Zhang *et al. Journal of Big Data* (2023) 10:11
https://doi.org/10.1186/s40537-023-00688-6

Journal of Big Data

RESEARCH Open Access

Click-through rate prediction model integrating user interest and multi-head attention mechanism

Wei Zhang, Yahui Han*, Baolin Yi and Zhaoli Zhang

*Correspondence:
hanyahui@mails.ccnu.edu.cn

Faculty of Artificial Intelligence in Education, Central China Normal University, 430079 Wuhan, China

Abstract

The purpose of click-through rate (CTR) prediction is to anticipate how likely a person is to click on an advertisement or item. It's required for a lot of internet applications, such online advertising and recommendation systems. The previous click-through rate estimation approach suffered from the following two flaws. On the one hand, input characteristics (such as user id, user age, user age, item id, item category) are usually sparse and multidimensional, making them effective. High-level combination characteristics are used for prediction. Obtaining it manually by domain experts takes a long time and is difficult to finish; also, customer interests are not all the same. The accuracy of the model findings will significantly increase if this immediately recognized component is incorporated in the prediction model. As a consequence, this study creates an IARM (interactive attention rate estimation model) that incorporates user interest as well as a multi-head self-attention mechanism. The deep learning network is used in the model to determine the user's interest expression based on user attributes. The multi-head self-attention mechanism with residual network is then employed to get feature interaction, which enhances the degree of effect of significant characteristics on the estimation result as well as its accuracy. The IARM model outperforms other recent prediction models in the assessment metrics AUC and LOSS, and it has superior accuracy, according to the results from the public experimental data set.

Keywords User interest, Multi-head self-attention mechanism, Residual network, Click-through rate prediction model

Introduction

The recommendation model's main purpose is to automatically suggest possible things to the user based on the user's personal and historical data [1, 2]. For example, in real-world online purchasing, a system with a flawless recommendation model might boost not just user happiness but also sales volume [3, 4]. Each individual produces unique information data as a result of the effect of gender, age, employment, and other aspects, and the recommendation model uses this data to assess the user's probable purchases. The information provided by the user then becomes crucial. The classic recommendation approach is based mostly on the attributes of a user's item

rating. The user's preferred category is decided based on the rating, and then a recommendation is issued [5, 6]. The following drawbacks are present: To begin with, many users in real life do not make assessments or intentionally praise or criticize others, instead relying on the rating feature to make suggestions. The model's precision and generalization will suffer. The inability to change is worse [7, 8]; second, every individual in life will have a label that distinguishes them from others, and that label may be discovered [9–11]. Third, users and objects have many characteristics, which are not all the same and have diverse impacts on recommendation outcomes. Traditional models pay less attention to crucial types of characteristics, resulting in a waste of valuable data as well as a reduction in model accuracy [12–14].

Machine learning and deep learning technologies have become widely employed in the application of recommendation models, thanks to the fast growth of artificial intelligence technology [15]. A number of deep learning-based models have been proposed during the last few years [16, 17]. PNN [18], xdeepFM [19], AFM [20], and a variety of other models are examples. Unlike traditional recommendation models, deep learning recommendation models can automatically capture the complex relationships within the data, as well as nonlinear interaction information between users and items, and obtain more complex and abstract high-level interactive feature representations [21, 22]. Researchers developed the hypothesis of attention mechanism after being inspired by visual attention. It instructs the neural network to concentrate exclusively on the most significant aspects of the input information, therefore giving them more weight [23, 24]. The model will be able to capture not just the user but also the item in this manner. The crucial combination of features, as well as the weight values of each feature, may be shown, ensuring that the model is easy to understand in the recommendation task [25, 26].

In conclusion, having user interests and automatically producing feature matrices with various weights can increase the accuracy of recommendation model outcomes. As a result, a merger of user interest and multi-head attention mechanism is proposed in this research as a click-through rate estimate model (IARM). The IARM model takes into consideration not just the impact of user interest on recommendation outcomes, but also feature differences. To create feature matrices with various weights, it employs a multi-head self-attention mechanism and a residual network. The model makes the following major contributions:

1. We proposes a new IARM model. The model uses deep learning and multi-head self-attention mechanism technology to automatically obtain various data information, making the data information further utilized.
2. The IARM model incorporates user interest. Expand the gap between users and improve the accuracy of the recommended results.
3. The IARM model employs a residual network and a multi-head self-attention mechanism to identify cross-feature combinations that are unworthy of weighting, allowing key characteristics to play a larger part in the recommendation process and improving the model's accuracy.
4. We ran comprehensive tests on a variety of real-world data sets. Our suggested technique not only beats existing state-of-the-art approaches for prediction, but also pro-

vides strong model explainability, according to experimental findings on the problem of CTR prediction.

The following is a breakdown of how we structure our work: We describe the relevant work in "Related work". Our model's structure was introduced in "IARM model". The experimental findings and extensive analysis are presented in "Experiment". In "Conclusion", we wrap up this study and discuss the next steps.

Related work

User interest

User interest may intuitively represent each user's unique qualities, hence it plays a critical role in the recommendation model [27–29]. For example, Google's Wide & Deep model, which combines the benefits of a linear shallow model with a deep model, employs the shallow model's memory properties to capture each user's interest. The Alibaba Company's suggested DIN [30] model combines the user's previous behavior sequence and attention mechanism to dynamically compute the user's interest changes, which increases the accuracy of the recommendation results to a degree.

Learning feature interactions

Learning feature interactions is a fundamental subject that has received a lot of research attention. Factorization Machines (FM), which were designed to primarily capture first- and second-order feature interactions and have been shown to be useful for a variety of tasks in recommender systems [31], are a well-known example. Following that, many factorization machine variations were suggested. Field-aware Factorization Machines (FFM), for instance, modeled fine-grained relationships between features from many fields. The relevance of various second-order feature interactions was studied in GBFM and AFM. All of these methods, on the other hand, are geared at simulating low-order feature interactions.

Recent research has attempted to predict high-order feature interactions. To simulate higher-order features, NFM built deep neural networks on top of the output of second-order feature interactions. Similarly, feed-forward neural networks were used to describe high-order feature interactions in PNN [32], FFM, DeepCrossing, Wide & Deep, and DeepFM. All of these methods, however, learn high-order feature interactions in an implicit manner, resulting in poor model explainability. On the other hand, there are three lines of study that explicitly learn feature interactions. First, Deep & Cross and xDeepFM took the bit-wise and vector-wise outer product of features, respectively. Although they execute explicit feature interactions, determining which combinations are advantageous is not straightforward. Second, certain tree-based techniques [33] integrated the strength of embedding-based and tree-based models, but the training procedure had to be broken down into many steps.

Self-attention and residual networks

Attention and residual networks are two of the most recent deep learning approaches used in our proposed model. Attention was initially suggested in the context of neural machine translation, and it has since been demonstrated to be useful in a range of tasks,

including question answering [34], text summarization and recommender systems. Vaswani et al. went on to suggest multi-head self-attention as a way to simulate complex word relationships in machine translation [35]. In the ImageNet competition, residual networks earned state-of-the-art results. The residual connection, which can be written as $y = F(x) + x$, promotes gradient flow over interval layers, making it a common network topology for training very deep neural networks [36, 37].

In summary, this research provides a model for estimating click-through rates that combines user interest with a multi-head attention mechanism. The model first employs deep learning technology to automatically collect each user's unique interest expression in order to build a distinction between users; next, using the multi-head attention mechanism and residual network, it obtains feature combinations with various weights. The output layer then outputs the forecast result.

IARM model

The suggested IARM approach, which can automatically learn the feature interaction for CTR prediction, is initially described in this section. Following that, this article will show how to employ the multi-head attention mechanism to learn user interest representation and model high-order combination characteristics. The model's structure is depicted in Fig. 1.

Overview

The IARM model's purpose is to transfer the user's long-term interest matrix, as well as high-order interaction characteristics and matrices with varying weight values, into a low-dimensional space. The approach suggested in this research takes the feature vector x as an input and projects all of the features into the same latitude space using an embedding layer. The interest layer then processes the user information to produce the user interest expression. To obtain a high-order cross feature matrix and features with varying weight information, input extensive field information into the interactive layer. Finally, the three feature matrices are merged to produce the final feature matrix, which is sent via the output layer.

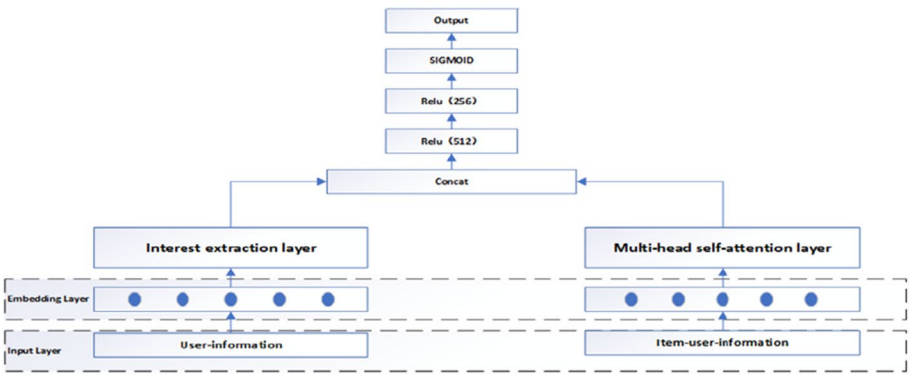

Fig. 1 Overview of our proposed model IARM

Input layer

We start with a sparse vector, which is the concatenation of all fields, to represent user profiles and item attributes. Specifically,

$$\mathbf{x} = [\mathbf{x_1}; \mathbf{x_2}; \ ...; \mathbf{x_M}], \tag{1}$$

where M is the total number of feature fields and xi is the i-th field's feature representation. If the i-th field is categorical, xi is a one-hot vector (e.g., $\times 1$ in Fig. 2). If the i-th field is numerical, xi is a scalar value (e.g., xM in Fig. 2).

Embedding layer

Because categorical feature representations are sparse and high-dimensional, converting them to low-dimensional spaces is a typical practice (e.g., word embeddings). In particular, we use a low-dimensional vector to represent each categorical feature, i.e.

$$\mathbf{e_i} = \mathbf{V_i x_i}, \tag{2}$$

where Vi is a field I embedding matrix and xi is a one-hot vector. Categorical features are frequently multi-valued, i.e., xi is a multi-hot vector. Take, for example, movie watching prediction; there might be a feature field Genre that identifies the genres of movies and can be multi-valued (e.g., Drama and Romance for the movie "Titanic"). To make Eq. (2) compatible with multi-valued inputs, we extend it and express the multi-valued feature field as the average of related feature embedding vectors:

$$\mathbf{e_i} = \mathbf{V_i x_i}, \tag{3}$$

where q is the number of values a sample has for the i-th field and xi denotes the multi-hot vector representation of this field.

We also encode numerical characteristics in the same low-dimensional feature space as category features to facilitate interaction between them. We represent the numerical characteristic as follows:

$$\mathbf{e_m} = \mathbf{v_m} x_m, \tag{4}$$

where **vm** is an embedding vector for field *m*, and *xm* is a scalar value.

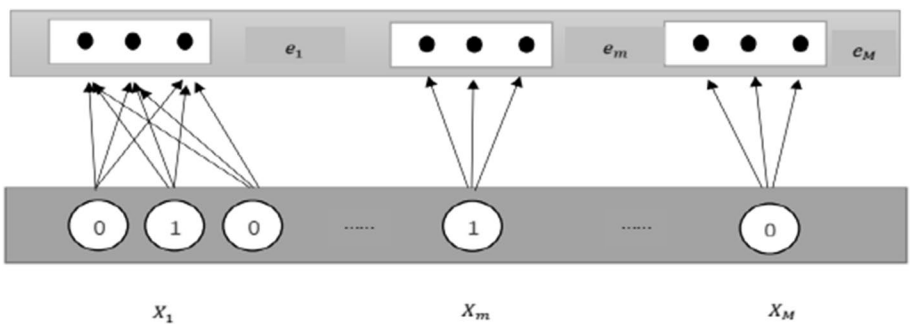

Fig. 2 Input and embedding layer illustration, with low-dimensional packed vectors representing both categorical and numerical fields

The embedding layer's output would thus be a concatenation of numerous embedding vectors, as seen in Fig. 2.

Interest acquisition layer

To begin, get the user information feature matrix, which is stated as follows:

$$\mathbf{u} = [\mathrm{u}\mathbf{1};\ \mathrm{u}\mathbf{2};\ ...;\ \mathrm{un}], \tag{5}$$

Here this article uses a multi-layer perceptron method to obtain the user's interest expression, the specific function is as follows:

$$\begin{aligned} Z_1 &= f(W_1 u + b_1) \\ Z_2 &= f(W_2 Z_1 + b_2) \\ Z_3 &= f(W_3 Z_2 + b_3) \\ U &= f(W_4 Z_3 + b_4) \end{aligned} \tag{6}$$

Among them, Zi represents the output result of each layer of the network, Wi represents the training matrix of each layer of the network, bi represents the paranoia item of each layer, f represents the relu activation function, and u represents the user. The information feature matrix of U, U reflects the user's interest feature matrix.

Interaction layer

Once the numerical and category characteristics are in the same low-dimensional space, we may move on to modeling high-order combinatorial features. The main issue is determining which characteristics should be merged to generate relevant high-order features. Traditionally, domain experts achieve this by creating meaningful combinations based on their knowledge. In this study, we address this issue using a unique approach called the multi-head self-attention mechanism.

Recently, a multi-head self-attentive network shown amazing effectiveness in modeling intricate relationships. For example, it outperforms arbitrary word dependency modeling in machine translation and sentence embedding, and has been effectively extended to capture node similarities in graph embedding. In this paper, we expand the newest approach to describe the relationships between distinct feature fields. The structure of the interaction layer is shown in Fig. 3.

To be more specific, we use the key-value attention mechanism to decide which feature combinations are relevant. Using feature m as an example, we will show how to find many significant high-order features using feature m. We begin by defining the association between feature m and feature k under a certain attention head h as follows:

$$\alpha_{m,k}^{(h)} = \frac{\exp(\varphi^{(h)}(e_m, e_k))}{\sum_{l=1}^{M} \exp(\varphi^{(h)}(e_m, e_l))}, \tag{7}$$

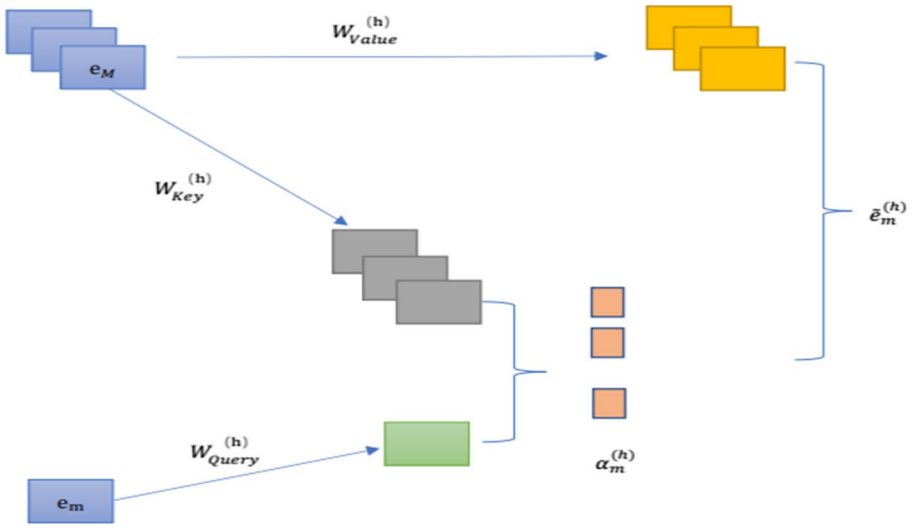

Fig. 3 The architecture of interacting layer. Combinatorial features are conditioned on attention weights, i.e., $\alpha_m^{(h)}$

$$\varphi^{(h)}(e_m, e_k) =< W_{Query}^{(h)} e_m, W_{Key}^{(h)} e_k >, \tag{8}$$

where (h)(,)is an attention function that determines the similarity between the features m and k. It can be defined as a neural network or as a simple inner product, i.e.,. We used inner product in this task since it is simple and effective. W(h)Query,W(h)KeyRd'd in Eq. (5) are transformation matrices that convert the original embedding space Rd into a new space Rd′. Following that, we update the representation of feature m in subspace h by merging all relevant features led by coefficients(h)mk:

$$\widetilde{e}_m^{(h)} = \sum_{k=1}^{M} \alpha_{m,k}^{(h)} \left(W_{Value}^{(h)} e_k \right), \tag{9}$$

where $\boldsymbol{W}_{\boldsymbol{Value}}^{(\boldsymbol{h})} \in \mathrm{Rd}' \times \mathrm{d}$, $\mathrm{e}^{(\mathrm{h})}{}_{\mathrm{m}} \in \mathrm{Rd}'$ denotes a new combinatorial feature acquired by our technique since it is a combination of feature m and its relevant features (under head h). In addition, a feature is likely to be implicated in many combinatorial features, which we do by employing multiple heads that form various subspaces and learn diverse feature interactions individually. In all subspaces, we gather the following combinatorial features:

$$\widetilde{e}_m = \widetilde{e}_m^{(1)} \oplus \widetilde{e}_m^{(2)} \oplus \cdots \oplus \widetilde{e}_m^{(H)}, \tag{10}$$

where H is the number of total heads and is the concatenation operator. We use typical residual connections in our network to maintain previously learnt combinatorial characteristics, such as raw individual (i.e., first-order) features. Formally

$$e_m^{Res} = \mathrm{ReLU}\left(\widetilde{e}_m + W_{Res} e_m\right), \tag{11}$$

where ReLU(z) = max(0, z) is a non-linear activation function, and WResRd'Hd is the projection matrix in case of dimension mismatching [38]. The representation of each

feature em will be modified into a new feature representation eResm, which is a representation of high-order features, as a result of such an interaction layer. Multiple similar layers can be stacked, with the output of the previous interacting layer feeding into the next interacting layer. We can represent arbitrary-order combinatorial features as a result of this.

Output layer

The interaction layer produces a collection of feature vectors eConmMm = 1, which contain raw individual features reserved by the residual block as well as combinatorial features gained by the multi-head self-attention process. We just concatenate them all and then use a non-linear projection as follows for the final CTR prediction:

$$\widehat{y} = \sigma\left(w^T\left(e_1^{Con} \oplus e_2^{Con} \oplus \cdots \oplus e_M^{Con}\right) + \mathrm{b}\right), \tag{12}$$

where wRd'HM is a column projection vector that linearly combines concatenated features, b is the bias, and (x) = 1/(1 + ex) converts the values to users clicking probabilities.

Training

Our loss function is *Log loss*, which is defined as follows:

$$\mathrm{Logloss} = -\frac{1}{N}\sum_{j=1}^{N}\left(y_j \log\left(\widehat{y}_j\right) + \left(1 - y_j\right)\log\left(1 - \widehat{y}_j\right)\right), \tag{13}$$

where *yj* and $\widehat{y}j$ are ground truth of user clicks and estimated CTR respectively, *j* indexes the training samples, and *N* is the total number of training samples. The parameters to learn in our model are $\{\mathrm{V_i}, \mathrm{Vm}, \mathrm{W}^{(\mathrm{h})}_{\mathrm{Query}}, \mathrm{W}^{(\mathrm{h})}_{\mathrm{Key}}, \mathrm{W}^{(\mathrm{h})}_{\mathrm{Value}}, \mathrm{W}^{\mathrm{Res}}, \mathrm{w}, \mathrm{b}\}$, which are updated via minimizing the total *Logloss* using gradient descent.

Experiment

Experimental setup

Experimental data set

Data Sets. Four publicly available real-world data sets are used in this study. Table 1 summarizes the statistics for the data sets. Criteo3 This is a CTR prediction benchmark dataset with 45 million click records on shown adverts. It has 26 numerical and 13 category feature fields. Avazu4 This dataset provides information on users' mobile activities, such as whether or not they click on a presented mobile ad. It comprises 23 feature fields that range from user/device characteristics to ad properties. MovieLens-1M6 Users' movie ratings are collected in this collection. We consider samples with a rating of less

Table 1 Statistics of evaluation data sets

Data	Samples	Fields	Features (sparse)
Criteo	45,840,617	39	998,960
Avazu	40,428,967	23	1,544,488
Movielens-1M	739,012	7	3529

than 3 to be negative samples during binarization since a low score suggests that the user dislikes the film. Positive samples (those with a rating more than 3) are kept, whereas neutral samples (those with a rating of 3 or less) are discarded.

Evaluation metrics

To assess the effectiveness of all strategies, we employ two widely used criteria.

Area of the University of Chicago The likelihood that a CTR predictor would award a higher score to a randomly chosen positive item than a randomly chosen negative item is measured by the area under the ROC Curve (AUC). AUC is a measure of how well something works. The greater the AUC, the better.

We adopt Logloss as a clear measure since all models try to reduce the Logloss described by Eq. (10).

It's worth noting that for the CTR prediction job, a slightly higher AUC or lower Logloss at the 0.001-level is considered significant, as has been previously mentioned.

Comparison model

FM models second-order feature interactions using factorization techniques.

AFM. AFM is one of the most advanced models for capturing the interplay of second-order features. It extends FM by using the attention mechanism to discern between the relative relevance of second-order combination characteristics.

NFM. On the second-order feature interaction layer, NFM superimposes a deep neural network. The interplay of high-order features is implicitly captured by the nonlinearity of neural networks.

deepFM.deepFM utilizes the deep layer's deep learning to gain high-level crossover features, FM collects low-level crossovers, and both high and low-level crossover features are acquired at the same time.

Widedeep. The memory features of the broad layer learning model are used in the Widedeep model, while the deep layer learns the model's generalization characteristics.

Deepcrossing. The Deepcrossing model incorporates a residual network based on deepfm, which enhances the model's interpretability.

DCN. DCN can successfully capture a narrow range of effective feature interactions, learn highly nonlinear effects, and has a cheap computing cost. It does not involve human feature engineering traversal or search.

PNN. The PNN model obtains high-level and low-level cross features using the inner product and outer product to arrive at the final recommendation result.

Autoint. To produce weighted cross features, Autoint employs a multi-head self-attention technique.

Comparative experiment

In accordance with the Table 2 experimental findings. The following conclusions may be derived from the results of the experiment: (1) Attention mechanisms are investigated using FM and AFM models. The AFM model has a greater experimental impact than the FM model on all data sets, indicating that the attention mechanism is involved in the recommendation model. (2) As shown in the table above, several models that capture high-level cross-feature interactions have advantages and disadvantages. When

Table 2 Effectiveness comparison of different algorithms

Model	Criteo		Movielens-1M		Avazu	
	AUC	LOSS	AUC	LOSS	AUC	LOSS
FM	0.6869	0.5286	0.5347	0.4462	0.5437	0.6221
Weidedeep	0.7066	0.4827	0.8328	0.3334	0.7424	0.4117
Deepfm	0.7283	0.4707	0.8340	0.3346	0.7461	0.4041
AFM	0.7220	0.4754	0.8295	0.3358	0.7567	0.4012
DCN	0.7094	0.4920	0.8249	0.3393	0.7349	0.4139
NFM	0.7027	0.5645	0.8297	0.3357	0.7400	0.4179
PNN	0.7084	0.4870	0.8312	0.3353	0.7374	0.4096
Autoint	0.7060	0.6049	0.8362	0.3393	0.7450	0.4496
Deepcrosing	0.7375	0.4732	0.8373	0.3305	0.7572	0.3989
IARM	0.7545	0.4830	0.8386	0.3296	0.7652	0.3994

Table 3 IARM's performance was studied using ablation tests

Data sets	Model	AUC	LOSS
Criteo	IARM	0.7545	0.4830
	IARM*	0.7371	0.4965
Avazu	IARM	0.7652	0.3994
	IARM*	0.7591	0.3995
Movielens	IARM	0.8386	0.3296
	IARM*	0.8349	0.3344

IARM is a full model, whereas IARM* is a model that eliminates user interest

the deepfm model with high-order cross features is compared to the fm model without high-order cross features, the suggested model's accuracy improves. (3) On three separate data sets, the suggested IARM model has the greatest AUC and the lowest LOSS when compared to other models. It demonstrates that the IARM model provides more accurate and effective recommendations.

Ablation experiment of the model

This paper conducts an ablation investigation and compares multiple IARM variations in order to further validate and comprehend the paradigm described in this article.

The influence of personal interest on the model

The user interest module is integrated into the basic IARM paradigm, allowing it to learn about each user's individual interests. This study isolates the interest module from the IARM model and keeps the status quo of other structures to establish an IARM* model in order to test the interest module's efficacy. The performance of all data sets will suffer if the interest module is removed, as demonstrated in the Table 3. In particular, on the criteo, avazu, and movielens data sets, the IARM model outperforms the change model IARM*. This demonstrates that the interest module of the IARM model developed in this research contributes significantly to the accuracy of the recommendation outcomes.

The influence of network layer parameters on the model

By superimposing numerous interacting layers on top of each other, the IARM model suggested in this study learns high-order feature combinations. As a result, the focus of this research is on how the model's performance varies with the number of interaction layers, specifically if the model's number of interaction layers influences the combination characteristics. It refers to the acquisition of high-level characteristics of non-progressive input from raw data if there is no interaction layer mentioned in this article. As illustrated in the diagram above, the findings are summarized. The performance of the movielens data set is greatly enhanced when an interaction layer is utilized, i.e., feature interaction is taken into account, demonstrating that the combined features give extremely relevant information for prediction. The model's performance improves further if the number of interaction layers is increased, taking into account high-order combination characteristics. When the number of layers approaches three, performance stabilizes, demonstrating that adding extremely high-order features does not give predictive information (Fig. 4).

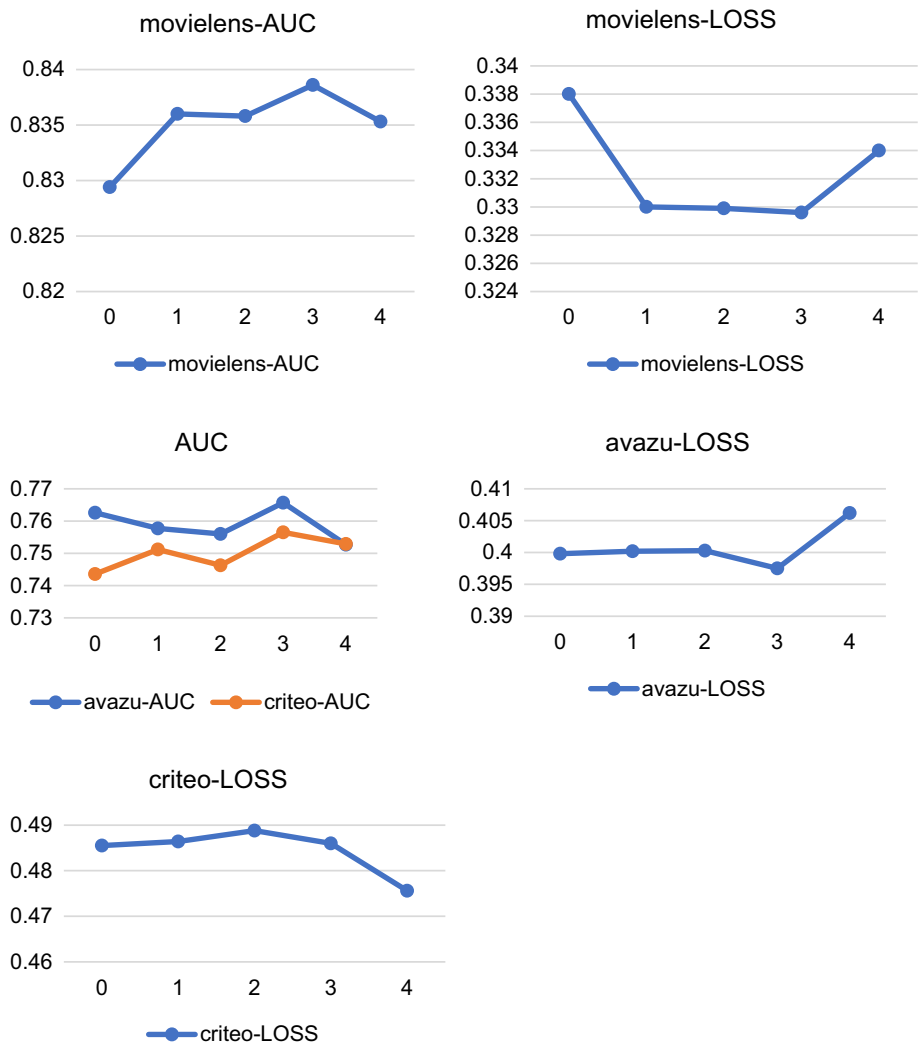

Fig. 4 Demonstrates the performance of IARM with various data types and network layers

The influence of self-attention mechanism

The IARM model has a multi-head self-attention mechanism that allows it to assign various weights to different variables, improve the influence of relevant features on recommendation outcomes, and improve recommendation accuracy. This research provides an IRM model that eliminates the multi-head self-attention mechanism while leaving other structures unaltered in order to test the usefulness of the module. The overall influence of the model on the criteo, avazu, and movielens data sets has diminished when the multi-head attention module is eliminated, as can be seen in the Table 4. This demonstrates that the IARM model's multi-head self-attention mechanism had a role.

Influence of residual network

The residual network, which can learn all the combined features, is used in the typical IARM model in this article, enabling for the modeling of extremely high-order combinations. This research removes the residual network from the standard model IARM in order to demonstrate its contribution to the model, while maintaining the status quo of other structures. The performance of all data sets will suffer if the residual network is removed, as seen in the Table 5. On the criteo, avazu, and MovieLens datasets, the entire model IARM performs much better than the version IARM-, demonstrating that residual connection is required for modeling high-order feature interactions in our proposed technique.

Table 4 IARM performance is being investigated using ablation studies

Data sets	Model	AUC	LOSS
Criteo	IARM	0.7545	0.4830
	IRM	0.7497	0.4902
Avazu	IARM	0.7652	0.3994
	IRM	0.7637	0.4019
Movielens	IARM	0.8386	0.3296
	IRM	0.8345	0.3292

IRM is a model without the multi-head attention mechanism, while IARM is a full model

Table 5 IARM's performance was examined using ablation tests

Data sets	Model	AUC	LOSS
Criteo	IARM	0.7545	0.4830
	IARM-	0.7511	0.4822
Avazu	IARM	0.7652	0.3994
	IARM-	0.7621	0.4086
Movielens	IARM	0.8386	0.3296
	IARM-	0.8236	0.3422

IARM is a full model, whereas IARM- is a residual network-free model

Table 6 Displays the IARM model's performance on the Criteo dataset under various test sets

Model	Criteo					
Proportion	0.2		0.3		AVG. changes	
	AUC	LOSS	AUC	LOSS	AUC	LOSS
FM	0.6869	0.5286	0.6785	0.5368	− 0.0084	+ 0.0082
Weidedeep	0.7066	0.4827	0.7007	0.4846	− 0.0059	+ 0.0019
Deepfm	0.7283	0.4707	0.7249	0.4792	− 0.0034	+ 0.0085
AFM	0.7220	0.4754	0.7084	0.4877	− 0.0136	+ 0.0123
DCN	0.7094	0.4920	0.7042	0.5010	− 0.0052	+ 0.009
NFM	0.7027	0.5645	0.7005	0.5654	− 0.0022	+ 0.0009
PNN	0.7084	0.4870	0.7013	0.4979	− 0.0071	+ 0.0109
Autoint	0.7060	0.6049	0.6921	0.6552	− 0.0139	+ 0.0503
Deepcrosing	0.7375	0.4732	0.7356	0.4792	−0.0019	+ 0.006
IARM	0.7545	0.4830	0.7527	0.4993	− 0.0018	+ 0.0163

Visual explanation

A good recommendation model can help improve not just the quality of recommendations, but also their interpretability. We'll use movielens as an example in this section to explain how the IARM model provides a good cross-feature combination.

The association between several feature fields in the data is also examined in this article. Based on the average attention scores of all characteristics in the data, this study calculates the correlation between them. The graph above summarizes the relationship between the various characteristics. It can be observed that the characteristics sex, age, and sex, age (that is, light-colored patches) have a high link, and this combination of features will play a significant role in the recommendation outcomes (Fig. 5).

Model generalization

The term "model generalization" describes whether or not a model is similarly accurate when applied to fresh data. The criteo data set is used in this section as an example of how to partition a data set into a training set and a test set with a ratio of 0.2 and 0.3, respectively. The data set divided by the 0.3 column may be divided into multiple test sets, yielding more data for the model. As seen in the Table 6, AUC and LOSS alter according on the model's division ratios. Overall, the IARM model has the highest AUC value while simultaneously having the lowest LOSS value. Furthermore, the model's AUC varies relatively little as the test set grows, having the maximum AUC value at the end. This demonstrates that the IARM model still outperforms other models on the new data set and has a significant generalization ability.

Conclusion

This research provides a recommendation model that incorporates both user interest and a multi-head attention mechanism. This model can learn the user's preferences and how high-level features interact automatically. The multi-head self-attention mechanism's newly added user interest layer and interaction layer, which allows each feature to interact with other features and assess feature significance through learning, are the key

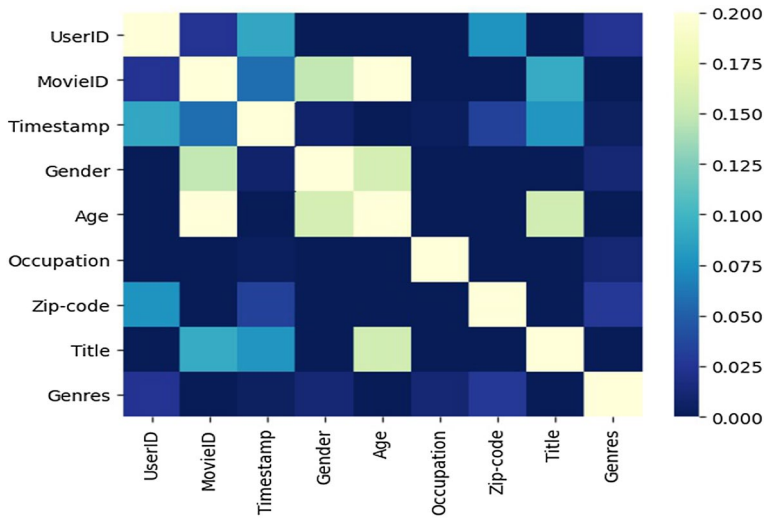

Fig. 5 Depicts the connection between the features

to the technique in this study. The model's structure, interpretability, and generalization are all discussed and analyzed in this article. The results of the experiments on three real data sets show that the model described in this research is more effective and accurate in its recommendations. In order to increase the recommendation model's accuracy, we'd like to incorporate contextual information into our process in the future.

Acknowledgements
The authors would like to thank all the anonymous reviewers for their insightful comments. This work was financially supported by the Bing-tuan Science and Technology Public Relations Project, a data-driven regional smart education service key technology research and application demonstration (2021AB023). The authors would like to thank colleagues and the anonymous reviewers who have provided valuable feedback to help improve the paper.

Author contributions
WZ, YH wrote the main manuscript text and BY, ZZ prepared Tables 1, 2 and 3. All authors reviewed the manuscript. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 5 January 2022 Accepted: 21 January 2023
Published online: 31 January 2023

References

1. Deep Session Interest Network for Click-Through Rate Prediction. arXiv:1905.06482v1 [cs.IR] 16 May 2019.
2. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. arXiv:1810.11921v2 [cs.IR] 23 Aug 2019.

3. Wang R, Shivanna R, Cheng D Z, et al. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems. 2020.
4. Yi T, Dehghani M, Bahri D, et al. Efficient Transformers: A Survey. 2020.
5. Shen X, Yi B, Liu H, et al. Deep variational matrix factorization with knowledge embedding for recommendation system. IEEE Trans Knowl Data Eng. 2019;99:1–1.
6. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. 2017.
7. Feature Generation by Convolutional Neural Network for Click-Through Rate Prediction. arXiv:1904.04447v1 [cs.IR] 9 Apr 2019.
8. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167; 2015.
9. Deep & Cross Network for Ad Click Predictions. arXiv:1708.05123 [cs.LG] 17 Aug 2017.
10. Operation-aware Neural Networks for User Response Prediction. arXiv:1904.12579v1 [cs.IR] 2 Apr 2019.
11. Jianfang W, Xilin W, Xu Y, Qiuling Z. Deviation-based graph attention neural network recommendation algorithm. Control Decis 2021; 1–9.
12. Wide & Deep Learning for Recommender Systems. arXiv:1606.07792v1 [cs.LG] 24 Jun 2016.
13. LLC, Lee DL, Liu Z, et al. Multi-interest network with dynamic routing for recommendation at Tmall. ACM. 2019.
14. Jianing Z, Jingsheng L, Xuexue Z. Graph network social recommendation algorithm based on agru-gnn. Comput Syst Appl. 2021;30(05):219–27.
15. Zou H, Zheng M. Random node recommend algorithm for influence maximization in social network[C]//2018 9th International Conference on Information Technology in Medicine and Education (ITME). IEEE Computer Society, 2018.
16. Shan Y, Ryan Hoens T, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 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. ACM, 255–262.
17. Pande H. Field-embedded factorization machines for click-through rate prediction. 2020.
18. Qu Y, Han C, Kan R, et al. Product-based neural networks for user response prediction[C]//2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, 2016.
19. xDeepFM: Combining explicit and implicit feature interactions for recommender systems. arXiv:1803.05170v3 [cs.LG] 30 May 2018.
20. Xiao J, Ye H, He X, et al. Attentional factorization machines: learning the weight of feature interactions via attention networks. 2017.
21. Kuifeng Yu, Guihua D, Xiang S. College entrance examination volunteer recommendation algorithm based on multi-feature weight fuzzy clustering. J Central South Univ (Nat Sci Edn). 2020;51(12):3418–29.
22. Rui WH, Sui L, Jian HL. News recommendation algorithm based on the combination of content recommendation and time function. Comput Digital Eng. 2020;48(12):2973–7.
23. Zhang W, Du T, Wang J. Deep learning over multi-field categorical data: a case study on user response prediction. 2016.
24. Pan J, Xu J, Ruiz A L, et al. Field-weighted factorization machines for click-through rate prediction in display advertising. 2018.
25. Ye Q, Xiongkai S, Rong G, Chunzhi W, Jing L. Social recommendation algorithm based on attention gated neural network. Comput Eng Appl. 2021; 1–9.
26. Yan G. Research on recommendation algorithm based on the convolutional neural network. Nanjing University of Posts and telecommunications, 2020.
27. Gai K, Zhu X, Li H, et al. Learning piece-wise linear models from large scale data for ad click prediction. 2017.
28. Shengyun Z, Hengzhong J. Factorization machine. J Digital Content Soc Korea. 2017; 18.
29. Guo H, Tang R, Ye Y, Li X, He X. Deepfm: a factorization-machine-based neural network for ctr prediction. arXiv preprint arXiv:1703.04247; 2017.
30. Zhou G, Song C, Zhu X, et al. Deep interest network for click-through rate prediction. 2017.
31. Slim A, Hush D, Ojha T, et al. An automated framework to recommend a suitable academic program, course and instructor[C]//2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService). IEEE, 2019.
32. Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations//Acm Conference on Recommender Systems. ACM, 2016;191–198.
33. Zhang W, Du T, Wang J. Deep learning over multi-field categorical data. In: European conference on information retrieval. Springer, 2016; p. 45–57.
34. Zhu J, Shan Y, Mao JC, Yu D, Rahmanian H, Zhang Y. Deep embedding forest: Forest-based serving with deep embedding features. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2017; p. 1703–1711.
35. Wang-Cheng K, McAuley J. Self-attentive sequential recommendation. In 2018 IEEE International Conference on Data Mining (ICDM), p. 197–206. IEEE, 2018.
36. Zhou G, Zhu X, Song C, Fan Y, Zhu H, Ma X, Yan Y, Jin J, Li H, Gai K. deep interest network for clickthrough rate prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2018; p. 1059–1068.
37. Xiao J, Ye H, He X, Zhang H, Wu F, Chua TS. Attentional factorization machines: learning the weight of feature interactions via attention networks. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. AAAI Press, 2017; p. 3119–3125.
38. Zhang QL, Rao L, Yang Y. DGFFM: generalized field-aware factorization machine based on DenseNet[C]//2019 International Joint Conference on Neural Networks (IJCNN). 2019.

Publisher's Note