74.54	2.59
+GCR	74.55	4.47	75.40	6.19
Deep Crossing	72.41	0.00	72.00	0.00
+GREEN	73.59	5.27	72.77	3.50
+GCR	73.68	5.67	72.79	3.59
DeepFM	73.66	0.00	75.74	0.00
+GREEN	74.73	4.52	76.06	1.24
+GCR	75.22	6.59	76.07	1.28
DIN	74.53	0.00	73.97	0.00
+GREEN	75.76	5.01	75.85	7.84
+GCR	75.83	5.30	76.04	8.64

Table 4 Prediction results on the Amazon (Movies and TV) dataset

	Item		Item&User	
	AUC	RelaImpr (%)	AUC	RelaImpr (%)
Wide & Deep	88.44	0.00	87.43	0.00
+GREEN	92.58	10.77	94.51	18.92
+GCR	92.60	10.82	94.69	19.40
PNN	89.33	0.00	89.40	0.00
+GREEN	93.19	9.81	96.14	17.11
+GCR	93.25	9.97	96.14	17.11
Deep Crossing	89.50	0.00	88.46	0.00
+GREEN	92.54	7.70	94.73	16.30
+GCR	92.55	7.72	94.83	16.56
DeepFM	90.13	0.00	89.27	0.00
+GREEN	93.04	7.25	95.24	15.20
+GCR	93.06	7.30	95.32	15.40
DIN	89.47	0.00	88.60	0.00
+GREEN	93.37	9.88	94.84	16.17
+GCR	93.45	10.08	95.03	16.66

Table 5 Prediction results of comparative study on three CTR datasets

Amazon (Electro)	Item		Item&User	
	AUC	RelaImpr (%)	AUC	RelaImpr (%)
MLP	85.78	0.00	84.92	0.00
+DUSIN	86.98	3.35	87.14	6.36
+DDIL	87.12	3.75	85.91	2.84
+GREEN	89.28	9.78	93.02	23.20
+DUSIN +GREEN	91.53	16.07	94.32	26.92
+DDIL +GREEN	88.33	7.13	93.08	23.37
MovieLens				
MLP	71.65	0.00	70.69	0.00
+DUSIN	71.01	−2.96	70.30	−1.88
+DDIL	75.08	15.84	74.93	20.49
+GREEN	72.90	5.77	71.67	4.74
+DUSIN +GREEN	72.95	6.00	72.12	6.91
+DDIL +GREEN	75.95	19.86	76.29	27.07
Amazon (Movies and TV)				
MLP	88.53	0.00	87.15	0.00
+DUSIN	89.22	1.79	87.80	1.75
+DDIL	89.82	3.35	86.13	−2.75
+GREEN	92.49	10.28	94.56	19.95
+DUSIN +GREEN	93.88	13.89	95.35	22.07
+DDIL +GREEN	94.79	16.25	95.07	21.32

Table 6 Historical behavior truncation experimental results on Amazon (Electro)

$m(r\%)$	20(90.10%)	12(80.49%) 6(54.47%)	9(71.90%) 5(44.25%)	7(61.92%) 4(29.50%)	3(14.75%)
Wide & Deep	**85.73**	85.56	85.29	85.14	
		85.06	84.73	83.78	82.05
+GREEN	93.13(+7.4)	93.40(+7.84)	93.34(+8.05)	93.11(+7.97)	
		93.16(+8.10)	**93.51**(+8.78)	92.46(+8.68)	91.66(+**9.61**)
PNN	**86.37**	86.24	86.06	85.81	
		85.69	85.31	84.50	83.24
+GREEN	95.86(+9.49)	96.19(+9.95)	96.37(+10.31)	96.51(+10.70)	
		96.85(+11.16)	97.16(+11.85)	97.72(+13.22)	**98.13**(+**14.89**)
Deep Crossing	**86.23**	86.15	85.96	85.82	
		85.41	85.26	84.43	82.94
+GREEN	93.00(+6.77)	92.99(+6.84)	92.87(+6.91)	93.15(+7.33)	
		93.37(+7.96)	94.07(+8.81)	93.86(+9.43)	**94.52**(+**11.58**)
DeepFM	**86.65**	86.57	86.60	86.13	
		86.04	85.56	84.83	83.69
+GREEN	95.22(+8.57)	95.69(+9.12)	95.89(+9.29)	96.24(+10.11)	
		96.49(+10.45)	96.96(+11.40)	97.19(+12.36)	**97.63**(+**13.94**)
DIN	**87.00**	86.65	86.67	86.25	
		85.98	85.52	84.58	83.15
+GREEN	94.08(+7.08)	94.29(+7.64)	94.24(+7.53)	94.44(+7.53)	
		94.66(+8.46)	94.61(+9.09)	**94.98**(+10.40)	94.36(+**11.21**)

4.5 Graph convolution order study

We conduct experiments on Amazon (Electro) about different convolution orders k on the graph convolution architecture and the GREEN architecture, respectively, and the experimental results are shown in Fig. 7. Graph convolution method utilizes the final order as node representations after multiple aggregation, where GREEN introduces adaptive order-wise weights among different orders. On the item co-occurrence graph, when the order k is more than 3, the accuracy of the graph convolution method decreases due to over-smoothing, while the accuracy of GREEN has been continuously improved with the increasing order. On the two graphs, the accuracy of the graph convolution method has been maintained at a low level with the increase of k, where GREEN achieves considerable performance by learning excellent adaptive order-wise weights among multiple orders.

4.6 Overfitting analysis

Figure 8 illustrates the trend of training loss and test loss on Amazon (Electro). Compared with base models, GREEN leads to a rapid drop in loss value, which is significantly lower than the original. It can be seen from Fig. 8, when the number of training steps reaches 160,000 to 180,000, the training loss of almost all models decreased, while their test loss increased, which shows that there is an overfitting phenomenon. By introducing GREEN and GCR, the phenomenon is alleviated effectively, and the degree of separation between

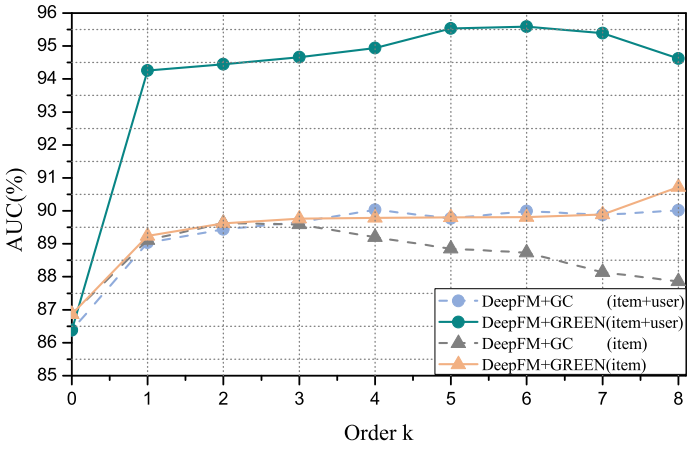


Fig. 7 The relation curve between order k and AUC on Amazon (Electro), where GC represents the graph convolution method without adapted order-wise weights

train and test loss curves is reduced, which proves that it can alleviate the phenomenon of overfitting. Moreover, GCR further reduces the minimum loss to obtain better accuracy base on GREEN.

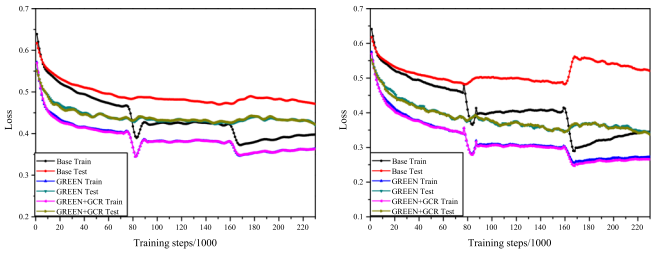
4.7 Application analysis

The inference time for the test set and the trainable parameter quantity of our models are shown in table 7. The inference time is measured in a single NVIDIA GTX 2080Ti GPU. Neither GREEN nor GCR will significantly increase the number of learnable parameters. GREEN sacrifices a certain amount of time to bring significant performance improvement, which promotes the accuracy of CTR prediction to a new level. Moreover, due to the independence of GCR, it does not affect the inference time during the test process.

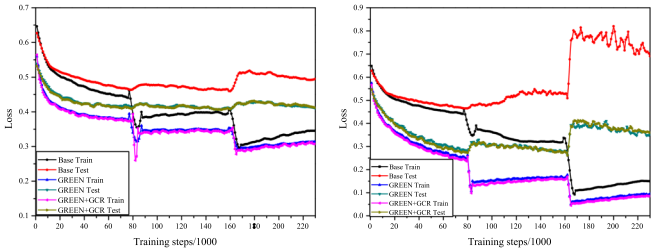
5 Related work

5.1 Deep CTR

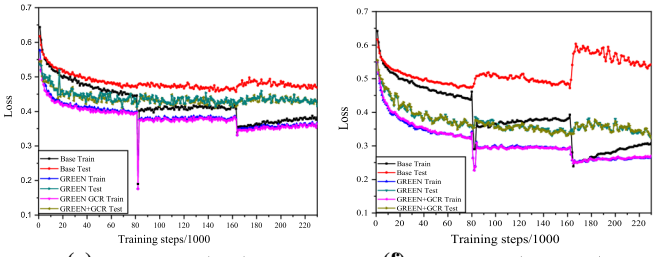
Many methods for feature interaction appeared in the field of CTR prediction, such as logistic regression (LR) [29], factorization machine (FM) [28], or field-aware factorization machines (FFM) [13]. Benefiting from the advantages of deep learning, some combined models based on deep neural networks have greatly improved the accuracy of CTR prediction. Product-based neural network (PNN) [27] utilizes product layers for feature intersection. Wide & Deep architecture [3] takes both memory ability and generalization ability of the model into account. DeepFM [7] uses a factorization machine to enhance the capability of feature interaction. Deep interest network (DIN) [44] introduces the attention mechanism to mine users' interest, and deep interest evolution network (DIEN) [43] further excavates the transfer of users' interest to assist prediction. Our optimization method for the embedding layer is universal and compatible with all the above models.



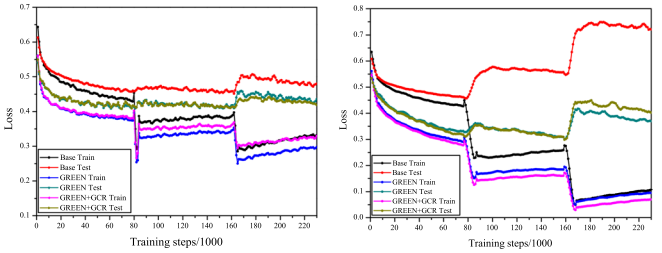
(a) Wide & Deep(item) **(b)** Wide & Deep(item+user)



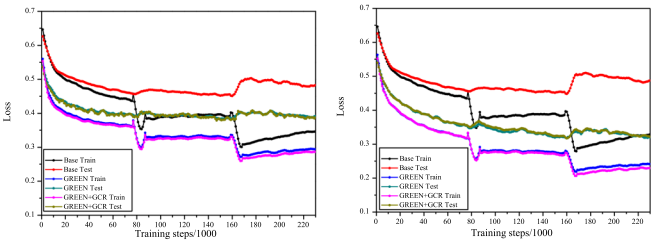
(c) PNN(item) **(d)** PNN(item+user)



(e) Deep Crossing(item) **(f)** Deep Crossing(item+user)



(g) DeepFM(item) **(h)** DeepFM(item+user)



(i) DIN(item) **(j)** DIN(item+user)

Fig. 8 The curve of loss on Amazon (Electro)

Table 7 The inference time for the test set and the trainable parameter quantity of our models for Amazon (Electro)

Model	Times(s)	Trainable Params
Wide & Deep	5.960	28.815M
+GREEN($k = 1$)	23.980	28.827M
+GREEN($k = 2$)	37.305	28.827M
+GREEN($k = 3$)	52.966	28.828M
+GREEN($k = 4$)	60.050	28.829M
+GCR	61.246	28.849M
PNN	5.809	28.826M
+GREEN($k = 1$)	21.747	28.826M
+GREEN($k = 2$)	32.490	28.827M
+GREEN($k = 3$)	48.207	28.828M
+GREEN($k = 4$)	60.318	28.829M
+GCR	61.170	28.849M
Deep Crossing	7.913	28.923M
+GREEN($k = 1$)	20.659	29.088M
+GREEN($k = 2$)	34.208	29.089M
+GREEN($k = 3$)	47.789	29.090M
+GREEN($k = 4$)	60.914	29.091M
+GCR	61.845	29.111M
DeepFM	7.892	28.815M
+GREEN($k = 1$)	23.671	28.826M
+GREEN($k = 2$)	33.735	28.827M
+GREEN($k = 3$)	47.778	28.828M
+GREEN($k = 4$)	59.766	28.828M
+GCR	59.055	28.849M
DIN	9.755	28.869M
+GREEN($k = 1$)	22.956	28.881M
+GREEN($k = 2$)	37.114	28.882M
+GREEN($k = 3$)	47.106	28.882M
+GREEN($k = 4$)	59.447	28.883M
+GCR	60.337	28.904M

5.2 Graph neural network

Graph neural networks (GNNs) received unprecedented attention in recent years because of its efficient performance [35], such as graph convolutional networks (GCNs) [15], graph attention networks (GATs) [31], and GraphSAGE [8]. They are based on the methods of neighbor aggregation to integrate the node information to optimize downstream tasks [45]. Graph-based tasks have been expanded to include representation learning[16] [4], clustering [1, 19], and link prediction [2], etc. We design the GNN framework to optimize the learning ability of the embedding layer and introduce various skills to minimize the inference time and solve the over-smoothing problem.

On the other side, graph contrastive learning prospers in the field of graph embedding. Deep graph infomax (DGI) [32] introduces the work of deep infomax (DIM) [12] into the

graph field. DGI constructs negative samples through feature shuffle and learns better node embeddings by maximizing mutual information between local representations and global graph representations. Inspired by this, we propose a graph contrastive regularization method for the deep CTR model to maintain a certain graph structure and suppress the overfitting problem.

5.3 Graph on recommendation

Recommender systems [22, 37] use certain algorithms to solve the problem of information overload, and filter out different candidate sets for different users quickly and individually, which are mostly used in search engines, movies [17, 38], e-commerce [42], and other fields. Graph learning has a wide range of meaningful applications [39] [5], and researchers have tried to introduce them into the recommendation field. Taking collaborative filtering as an example, its core information, the sparse user-item matrix, is a ready-made graph structure. Therefore, the inherent information can be mined in the form of graphs, such as NGCF [33] and the faster and lighter Light-GCN [11].

In the field of CTR, graph intention network (GIN) [18] utilizes historical click behaviors to construct the co-occurrence graph of items, and uses the GAT to aggregate neighbor nodes to solve the problems of sparseness and weak generalization. However, GIN leaves a lot to be desired, such as underutilized relationship information, considerable parameters, and slow convergence speed. Our work is mainly to carry on a series of research and optimization to solve the weakness of the graph method in the field of CTR prediction.

6 Conclusion

In this paper, we offer the guidance of graph construction with interpretability, introducing graph learning methods into the field of CTR prediction. To take advantage of the prior relationship mined in the graphs, we propose a novel embedding framework named graph relation embedding network (GREEN), which utilizes multi-order graph convolution and adaptive order-wise weighting to aggregate information for a more reasonable feature space. Moreover, a graph contrastive regularization (GCR) module is designed to further normalize graph embedding by maintaining certain graph information. We conduct extensive experiments and the results verify that our methods can achieve considerable performance improvement to promote the accuracy of CTR prediction to a new level. In future work, the methods of efficient GNN and lightweight graph construction will bring more application prospects to the application of graphs in the CTR field.

Acknowledgements This work was partially supported by the Key Research Project of Zhejiang Province (No. 2022C01145) and the National Science Foundation of China (No. U20A20173 and No. 62125206).

References

1. Bo D, Wang X, Shi C, Zhu M, Lu E, Cui P (2020) Structural deep clustering network. In: Proceedings of the Web conference 2020, pp 1400–1410
2. Chen H, Yin H, Sun X, Chen T, Gabrys B, Musial K (2020) Multi-level graph convolutional networks for cross-platform anchor link prediction. arXiv preprint arXiv:2006.01963

3. Cheng H-T, Koc L, Harmsen J, Shaked T, Chandra T, Aradhye H, Anderson G, Corrado G, Chai W, Ispir M et al (2016) Wide & deep learning for recommender systems. In: Proceedings of the 1st workshop on deep learning for recommender systems, pp 7–10
4. Cui G, Zhou J, Yang C, Liu Z (2020) Adaptive graph encoder for attributed graph embedding. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 976–985
5. Fan W, Ma Y, Li Q, He Y, Zhao E, Tang J, Yin D (2019) Graph neural networks for social recommendation. In: The World Wide Web conference, pp 417–426
6. Fawcett T (2006) An introduction to roc analysis. Pattern Recognit Lett 27(8):861–874
7. Guo H, Tang R, Ye Y, Li Z, He X (2017) Deepfm: a factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247
8. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Advances in neural information processing systems, pp 1024–1034
9. Harper FM, Konstan JA (2015) The movielens datasets: history and context. ACM Trans Interact Intell Syst (tiis) 5(4):1–19
10. He R, McAuley J (2016) Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proceedings of the 25th international conference on world wide web, pp 507–517
11. He X, Deng K, Wang X, Li Y, Zhang Y, Wang M (2020) Lightgcn: simplifying and powering graph convolution network for recommendation. arXiv preprint arXiv:2002.02126
12. Hjelm RD, Fedorov A, Lavoie-Marchildon S, Grewal K, Bachman P, Trischler A, Bengio Y (2018) Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670
13. Juan Y, Zhuang Y, Chin W-S, Lin C-J (2016) Field-aware factorization machines for ctr prediction. In: Proceedings of the 10th ACM conference on recommender systems, pp 43–50
14. Kim K, Kwon E, Park J (2021) Deep user segment interest network modeling for click-through rate prediction of online advertising. IEEE Access 9:9812–9821
15. Kipf TN, Welling M (2016a) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
16. Kipf TN, Welling M (2016b) Variational graph auto-encoders. arXiv preprint arXiv:1611.07308
17. Li D, Liu H, Zhang Z, Lin K, Fang S, Li Z, Xiong NN (2022) Carm: Confidence-aware recommender model via review representation learning and historical rating behavior in the online platforms. Neurocomputing 455:283–296
18. Li F, Chen Z, Wang P, Ren Y, Zhang D, Zhu X (2019) Graph intention network for click-through rate prediction in sponsored search. *In*: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 961–964
19. Li X, Hu Y, Sun Y, Hu J, Zhang J, Qu M (2020) A deep graph structured clustering network. IEEE Access
20. Li Z, Liu H, Zhang Z, Liu T, Xiong NN (2021) Learning knowledge graph embedding with heterogeneous relation attention networks. IEEE Trans Neural Netw Learn Syst
21. Liu H, Fang S, Zhang Z, Li D, Lin K, Wang J (2021) Mfdnet: collaborative poses perception and matrix fisher distribution for head pose estimation. IEEE Trans Multim
22. Liu H, Zheng C, Li D, Shen X, Lin K, Wang J, Zhang Z, Zhang Z, Xiong NN (2020) Edmf: efficient deep matrix factorization with review feature learning for industrial recommender system. IEEE Trans Ind Inform
23. Liu T, Liu H, Li Y-F, Chen Z, Zhang Z, Liu S (2019) Flexible ftir spectral imaging enhancement for industrial robot infrared vision sensing. IEEE Trans Ind Inform 16(1):544–554
24. Liu T, Liu H, Li Y, Zhang Z, Liu S (2018) Efficient blind signal reconstruction with wavelet transforms regularization for educational robot infrared vision sensing. IEEE/ASME Trans Mechatron 24(1):384–394
25. Mnih A, Kavukcuoglu K (2013) Learning word embeddings efficiently with noise-contrastive estimation. Adv Neural Inform Process Syst 26:2265–2273
26. Pironkov G, Dupont S, Dutoit T (2016) Speaker-aware long short-term memory multi-task learning for speech recognition. In: 2016 24th European signal processing conference (EUSIPCO). IEEE, pp 1911–1915
27. Qu Y, Cai H, Ren K, Zhang W, Yu Y, Wen Y, Wang J (2016) Product-based neural networks for user response prediction. In: 2016 IEEE 16th international conference on data mining (ICDM). IEEE, pp 1149–1154
28. Rendle S, Schmidt-Thieme L (2010) Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings of the third ACM international conference on Web search and data mining', pp 81–90

29. Richardson M, Dominowska E, Ragno R (2007) Predicting clicks: estimating the click-through rate for new ads. In: Proceedings of the 16th international conference on World Wide Web, pp 521–530
30. Shan Y, Hoens TR, Jiao J, Wang H, Yu D, Mao J (2016) 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
31. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio, Y (2017) 'Graph attention networks. arXiv preprint arXiv:1710.10903
32. Veličković P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD (2018) Deep graph infomax. arXiv preprint arXiv:1809.10341
33. Wang X, He X, Wang M, Feng F, Chua T-S (2019) Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 165–174
34. Wu F, Zhang T, Souza Jr, AHd, Fifty C, Yu T, Weinberger KQ (2019) Simplifying graph convolutional networks. arXiv preprint arXiv:1902.07153
35. Xu K, Hu W, Leskovec J, Jegelka S (2018) How powerful are graph neural networks? arXiv preprint arXiv:1810.00826
36. Yan L, Li W-J, Xue G-R, Han D (2014) Coupled group lasso for web-scale ctr prediction in display advertising. In: International conference on machine learning, pp 802–810
37. Yi B, Shen X, Liu H, Zhang Z, Zhang W, Liu S, Xiong N (2019) Deep matrix factorization with implicit feedback embedding for recommendation system. IEEE Trans Ind Inf 15(8):4591–4601
38. Zhang H, Ji Y, Li J, Ye Y (2015) A triple wing harmonium model for movie recommendation. IEEE Trans Ind Inf 12(1):231–239
39. Zhang J, Shi X, Zhao S, King I (2019) Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. arXiv preprint arXiv:1905.13129
40. Zhang X, Wang Z, Du B (2021) Deep dynamic interest learning with session local and global consistency for click-through rate predictions. IEEE Trans Ind Inf
41. Zhang Z, Li Z, Liu H, Xiong NN (2020) Multi-scale dynamic convolutional network for knowledge graph embedding. IEEE Trans Knowl Data Eng
42. Zhao WX, Li S, He Y, Chang EY, Wen J-R, Li X (2015) Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. IEEE Trans Knowl Data Eng 28(5):1147–1159
43. Zhou G, Mou N, Fan Y, Pi Q, Bian W, Zhou C, Zhu X, Gai K (2019) Deep interest evolution network for click-through rate prediction. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 5941–5948
44. Zhou G, Zhu X, Song C, Fan Y, Zhu H, Ma X, Yan Y, Jin J, Li H, Gai K (2017) 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
45. Zhou J, Cui G, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2018) Graph neural networks: a review of methods and applications. arXiv preprint arXiv:1812.08434
46. Zhu H, Jin J, Tan C, Pan F, Zeng Y, Li H, Gai K (2017) Optimized cost per click in taobao display advertising. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 2191–2200

Yixuan Wu was born in Handan, Hebei, China, in 1999. She received the B.S. degree in computer science from Shandong University, China, in 2021. She is currently pursuing the M.S. degree in computer science from Zhejiang University, China. Her research interests include recommender system and graph-related learning.

Youpeng Hu was born in Jiujiang, Jiangxi, China, in 1997. He received the B.S. degree in automation from Hangzhou Dianzi University, China, in 2019. He is currently pursuing the M.S. degree in computer science from Shandong University, China. His research interests include graph-related learning and intelligent information processing.

Xin Xiong was born in Huaihua, Hunan, China, in 1999. She received the B.S. degree in computer science from Shandong University, China, in 2021. She is currently pursuing the M.S. degree in computer science from Nanjing University, China. Her research interests include graph embedding.

Xunkai Li was born in Harbin, Heilongjiang, China, in 2000. He is currently pursuing the B.S. degree in computer science from Shandong University, China. His research interests include graph machine learning and federated learning.

Ronghui Guo was born in Ganzhou, Jiangxi, China, in 2000. He is currently pursuing the B.S. degree in computer science with Shandong University, China. His research interests include graph-related learning and intelligent information processing.

Shuiguang Deng is a full professor at the College of Computer Science and Tech logy in Zhejiang University. He received the BS and PhD both in Computer Science from Zhejiang University in 2002 and 2007, respectively. His research interests include Service Computing, Mobile Computing, and Edge Computing. Up to now he has published more than 100 papers in journals such as IEEE TOC, TPDS, TSC, TCYB, and TNNLS, and refereed conferences. He is the Associate Editor of the journal IEEE Trans. on Services Computing and IET Cyber-Physical Systems: Theory & Applications. He is a senior member of IEEE.