

GCN-Int: A Click-Through Rate Prediction Model Based on Graph Convolutional Network Interaction

YUCHEN LIU¹, CHUANZHEN LI¹, HAN XIAO¹, AND JUANJUAN CAI²

¹School of Information and Communication Engineering, Communication University of China, Beijing 100024, China

²Key Laboratory of Media Audio and Video, Ministry of Education, Communication University of China, Beijing 100024, China

Corresponding author: Juanjuan Cai (caijuanjuan@cuc.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFF0305300.

ABSTRACT Recommendation system has drawn growing attention in the academia and industry because it can solve the problem of information overload. Among a variety of methods, the click-through rate prediction model plays an important role in predicting user's attention to a specific item. To predict click-through rate, high-dimensional and sparse features are usually adopted, and the accuracy of the prediction result depends on the combination of high-order features to a great extent. Therefore, many methods have been proposed to find the low-dimensional representation from sparse high-dimensional original features, and the meaningful way of feature combination has also been mined to improve the accuracy of the model. However, the click-through rate prediction models generally have two problems. One is that they can't extract the feature interaction of non-Euclidean features very well. Another one is that it is hard to explain the inward meaning of feature interaction. In this paper, a GCN-int model based on the interaction of Graph Convolutional Network is proposed to solve the above problems. The proposed model simplifies the complex interaction among multiple features, gets a better representation of the interaction between high-order features, and improves the interpretability of feature interaction. The experimental results on the public movie recommendation dataset and our own IPTV movie recommendation dataset show that the proposed GCN-int model gets higher accuracy and efficiency compared with the state-of-the-art models.

INDEX TERMS Recommendation system, GCN-int, IPTV platform data, click-through rate prediction, feature interaction.

I. INTRODUCTION

Recommendation systems can be traced back to Tapestry [1], a collaborative filtering email system designed by Xerox Corporation in 1992 to solve the problem of information overload at the Palo Alto Research Center. In the same year, Goldberg introduced the concept of "Recommendation System". By predicting users' preferences, recommendation system can alleviate the pressure of information overload. To help people get the effective information they need, recommendation systems are widely applied in e-commerce platforms, e-government platforms, and entertainment websites, such as book recommendations, music recommendations, and movie recommendations. Moreover, recommendation system has been also adopted in the Internet Protocol Television (IPTV)

The associate editor coordinating the review of this manuscript and approving it for publication was Huiyu Zhou.

platform since embedded data collection software makes it possible to analyze the viewing patterns of TV viewers.

"Recall and Ranking" is the mainstream architecture of the recommendation system. Recall strategies can be broadly classified into the following two categories: rule-based recall and vector-based recall. Based on the recall strategy, the recall layer quickly obtains the candidate sets of recommended items from huge amounts of data. Then, in the ranking layer, these candidate sets are further sorted to get the final recommended items. In general, the candidate sets are obtained in the recall layer according to the users' personalized preference, and then sorted in the ranking layer by predicting the click rate of items, so the model used in the ranking layer may not predict for specific users. Click-through rate (CTR) prediction model is the widely used ranking model, which can predict the probability of one user clicking on a movie. In the early stage, CTR was applied to search advertising [2].

Due to the excellent prediction ability of CTR, it has been widely used in the recommendation system. In the field of CTR prediction, people have been trying to find low-dimensional representations of sparse high-dimensional original features and their meaningful combinations to improve the accuracy of the model. However, there are two problems in CTR prediction: most of the existing CTR prediction models can't well extract the interaction between high-order features and feature interaction is not explainable. In addition, the data processed by Pattern Recognition, Speech Recognition, Natural Language Processing, and other technologies emerging in recent years are mostly in Euclidean space, which have spatial continuity and can be represented by a matrix. Euclidean structure data can be easily extracted using Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN) networks. However, in click-through rate prediction, features are mostly multi-field categorical features, which don't have spatial or temporal continuity, and it is difficult to carry out feature interaction and feature extraction for such unstructured features. The emergence of Graph Convolutional Network (GCN) makes it possible to extract features from non-Euclidean data. By introducing GCN into the click-through rate prediction model, the above problems can be solved.

The contributions of this work are summarized as follows:

- 1) We proposed a click-through rate prediction model based on Graph Convolutional Network interaction (GCN-int), which can well extract the high-order features in non-Euclidean space.
- 2) We build an IPTV dataset on the IPTV platform data, and perform experiments on the IPTV dataset and two public datasets for movie recommendation. Experimental results demonstrate the effectiveness and interpretability of GCN-int.

The rest of this paper is organized as follows. Section II discusses related work about recommendation system, click-through rate prediction model, and Graph Convolutional Network. Section III presents the three modules of GCN-int. Section IV briefly introduces the datasets and evaluation metrics used in the experiments. Section V evaluates the performance of GCN-int on the above datasets and discusses the reasons for the excellent performance of the model and the existing shortcomings. Finally, Section VI concludes this paper.

II. RELATED WORK

A. RECOMMENDATION SYSTEM

Traditionally, recommendation systems can be divided into three broad categories: content-based recommendation systems, collaborative filtering recommendation systems, and hybrid recommendation systems [3].

Content-based recommendation algorithm [4] is one of the earliest recommendation algorithms, which is widely used. This algorithm recommends products that are similar to users' favorite items. Content-based recommendation algorithm has good interpretability and is not restricted

by the cold start problem, but it has the shortcoming of relying on artificial features and lack of novelty of recommendation results [5]. Collaborative filtering algorithm groups users based on different preferences and recommends products of similar types. This algorithm can be further divided into three categories: user-based collaborative filtering algorithm [6], item-based collaborative filtering algorithm [7] and model-based collaborative filtering algorithm. Compared with content-based recommendation algorithm, collaborative filtering can recommend a variety of non-text resources and mine the potential interests of users. However, collaborative filtering algorithm has the disadvantages of cold start and data sparsity [8]. In practice, hybrid recommendation system is widely used. Hybrid recommendation system aims at improving the accuracy of recommendation by combining various recommendation algorithms. The common hybrid methods are weighted, cascaded, and so on [9]. Although the hybrid recommendation algorithm improves the accuracy of recommendation system, it also increases the computational complexity [10]. In addition, how to deal with the cooperation between different modules is also one of the most difficult problems of hybrid recommendation algorithm.

B. CLICK-THROUGH RATE PREDICTION MODEL

Click-through rate prediction has gradually become the focus of the industry community and academia community in recent years. Early click-through rate prediction tasks mostly adopted Logistic Regression (LR) [11], which has become the most widely used click-through prediction model in the industry for its simplicity and high performance. However, LR lacks the ability to learn feature interaction [12]. To overcome this limitation, Oentaryo *et al.* [13] proposed a click-through rate prediction model based on Factorization Machine (FM). This model considers not only the information of single feature, but also the information of second-order feature interaction, which are represented by the dot product of two latent vectors. The emergence of FM solves the shortcoming of insufficient parameter learning caused by sparse data in LR. However, the defect of FM is that each feature only learns one unique latent vector [14]. In fact, when combined with features of different fields, the distributions of latent vectors may be different. On the basis of FM, Duan *et al.* [15] proposed Field-Aware Factorization Machine (FFM), which introduced the concept of "Field" into feature combination. The basic idea is to divide features into several fields, and each feature will learn different latent vectors for different fields. FM and FFM add the learning of second-order feature combination based on LR linear model, and the above three models are called click-through rate prediction schemes based on the shallow model. They have the advantages of simple structure and good prediction performance. What's more, it's convenient to deploy them on a large scale in industry. The disadvantage is that it is difficult to automatically extract the information carried by high-order feature combinations [16].

As Deep Learning has achieved great success in Computer Vision [17], Speech Recognition [18], Natural Language Processing [19] and other fields, its ability to explore high-order implicit information between features has also been applied to the click-through rate prediction field. Factorization Machine Supported Neural network (FNN) is the early influential click-through rate prediction model based on deep learning, which was proposed by Zhang *et al.* [16]. The dense vectors of each feature are obtained through FM model pre-training, and then input into DNN directly after splicing. This is how FNN achieves high-order feature interaction. In recent years, The combination of FM and DNN has become the mainstream method in CTR prediction. There are two main ways to combine FM and DNN: parallel architecture and serial architecture. Wide&Deep [20] proposed by Google in 2016 belongs to the parallel structure, which combines the advantages of the Wide model and the Deep model. Since the Wide part is realized by LR, which still relies on feature engineering, Wide&Deep is not a complete end-to-end model. In order to solve this problem, Google proposed Deep&Cross Network (DCN) [21] in 2017, which can combine features explicitly at will. Qu *et al.* [22] believed that, in FNN, the method of directly inputting FM learned features into DNN could not adequately learn the feature interaction. They proposed Product-based Neural Network (PNN), which is a serial fusion scheme based on multiplication to reflect the interaction of features. On the basis of FNN, Attention Neural Factorization Machines (AFM) [23] takes advantage of the Attention mechanism [24], which has achieved great success in Natural Language Processing and other fields in recent years, to solve the click-through rate prediction problem.

C. GRAPH CONVOLUTIONAL NETWORK

Graph Convolutional Network was proposed by Kipf and Welling [25], which provides a new idea for the processing of graph-structured data, and applies the convolution method of images to the graph-structured data with topological relations. CNN and RNN are suitable for one-dimensional sequence data and two-dimensional matrix data, but they cannot extract the features of non-Euclidean structure data. Data drawn from practical tasks such as social networks and recommendation systems are graph-structured data, which belong to Euclidean structure data. Graph Convolutional Network is devoted to extracting features from graph structure data. By learning the interaction of each node in the graph structure in the way of propagation, the information transmission of multi-level neighbors can be carried out. This process is realized by the superposition of several layers of neural networks.

In order to explore the graph structure more comprehensively and completely, Yang *et al.* [26] proposed the Shortest Path Graph Attention Network (SPAGAN). This model conducts path-based attention that explicitly accounts for the influence of a sequence of nodes yielding the minimum cost, or the shortest path, between the center node and its higher-order neighbors, so it can further aggregate

the information from distant neighbors more effectively. Although SPAGAN model can effectively mine the high-order relationships of graphs, in most cases, multiple heterogeneous relationships of graphs are mixed into a single edge, resulting in inaccurate features learned by the model. To solve this problem, Yang *et al.* [27] proposed Factorizable Graph Convolutional Network (FactorGCN), which achieves graph convolution through graph-level disentangling. FactorGCN explicitly disentangles the coding of intertwined relations in a graph, leading to better predictions.

The key issue in click-through prediction is how to make an effective combination of features to improve the accuracy of the model prediction. As mentioned above, many scholars have proposed a variety of methods for feature interaction, but there are still two problems in the current click-through rate prediction model. One is that it is difficult to well extract the feature interaction of non-Euclidean features. The other is that it is hard to explain the inward meaning of feature interaction. The introduction of GCN into the click-through prediction model helps to solve the above problems.

In this paper, we propose a click-through rate prediction model based on Graph Convolutional Network interaction (GCN-int) by combining the previous experience with the characteristics of Graph Neural Network. GCN-int can achieve more effective high-order interaction of features by using Graph Convolutional Network. Based on the graph structure, more flexible and explicit feature interaction can be realized, which improves the interpretability of feature interaction.

III. CTR PREDICTION MODEL BASED ON GCN INTERACTION

In this section, we present a CTR prediction model based on GCN interaction (GCN-int). Fig. 1 shows the architecture of GCN-int. The proposed model consists of three modules: input and embedding module, feature interaction module, and prediction module.

A. INPUT AND EMBEDDING MODULE

In click-through rate prediction, item features can be divided into numerical features and categorical features. Numerical features can be input into the model directly, but categorical features need to be processed before inputting into the model. For categorical features, the input data will become high-dimensional, multi-field, sparse vectors after one-hot encoding. If these vectors are fed directly into the model, they will not only result in the waste of computational resources, but will also lead to insufficient parameter learning and reduce the accuracy of the model. The common solution is to transform the features of different fields into embedding vectors, thus reducing the dimension of high-dimensional sparse vectors to low-dimensional dense features. The embedding vector is

$$E = [e_1, e_2, e_3, \dots, e_m], \quad (1)$$

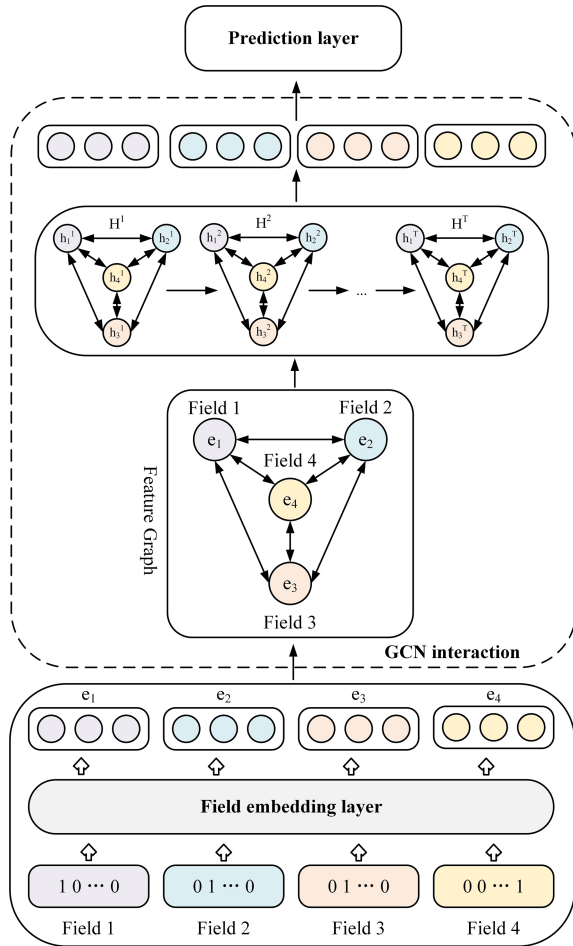


FIGURE 1. The architecture of GCN-int.

where e_i denotes the embedding vector of the i^{th} field. m represents the number of fields. Through the field embedding module, the previously high-dimensional sparse vectors are converted to the low-dimensional dense vectors, which can be input into the model for calculation.

B. FEATURE INTERACTION MODULE

The effective way of feature interaction can save the cost of manual feature engineering and improve the accuracy of the model prediction. The Logistic Regression model assumes that each feature is independent and does not consider the correlation between features. However, the two features are correlated with each other. Take the movie as an example. Generally speaking, females prefer romantic movies, love movies, and warm movies, whereas males prefer action movies, martial arts movies, and horror movies. It means that there is a strong correlation between woman feature and romantic feature, and a strong correlation between man feature and action feature. So how to discover the correlation between features and capture the interaction between features has become a key issue. Compared with the previous models, which simply concatenate the embedding vectors together and input them into the designed model for feature

interaction, we propose a special feature interaction method. We represent the features of each field as the graph structure and then use Graph Convolutional Network for feature interaction.

GCN is the first-order approximation of spectral convolution [28], which is composed of multi-layer convolutional networks. Each convolutional layer only deals with first-order neighborhood information, and information transmission of high-order neighborhoods can be realized by stacking multiple convolutional layers. The propagation way of each convolutional layer in GCN is shown as follows:

$$H^{(T+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(T)} W^{(T)} \right). \tag{2}$$

In (2), \tilde{D} is a diagonal degree matrix $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $H^{(T)}$ is the activation unit matrix in the layer T , and $H^{(0)} = E$. $W^{(T)} \in \mathbb{R}^{n_{field} \times n_{hidden}}$ is a layer-specific trainable weight matrix, in which n_{field} is the number of fields and n_{hidden} is the embedding size of GCN hidden layer. $\sigma(\cdot)$ denotes a non-linear activation function. $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph with added self-connections, in which A is the adjacency matrix of the undirected graph, and I_N is the unit matrix. The formula of the adjacency matrix A is

$$A[n_i, n_j] = \begin{cases} w(n_i, n_j) & \text{if } i \neq j \\ 0 & \text{else} \end{cases}, \tag{3}$$

where w is the weight of the connection edge. As shown in Fig. 1, the features of each field are regarded as the nodes in the Graph Neural Network. Therefore, the interaction between each feature can be understood as the interaction of each node in the graph. In this way, the features of one item and their interactions form one feature graph. Multiple items correspond to multiple feature interaction graphs. From this point of view, using GCN for feature interaction enhances the interpretability of feature interaction. For example, we have four fields: Country, Company, Tag, and Color. And the features of movie v are as follows: China, Bona Film, Comedy, and Color. Then we give an adjacency matrix A and draw the feature interaction graph according to A . If an element of A is 1, it indicates that an edge needs to be drawn between the two features corresponding to the element. If an element in A is 0, there is no edge between the two features. In order to obtain as much information as possible, we set A to be a matrix with diagonal 0 and other elements 1. This means that all features interact with other features besides themselves. Thus, the feature interaction of movie v is shown in Fig. 2.

We can use A to control which features will be used for interaction. In this way, we can intuitively show the way of feature combination. By observing the constructed feature interaction schematic diagram, we can see how features interact within GCN. Each edge in the graph represents the combination of connected features, which explains the internal meaning of feature interaction.

Compared with Deep Neural Network, which is a black-box structure, Graph Neural Network can be used to observe which features interact. Here, according to (2), the adjacency

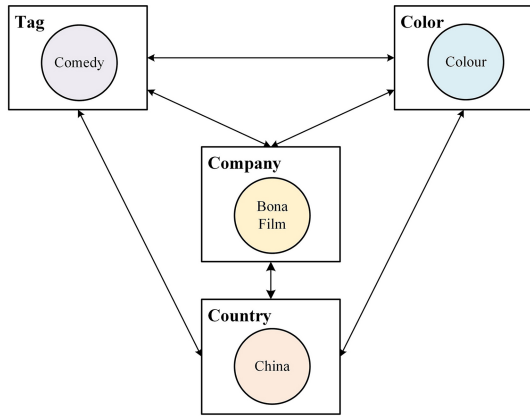


FIGURE 2. Schematic diagram of GCN feature interaction.

matrix A is defined as a matrix with all elements 1 except the main diagonal. That is, in (3), the value of w is 1. The reason why w is set as 1 is that it is more accurate to interact with all features when it is uncertain which features interaction will improve the accuracy of prediction. The depth of interaction is determined by the number of layers in the Graph Neural Network.

Through the feature interaction module, the embedding vectors of each field aggregate the state information of their neighbor nodes, which realizes the purpose of feature interaction. The output of feature interaction module is

$$H^{(T)} = [h_1^T, h_2^T, \dots, h_m^T], \quad (4)$$

where m represents the number of fields, and T is the number of convolution layers.

C. PREDICTION MODULE

The prediction module concatenates the embedding vectors of each field through the GCN feature interaction module together and obtains an $m \times D$ dimensional vector, where m is the number of fields and D is the size of embedding vector. With the sigmoid function, GCN-int model outputs the score between $[0,1]$, indicating the likelihood that user clicks:

$$\tilde{y} = \text{sigmoid} \left(W^T H^{(T)} + b \right). \quad (5)$$

In (5), H^T is the matrix of hidden representations of entities in layer T . W is the weight matrix, and b is the bias vector.

In addition, the loss function of GCN-int is Logloss, which is defined as

$$\text{Logloss} = -\frac{1}{N} \sum_{i=1}^N (y_i \log(\tilde{y}_i) + (1 - y_i) \log(1 - \tilde{y}_i)), \quad (6)$$

where N is the total number of samples, y_i is the true label of i , \tilde{y}_i is the predicted label of i . Adam optimizer is used to optimize the model and the initial learning rate l is set to 0.001. In order to prevent overfitting during training, GCN-int model also uses batch-normalization [29] and Dropout layer [30] after each layer. The dropout ratio r is set to 0.5 for training.

The activation function is RELU. What's more, Residual Network [31] is also introduced into the model, which can prevent network degradation caused by the increase in the number of layers of deep Network and ensure that the deep network has at least the same fitting ability as the shallow network.

IV. EXPERIMENTAL SETUP

We evaluate the proposed GCN-int and present its performance on movie recommendation by three datasets: Criteo, Avazu, and our IPTV dataset. In this section, we introduce the datasets and evaluation metrics used in the experiments.

A. DATASETS

In order to present the objectivity of GCN-int, we utilize two kinds of datasets in our experiments. One is the public dataset. The other is the IPTV dataset, which is built by ourselves. Criteo and Avazu are public datasets commonly used by academics to study click-through prediction models.

Criteo¹ is an online advertising dataset published by Criteo Labs. This dataset contains 40 million lines of data and the features of 39 fields, among which there are 13 numerical features and 26 categorical features. Due to the confidentiality of data, the name of each feature is not disclosed but is represented in the form of code.

Avazu² is published by Kaggle for a click-through rate prediction contest. This dataset contains the users' behavior information of clicking advertisements on mobile terminals in the past 10 days. This dataset contains the features of 23 fields, all of which are categorical features. Table 1 shows the detailed statistical information of the above two datasets.

TABLE 1. Statistics of two public datasets: Criteo and Avazu.

Dataset	Samples	Fields	Features
Criteo	45,840,617	39	998,960
Avazu	40,428,967	23	1,544,488

In addition, referring to the above two public datasets, we built the IPTV dataset from the historical data of IPTV program viewing in two months. Viewing data used in the IPTV dataset are the single episode of on-demand programs on the IPTV platform of one province. In a certain period, if one movie has been clicked by one user, the "isviewed" field is set to "1", and if it does not appear in the user's history viewing record, the "isviewed" field is set to "0". Like Criteo, according to the ID of programs, all kinds of attribute information of programs are added as auxiliary information. Based on the principle of information confidentiality, the fields containing special information are encrypted. This dataset contains 134297 lines of user interaction data and 201411 features of 13 fields, including 8 numerical features and 5 categorical features, with the positive and negative

¹<https://www.kaggle.com/c/criteo-display-ad-challenge>

²<https://www.kaggle.com/c/avazu-ctr-prediction>

TABLE 2. Field information and number of features for the IPTV dataset.

ID	Category	Name	Number of Features
1	label	isviewed	–
2	unique id	program_id	–
3	categorical feature	program_area	48
4		type_id	24
5		Status	3
6		SeriesFlag	1
7		Director	48
8		ContentProvider	45
9		base_type	3059
10		Resolution	5
11	numerical feature	release_years	53
12		LicensingWindowStart	916
13		LicensingWindowEnd	1301
14		CreateDate	99481
15		AuditDate	96427

sample ratio of 4:9. The Field information and number of features of IPTV datasets is shown in Table 2.

B. EVALUATION METRICS

To estimate the performance of the proposed model, two evaluation metrics commonly used in CTR prediction problems are adopted in our experiments: AUC [32] and Logloss [33]. The above two metrics are recognized in CTR prediction, which can evaluate the performance of the model from two different perspectives.

AUC (Area Under Curve) is defined as the area bounded by the coordinate axis under the ROC (Receiver Operating Characteristic Curve), and its value ranges between 0 and 1. The closer AUC is to 1, the better model performs. The model performs worst when the AUC is equal to 0.5, which corresponds to random guessing. Compared with other indices, the advantage of ROC is that when the distribution of positive and negative samples changes, the ROC curve can basically remain the same shape in the figure. It indicates that ROC is not sensitive to the proportion of positive and negative samples. Therefore, ROC can reduce the interference brought by different test sets and reflect the effect of the model more objectively.

Logloss is a common evaluation method for machine learning problems. The smaller the value of Logloss is, the better the model performs.

V. EXPERIMENTS AND DISCUSSIONS

A. EXPERIMENTAL RESULTS

In this section, we conduct four groups of experiments respectively: the performance of GCN-int on public datasets, the performance of GCN-int on IPTV dataset, the sensitive experiments of GCN-int, and the efficiency comparison of different models.

1) THE PERFORMANCE OF GCN-INT ON PUBLIC DATASETS

In order to objectively verify the performance of GCN-int on the public datasets, this experiment compares GCN-int with the three kinds of models mentioned above: the linear model considering only first-order feature interaction [11], the model of second-order feature interaction [23], [34] and

the model of high-order feature interaction [21], [35]–[38]. This experiment is mainly compared with AutoInt [38], which was proposed in 2019 and achieved good performance by using the multi-head attention mechanism. We set the embedding size to 8 and the number of GCN layers to 3. For Criteo and Avazu, we set the hidden embedding sizes in GCN to 64 and batch size to 1024. For the IPTV dataset, we set the hidden embedding sizes in GCN to 16 and batch size to 64.

Table 3 presents the experimental results compared GCN-int with baseline models on the public datasets. In order to show the improvement of GCN-int more intuitively, we respectively calculated the percentage improvement of GCN-int on AUC and Logloss relative to the baseline model and express them as Improve-AUC and Improve-Logloss in Table 3. The formulas of Improve-AUC and Improve-Logloss can be written as

$$\text{Improve} - \text{AUC} = \frac{AUC_{GCN-int} - AUC_{baseline}}{AUC_{baseline}}, \quad (7)$$

$$\text{Improve} - \text{Logloss} = \frac{\text{Logloss}_{GCN-int} - \text{Logloss}_{baseline}}{\text{Logloss}_{baseline}}. \quad (8)$$

The following observations can be obtained from Table 3. Compared with the traditional linear model, which only considers first-order interaction of features, AUC and Logloss of FM and AFM which consider the feature interaction are better. It shows that feature interaction is meaningful for improving the performance of click-through prediction model, and effective feature interaction can get higher accuracy. In addition, the experimental results also show the positive effect of introducing GCN into the CTR prediction model. If the GCN interaction part is removed from the GCN-int model, the model is close to LR. By comparing the experimental results of LR and GCN-int, it can be seen that the introduction of the GCN module greatly improves the prediction accuracy. On Criteo and Avazu datasets, GCN-int shows a significant improvement over the click-through rate prediction model based on deep learning high-order feature interaction proposed in recent years.

2) THE PERFORMANCE OF GCN-INT ON IPTV DATASET

This experiment focuses on comparing the performance of AutoInt and GCN-int on IPTV dataset, and the experiment results are shown in Table 4.

We observe that both AutoInt and GCN-int have a good performance on IPTV dataset, with AUC above 0.94. It also shows that the IPTV dataset constructed by ourselves has good classification characteristics. It means that in the IPTV platform, the programs watched by users have obvious classification characteristics. What's more, the model proposed in this paper is better than AutoInt.

Combined with the above two experiments, it can be shown that the proposed feature interaction method based on Graph Convolutional Network can effectively improve the accuracy of the click-through rate prediction model.

TABLE 3. The experimental results compared GCN-int with baseline models on the public datasets.

Model Class	Model	Criteo				Avazu			
		AUC	Improve-AUC	Logloss	Improve-Logloss	AUC	Improve-AUC	Logloss	Improve-Logloss
First-order	LR	0.782	3.21%	0.4695	5.37%	0.756	4.03%	0.3964	5.15%
Second-order	FM [34]	0.7836	3.00%	0.47	5.47%	0.7706	2.06%	0.3854	2.49%
	AFM [23]	0.7938	1.68%	0.4584	3.08%	0.7718	1.90%	0.3854	2.44%
High-order	DeepCrossing [35]	0.8009	0.77%	0.4513	1.55%	0.7643	2.90%	0.3889	3.32%
	NFM [36]	0.7957	1.43%	0.4562	2.61%	0.7708	2.04%	0.3864	2.69%
	DCN [21]	0.7907	2.07%	0.4591	3.22%	0.7667	2.58%	0.3868	2.79%
	xDeepFM [37]	0.8009	0.77%	0.4517	1.64%	0.7758	1.38%	0.3829	1.80%
	AutoInt [38]	0.8061	0.12%	0.4455	0.27%	0.7752	1.46%	0.3824	1.67%
	GCN-int	0.8071	-	0.4443	-	0.7865	-	0.376	-

TABLE 4. The model comparison experiments on IPTV dataset.

Model Class	Model	IPTV(our own dataset)			
		AUC	Improve-AUC	Logloss	Improve-Logloss
High-order	AutoInt	0.946	0.08%	0.2663	1.84%
	GCN-int	0.9468	-	0.2614	-

3) THE SENSITIVE EXPERIMENTS OF GCN-INT

In GCN-int, the dimension of embedding vector in the embedding module is an important parameter, which has a significant influence on the effectiveness and complexity of GCN-int. We conducted experiments on Avazu by fixing other hyper-parameters and modifying only the value of the embedding size to observe the effect of this parameter on GCN-int.

TABLE 5. The sensitive experiments of embedding size hyper-parameter.

embedding size	AUC	Logloss	time(s)
8	0.7844	0.3777	3606
16	0.7845	0.3773	5142
32	0.7856	0.3764	8036
64	0.7865	0.376	12513

Table 5 and Fig. 3 show the results of the sensitivity experiments of the hyper-parameter embedding size. We observe that embedding size is positively correlated with AUC and negatively correlated with Logloss. It shows that increasing the dimension of embedding vector has a positive impact on the efficiency of model, but also leads to the increase in training time, as it increases the complexity of the model.

Therefore, in practical application scenarios, the accuracy and complexity should be considered comprehensively, so as to provide more accurate recommendation results for users on the premise of ensuring a quick response.

4) THE EFFICIENCY COMPARISON OF DIFFERENT MODELS

The efficiency of model is also an important standard to evaluate the model. It's usually measured in two aspects, one is the number of parameters to be calculated of the model, and the other is the runtime of the model. This experiment compared the efficiency of different CTR models on Criteo. Note that the parameters calculated in this experiment don't

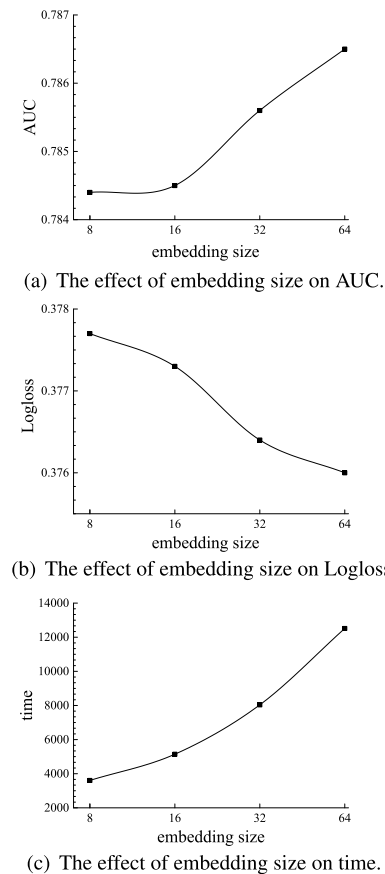


FIGURE 3. The effect of embedding size on GCN-int.

include the embedding layer. Table 6 shows the experimental results.

It can be seen that GCN-int requires fewer parameters than AutoInt, the best of all baseline models, for approximately the same running time. It proves that GCN-int is highly efficient on the premise of ensuring the accuracy of model.

In conclusion, GCN-int proposed in this paper achieves the best performance among all CTR prediction baseline models. Compared with AutoInt, which is the best effective baseline model, GCN-int requires fewer parameters and is more efficient.

TABLE 6. The efficiency comparison of different models.

Model	DeepCrossing	DCN	NFM	AutoInt	GCN-int
Parameters	1.6×10^5	3×10^3	4×10^3	3.9×10^4	2.5×10^4
Time(s)	1432	4211	1284	2457	2646

B. DISCUSSIONS

In this paper, we propose GCN-int, which is a CTR prediction model based on GCN interaction. We carry out experiments on the movie recommendation scenario, and the experimental results present that GCN-int has good performance and high accuracy, outperforming the state-of-the-art baseline models. Several reasons can explain the excellent performance of GCN-int.

First, GCN-int realizes more effective high-order feature interaction based on GCN. Compared with first-order or second-order feature interaction, high-order feature interaction contains more possible situations of multiple feature combinations, so it can improve the accuracy of the click-through rate prediction model. What's more, GCN aims at extracting features from graph-structured data, so it can make the interaction between high-order features more sufficient.

Second, feature interaction based on graph structure is more flexible. The CTR models mentioned above have similar processes in constructing high-order feature interaction. Firstly, sparse and high-dimensional features are transformed into low-dimensional and dense features, and then features are spliced together and finally input into the deep network. This simple feature splicing method limits the ability of complex interaction among features, making feature interaction in different fields can only be carried out in a fixed way. However, GCN-int can control the interaction field through the adjacency matrix, and control the interaction depth through the number of convolution layers, which can make the feature interaction mode more flexible.

Finally, presenting feature interaction in the form of graph improves the interpretability of feature interaction, which can't be simulated by other structures. Generally, DNN uses multiple hidden layers of the neural network for embedding learning, resulting in feature interactions within the neural network being learned automatically. It is invisible and uncontrollable. In contrast, GCN has obvious advantages in feature interaction. It takes features as nodes in the graph, takes interactions as edges between nodes, and displays feature interaction in the form of feature map. This form can show the interaction of multiple features of one item more intuitively.

Despite the good performance of GCN-int, it still has one drawback. The CTR prediction model based on Graph Neural Network proposed in this paper does not consider the weights of interactions between features, but carries out feature interactions with the same weight values. In practical application scenarios, the contribution of the interaction between different features to the accuracy of the model should be different. If the weights of different feature interactions can be adaptively adjusted according to the contributions during the interaction, the accuracy of GCN-int can be further improved.

VI. CONCLUSION

In this paper, we have presented a click-through rate prediction model based on Graph Convolutional Network. In GCN-int, the features of each field and the interaction between different features are represented as the feature relation graph. And high-order feature interaction is realized by Graph Convolutional Network, which improves the interpretability of feature interaction. Experiments are carried out on two public datasets and the IPTV dataset on the movie recommendation scenario. It is proved that compared with the state-of-the-art models, the proposed model has both higher accuracy and higher efficiency, and can better capture the interaction among high-order features. In addition, the model may have better performance if the weights of different feature interactions are taken into account.

REFERENCES

- [1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992, doi: [10.1145/138859.138867](https://doi.org/10.1145/138859.138867).
- [2] M. J. Liu, G. C. Zeng, W. Yue, Y. Liu, and Z. G. Qin, "A hybrid network based CTR prediction model for online advertising," *Chin. J. Comput.*, vol. 42, no. 7, pp. 1–18, 2019.
- [3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
- [4] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*. Berlin, Germany: Springer-Verlag, 2007, pp. 325–341.
- [5] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*. Boston, MA, USA: Springer-Verlag, 2011, pp. 73–105.
- [6] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work (CSCW)*, 1994, pp. 175–186.
- [7] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.
- [8] Z. Sun, Q. Guo, J. Yang, H. Fang, G. Guo, J. Zhang, and R. Burke, "Research commentary on recommendations with side information: A survey and research directions," *Electron. Commerce Res. Appl.*, vol. 37, Sep. 2019, Art. no. 100879.
- [9] K. Bhareti, S. Perera, S. Jamal, M. H. Pallegge, V. Akash, and S. Wiieweera, "A literature review of recommendation systems," in *Proc. IEEE Int. Conf. Innov. Technol. (INOCON)*, Nov. 2020, pp. 1–7.
- [10] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, Oct. 2009, doi: [10.1155/2009/421425](https://doi.org/10.1155/2009/421425).
- [11] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and scalable response prediction for display advertising," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, p. 61, 2015, doi: [10.1145/2532128](https://doi.org/10.1145/2532128).
- [12] B. Liu, R. Tang, Y. Chen, J. Yu, H. Guo, and Y. Zhang, "Feature generation by convolutional neural network for click-through rate prediction," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 1119–1129.
- [13] R. J. Oentaryo, E.-P. Lim, J.-W. Low, D. Lo, and M. Finegold, "Predicting response in mobile advertising with hierarchical importance-aware factorization machine," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, Feb. 2014, pp. 123–132.
- [14] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for CTR prediction," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 43–50.
- [15] P. Yan, X. Zhou, and Y. Duan, "E-commerce item recommendation based on field-aware factorization machine," in *Proc. Int. ACM Recommender Syst. Challenge*, Sep. 2015, pp. 1–4.
- [16] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data—A case study on user response prediction," in *Proc. Adv. Inf. Retr. 38th Eur. Conf. IR Res. (ECIR)*, 2016, pp. 45–57.

- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012, doi: [10.1145/3065386](https://doi.org/10.1145/3065386).
- [18] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
- [19] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, Nov. 2014, pp. 101–110.
- [20] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Sep. 2016, pp. 7–10.
- [21] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proc. ADKDD*, Aug. 2017, p. 12.
- [22] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 1149–1154.
- [23] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3119–3125.
- [24] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 335–344.
- [25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–14.
- [26] Y. Yang, X. Wang, M. Song, J. Yuan, and D. Tao, "SPAGAN: Shortest path graph attention network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4099–4105.
- [27] Y. Yang, Z. Feng, M. Song, and X. Wang, "Factorizable graph convolutional networks," in *Proc. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 20286–20296.
- [28] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 29, 2016, pp. 3844–3852.
- [29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 1, Jul. 2015, pp. 448–456.
- [30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [31] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2016, pp. 4278–4284.
- [32] Y. Fang and J. Liu, "A novel prior-based real-time click through rate prediction model," *Int. J. Mach. Learn. Cybern.*, vol. 5, no. 6, pp. 887–895, Dec. 2014, doi: [10.1007/s13042-014-0231-7](https://doi.org/10.1007/s13042-014-0231-7).
- [33] J. Zhao, "Research on the algorithm of click-through rate prediction," M.S. thesis, College Softw., Jilin Univ., Changchun, China, 2020.
- [34] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and testing low-degree polynomial data mappings via linear SVM," *J. Mach. Learn. Res.*, vol. 11, no. 48, pp. 1471–1490, Jan. 2010.
- [35] Y. Shan, T. R. Hoens, J. Jiao, H. Wang, D. Yu, and J. Mao, "Deep crossing: Web-scale modeling without manually crafted combinatorial features," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 255–262.
- [36] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 355–364.
- [37] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "XDeepFM: Combining explicit and implicit feature interactions for recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1754–1763.
- [38] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, "AutoInt: Automatic feature interaction learning via self-attentive neural networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1161–1170.



YUCHEN LIU received the B.S. degree in communication engineering from the Communication University of China, in 2019, where she is currently pursuing the M.S. degree in information and communication engineering. Her research interests include knowledge graph, graph convolutional networks, recommendation systems, and big data analysis.



CHUANZHEN LI received the Ph.D. degree from the Communication University of China, in 2013. She is currently an Associate Professor with the Digital Media Technology Department, Communication University of China. Her research interests include media data analysis, recommendation algorithm, and multimodal information processing.



HAN XIAO received the B.S. degree in communication engineering from the Communication University of China, in 2018, where she is currently pursuing the M.S. degree in information and communication engineering. Her research interests include recommendation systems, graph convolutional networks, click-through-rate prediction, and feature interactions.



JUANJUAN CAI received the M.S. degree from the Communication University of China, in 2007. She is currently an Associate Researcher with the Key Laboratory of Media Audio and Video, Ministry of Education, Communication University of China. Her research interests include multimedia technology, audio signal processing, and big data analysis.

...