applied sciences

MDPI

Article

An Attention-Based Latent Information Extraction Network (ALIEN) for High-Order Feature Interactions

Ruo Huang [1], Shelby McIntyre [2], Meina Song [1,*], Haihong E [1,*] and Zhonghong Ou [1]

[1] School of Computer Science, Beijing University of Posts & Telecommunications, Beijing 100876, China; charleshuangruo@bupt.edu.cn (R.H.); zhonghong.ou@bupt.edu.cn (Z.Ou.)
[2] Leavey School of Business, Santa Clara University, Santa Clara, CA 95053, USA; smcintyre@scu.edu
* Correspondence: mnsong@bupt.edu.cn (M.S.); ehaihong@bupt.edu.cn (H.E.)

Received: 16 July 2020; Accepted: 3 August 2020; Published: 7 August 2020

Abstract: One of the primary tasks for commercial recommender systems is to predict the probabilities of users clicking items, e.g., advertisements, music and products. This is because such predictions have a decisive impact on profitability. The classic recommendation algorithm, collaborative filtering (CF), still plays a vital role in many industrial recommender systems. However, although straight CF is good at capturing similar users' preferences for items based on their past interactions, it lacks regarding (1) modeling the influences of users' sequential patterns from their individual history interaction sequences and (2) the relevance of users' and items' attributes. In this work, we developed an attention-based latent information extraction network (ALIEN) for click-through rate prediction, to integrate (1) implicit user similarity in terms of click patterns (analogous to CF), and (2) modeling the low and high-order feature interactions and (3) historical sequence information. The new model is based on the deep learning, which goes beyond the capabilities of econometric approaches, such as matrix factorization (MF) and k-means. In addition, the approach provides explainability to the recommendation by interpreting the contributions of different features and historical interactions. We have conducted experiments on real-world datasets that demonstrate considerable improvements over strong baselines.

Keywords: recommender systems; collaborative filtering; click-through rate prediction; high-order feature interactions; attention mechanism; explainable recommendation

1. Introduction

In recent years, recommender systems have improved substantially and are now widely adopted by many online services in domains such as news, e-commerce and social media, among many others. The key to a personalized recommendation for a target user is in modeling similar users' preferences for items in a domain based on those similar users' past interactions and the similarity of their patterns to those of the target user (e.g., in ratings and clicks). In a broader sense, any use of such user similarity (sometimes called "nearest neighbors") is an umbrella concept known as collaborative filtering [1,2].

With the inclusion of time-stamp and sequence, collaborative filtering is being merged with history sequences to play a vital role in many industrial recommender systems. The most well-known collaborative filtering technique, matrix factorization [3–5], projects users and items into a shared latent space and utilizes a vector of latent dimensions to represent a user or an item. Thereafter a user's interaction with an item is modeled as the inner product of their latent vectors. Recently, researchers have been embracing deep-learning neural architectures that can learn very complicated functions from data, to replace the inner product applied in matrix factorization [6,7] and also include information from history sequences [8,9].

However, even with (a) user-user similarity (based on, e.g., clicks), and (b) history sequences, there is still another source of information which emerges from (c) the "attributes" (sometime called "features") of users and items, and the interactions of these attributes, originally at the main effect level, but that is now moving to the second, third and even higher-order interactions. This is the landscape on which our model operates (attention-based latent information extraction network (ALIEN) for high-order feature interactions).

The user's attributes include demographic factors, e.g., age, gender, occupation and educational background [7,10–14]. In addition, item attributes such as the category of a product, the genre of a movie and the release date of an album, not only render the basic information about the item, but also provide clues as to why the user is interested in it [10,15]. For example, it is reasonable to recommend *Toy Story*, a famous cartoon movie, to an eight-year-old boy Peter when he enters a video streaming website. Therefore, a third-order feature interaction, <gender = male, age = 8, movie's genre = (animation, children's, comedy)>, can be an informative description of this scenario for prediction. Existing works which focused on modeling low-order feature interactions from user and item attributes have been proposed for click-through rate (CTR) prediction [16–19], whose primary task is to predict the probabilities of users clicking items, e.g., advertisements, music and products. However, although the problem of feature engineering has been automated beyond manual feature selection, these models still lack the capability of extracting latent information from high-order feature interactions, which usually increases the dimensions and sparsity of the input features exponentially, leading to a more serious problem of model overfitting [20].

In addition, Peter may be curious about the reason why *Toy Story* was recommended to him. A possible assumption is, on the one hand, that it was recommended because he watched *Lion King* last week, whose genres are (animation, children's, musical), similar to *Toy Story*'s, and welcomed by children as well. On the other hand, the third-order feature interaction <gender = male, age = 8, movie's genre = (animation, children's, musical)> has a greater impact on the recommendation of *Toy Story* than any other feature interactions, e.g., <zipcode = 48067 and movie's year of release = 1994>. To meet his expectations, it is appropriate to design a recommender system which not only provides a precise recommendation, but also is capable of finding out the exact feature interactions and history items which have greater influences on this recommendation. Thus, we propose a novel model, the attention-based latent information extraction network (ALIEN), to solve the CTR problem mentioned above. It has been demonstrated that the self-attention mechanism [15,21,22] is able to investigate the internal relationships between words within a sentence in the natural language processing task. Similarly, it enables ALIEN to provide explanations of recommendations by capturing comprehensive relationships between feature interactions from user and item attributes. In order to resolve the issues of modeling high-order feature interactions and providing explainability at the same time in a unified way, we built two attention-based layers from both micro and macro perspectives. The main contributions of this paper include:

- An attention-based latent information extraction network (ALIEN) is proposed which takes user and item attributes and the user's history interactions as features; two attention-based layers are applied in both macro and micro perspectives: (1) the macro one learns the latent information by modeling the low and high-order of feature interactions, and interprets the contribution of each feature interaction; (2) the micro layer investigates the different impact which each history interaction has on the candidate item.
- Dice [8] is introduced as the activation function, to standardize the input data and place the mean at the inflection point of the sigmoid.
- We conducted empirical evaluations and validated the effectiveness of the model on two real-world datasets.
- We demonstrate the effectiveness of the approach in providing the explainability of the model.

2. Related Work

CTR prediction models have been successfully developed in both academia and industry [8,16,23–27]. Until fairly recently, feature selection from attributes was mainly hand-crafted by experts [28]. However, it was a tedious task, and experts' experience and expertise were highly required [28]. Therefore, there were works proposed [16–19] to model feature interactions automatically. Among these works, factorization machines (FM) [19] is a representative model, which was built to capture the first and second-order feature interactions in a linear way, and its effectiveness has been demonstrated in many recommendation tasks [29,30]. However, these models only focused on modeling low-order feature interactions, and lacked the capability of extracting latent information from high-order feature interactions, which tends to be combinatorially explosive.

To solve the problem of modeling high-order feature interactions, approaches were made which utilized feed-forward neural networks. He et al. [31] proposed neural factorization machines (NFM) to seamlessly combine the linearity of FM and the non-linearity of neural networks in modeling second-order and higher-order feature interactions, respectively. Qu et al. [32] proposed a product-based neural network (PNN), wherein a product layer is used to explore high-order feature interactions after an embedding layer. Lian et al. [33] proposed a novel compressed interaction network (CIN), which aimed to generate feature interactions at the vector-wise level. In addition to models for academia, Internet companies proposed several representative deep models which aimed to learn non-linear feature interactions from large-scale data. Cheng et al. [23] from Google proposed "Wide & Deep" for app recommendation, wherein the multi-layer perceptron (MLP) was used on the concatenation of feature embedding vectors, to learn feature interactions. Shan et al. [34] from Microsoft proposed DeepCross which utilized a deep residual MLP [35] to learn feature interactions. However, these methods were not capable of interpreting the contribution of each feature interaction.

To deal with the problem of explainability, Xiao et al. [36] applied the attention mechanism [15,22] to learn the importance of each feature interaction. Song et al. [20] moved a step further by combining a multi-head self-attentive neural network with residual connections, to model the importance of feature interactions with different orders. To solve these two issues at the same time in a unified way, we make the following contributions based on the improvements over the existing techniques:

(1) For the macro domain, we developed a novel self-attention-based layer named the attributes-driven latent information extraction layer (which is denoted USER $\times$ CANDIDATE), to learn the latent information from the user and item attributes by modeling the low and high-order of feature interactions, and interpret the contribution of each feature interaction. Our approach evolved from the existing techniques of the synthesizer [37] and the compressed interaction layer (CIN) [33]. Contrary to the role the synthesizer played in the natural language processing (NLP) task, for the first time to our best knowledge, it was transformed by us into the approach which is able to model feature interactions and interpret their contributions for the recommendation task.

(2) In addition, by adding the capability of modeling from a user's history interactions from the micro perspective, we solve the issue of CIN only modeling static feature interactions without having the ability to capture the user's diverse interests. Therefore, another novel attention-based layer named user behavior-driven latent information extraction layer (HISTORY) was developed to learn the latent information from user's historical behavior by modeling the items in the user's history interactions, and investigating the different impact which each history interaction has on the candidate item. Our approach evolved from the existing technique of the deep interest network (DIN) [8]. Unlike the items' embedding method utilized in DIN, we propose a different approach described in Section 3.3, to better suit ALIEN's architecture. Moreover, since DIN lacks the modeling of feature interactions, the attributes-driven latent information extraction layer (USER $\times$ CANDIDATE) described above compensates for that disadvantage.

3. The ALIEN Architecture

In this section, we first present the main idea of the attention-based latent information extraction network (ALIEN) proposed in this paper, and formulate the description of the problem, as explained above. Afterwards, we elaborate with a diagram of the architecture of ALIEN.

3.1. Main Idea

Our notation is summarized in Table 1. In the ALIEN model, there are three layers:

(1) The attributes-driven latent information extraction layer (USER × CANDIDATE);
(2) The user behavior-driven latent information extraction layer (HISTORY);
(3) The latent information digestion and prediction layer (PREDICTION).

Table 1. Notation descriptions.

Notation	Description
$\mathcal{U}, \mathcal{V}$	the sets of users and items
$\boldsymbol{u} \in \mathbb{R}^{m \times k}, \boldsymbol{v} \in \mathbb{R}^{n \times k}$	user u's and v's k-dimensional embeddings
$\boldsymbol{S}_u^{v_c}$	the embedding set of u's history items related to u's candidate item v_c
$\widetilde{\boldsymbol{u}}^f, \widetilde{\boldsymbol{v}}^f$	u's and v's high-dimensional and sparse one-hot encodings which contain all the fields of attributes
$\boldsymbol{W}_{\widetilde{\boldsymbol{u}}^{(i)}} \in \mathbb{R}^{k \times \lvert\widetilde{\boldsymbol{u}}^{(i)}\rvert}, \boldsymbol{W}_{\widetilde{\boldsymbol{v}}^{(j)}} \in \mathbb{R}^{k \times \lvert\widetilde{\boldsymbol{v}}^{(j)}\rvert}$	the weights assigned to $\widetilde{\boldsymbol{u}}^{(i)}$ and $\widetilde{\boldsymbol{v}}^{(j)}$ for latent space projection
$\ell_{\mathcal{F}_{(i)}^{u,v_c}}, \ell_{\mathcal{F}^{u,v_c}}$	the numbers of feature interactions of the i-order and all the orders mentioned
$\mathcal{F}^{u,v_c} \in \mathbb{R}^{\ell_{\mathcal{F}^{u,v_c}} \times k}$	the complete representation vector of feature interactions of all the orders mentioned
$\widehat{\mathcal{F}}^{u,v_c} \in \mathbb{R}^{\ell_{\mathcal{F}_{(1)}^{u,v_c}} \times k}$	the output vector of the Synthesizer module
$\boldsymbol{X}_h^l$	the h-th feature vector of the l-th layer in the CIN module
$\boldsymbol{\alpha}_{syn}$	the feature interaction-level attention score matrix for $\mathcal{F}_{(1)}^{u,v_c}$
$\psi_\cdot(\cdot) \in \mathbb{R}^{\ell_{\mathcal{F}_{(1)}^{u,v_c}} \times \ell_{\mathcal{F}_{(1)}^{u,v_c}}}$	the Multi-Layer Perceptron units in the Synthesizer module
$\sigma(\cdot)$	the Dice activation function
$\boldsymbol{W}_{syn} \in \mathbb{R}^{\ell_{\mathcal{F}_{(1)}^{u,v_c}} \times \ell_{\mathcal{F}_{(1)}^{u,v_c}}}$	the randomly initialized matrix utilized in the Synthesizer module
$\mathcal{I}_u^{v_c}$	the intensity of u's interest in v_c
$\boldsymbol{w}_{ci}$	the correlation score between $\boldsymbol{v}_c$ and $\boldsymbol{v}_i$
$\boldsymbol{\mathcal{P}}_u^{v_c}$	u's preference vector for v_c

Our model is illustrated in Figure 1. ALIEN performs two tasks in the macro and micro perspectives, and then merges those in the PREDICTION layer:

Attributes-driven latent information extraction layer (USER × CANDIDATE): This layer performs a macro task of learning the latent information from modeling the low and high-order feature interactions, thereby establishing the contribution of each feature interaction to the user u's interest. Modeling low and high-order feature interactions from user and item attributes is overlooked by many recommendation models [7,10]. In addition, in their models, user and item embeddings $\boldsymbol{u}$ and $\boldsymbol{v}$ are initialized only with indices of u and v, which have vague meanings and converge slowly. However, user and item attributes are very important and provide key features which describe u's and v's basic characteristics. In our model, user and item attributes are used to initialize $\boldsymbol{u}$ and $\boldsymbol{v}$ via an embedding method, in order to render a clearer meaning with the side information and make the model converge faster. The embedding method is discussed in Section 3.3. Moreover, taking into consideration the exhaustive volume of calculations for generating high-order feature interactions, which grows exponentially when the order of interactions increases, we propose a novel architecture to reduce the computational complexity and parameter costs.

User behavior-driven latent information extraction layer (HISTORY): This layer finishes the micro task and investigates the different impact which each history interaction has on the candidate item. Obviously it is not enough to build user profile with only stationary attributes. User behavior not only

implies u's historical interests, but also provides clues as to why u will choose to click the candidate item v_c (or not). Given v_c, a user's interest related to v_c can be locally activated, and therefore its intensity is available for measurement, which is regarded as one important factor to improve the CTR prediction in our work.

Latent information digestion and prediction layer (PREDICTION): This layer concatenates the outputs of the first two layers and feeds them into a fully connected layer with the Dice activation function to generate the CTR prediction of v_c for u. Dice standardizes the input data and puts the mean at the inflection point of the sigmoid, which is important when the inputs of each layer follow different distributions.

The detailed work flow of these three layers is discussed in Section 3.4. The main idea of our approach is to:

(1) Build a basic user profile by constructing the low and high-order feature interactions from user and item attributes, and assign them different weights, to automatically learn the different influences on the candidate item.
(2) Enrich the user's profile comprehensively by locally activating u's interest related to v_c with u's corresponding history items $S_u^{v_c}$, and measure the intensity of u's locally activated interest, to enhance the performance of CTR prediction.

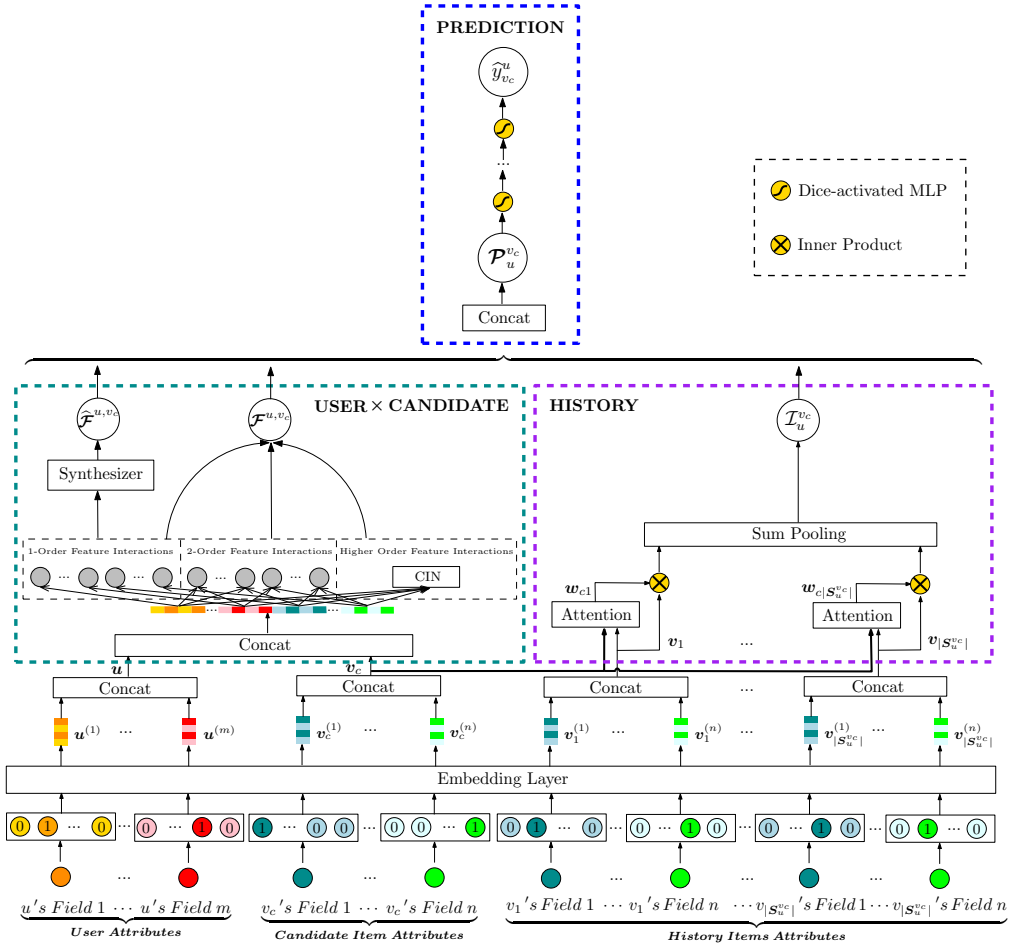

Figure 1. ALIEN architecture. The dark-green dashed box on the left side illustrates the attributes-driven latent information extraction layer (USER × CANDIDATE); the purple dashed box on the right, the user behavior-driven latent information extraction layer (HISTORY); and the blue dashed box at the top, the latent information digestion and prediction layer (PREDICTION).

3.2. Problem Formulation

We formulate the problem of CTR prediction. Let $\mathcal{U}$ denote a set of users and $\mathcal{V}$ denote a set of items, where $|\mathcal{U}|$ and $|\mathcal{V}|$ are the total numbers of users and items, respectively. The user-item interaction matrix $Y \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{V}|}$ is defined according to users' implicit feedback, where binary $y_{uv} = 1$ indicates that there is an implicit interaction between user u and item v, e.g., behaviors of clicking, watching and browsing. Otherwise, there is no interaction between u and v when $y_{uv} = 0$. Take the example of user $u \in \mathcal{U}$ and item $v \in \mathcal{V}$. There are m and n fields of features in u's and v's attributes, respectively. u's and v's k-dimensional embeddings $\boldsymbol{u} = [\boldsymbol{u}^{(1)}\ \boldsymbol{u}^{(2)}\ ...\ \boldsymbol{u}^{(m)}] \in \mathbb{R}^{m\times k}$ and $\boldsymbol{v} = [\boldsymbol{v}^{(1)}\ \boldsymbol{v}^{(2)}\ ...\ \boldsymbol{v}^{(n)}] \in \mathbb{R}^{n\times k}$ denote the concatenations of u's and v's feature embeddings, where k-dimensional $\boldsymbol{u}^{(i)}$ and $\boldsymbol{v}^{(j)}$ are encoded from the i-th and j-th fields in u's and v's attributes, respectively. The procedure of embedding will be discussed in Section 3.3. The problem of click-through rate prediction is to predict the probability of u clicking the candidate item v_c. Our goal is to make precise the CTR prediction by modeling u and v_c's low and high-order feature interactions and u's history interactions. u's history interactions consist of u's history items before he or she interacts with v_c, and the embedding set of u's history items is denoted as $\boldsymbol{S}_u^{v_c} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_{|S_u^{v_c}|}\}$.

3.3. Embedding Procedure

Differently from the embedding procedure used in [8], $\boldsymbol{u}$ and $\boldsymbol{v}$ are built with their attributes, respectively. We follow the feature embedding method applied in [32]. Take as an example the procedure of constructing item embeddings only, to conserve space. It consists of three steps. Firstly, all the numerical and categorical fields of item attributes are transformed into high-dimensional and sparse one-hot encodings, i.e., $\widetilde{\boldsymbol{v}}^f = [\widetilde{\boldsymbol{v}}^{(1)}\ \widetilde{\boldsymbol{v}}^{(2)}\ ...\ \widetilde{\boldsymbol{v}}^{(n)}]$. Since each field $\widetilde{\boldsymbol{v}}^{(i)}$ is a different type of attribute, apparently its one-hot encoding has a different dimension, denoted as $|\widetilde{\boldsymbol{v}}^{(i)}|$. To reduce and unify dimensions, secondly, each encoding $\widetilde{\boldsymbol{v}}^{(i)}$ is assigned a weight $\boldsymbol{W}_{\widetilde{\boldsymbol{v}}^{(i)}} \in \mathbb{R}^{k\times|\widetilde{\boldsymbol{v}}^{(i)}|}$ and projected into a low-dimensional latent vector space with $\boldsymbol{W}_{\widetilde{\boldsymbol{v}}^{(i)}}$:

$$\boldsymbol{v}^{(i)} = \boldsymbol{W}_{\widetilde{\boldsymbol{v}}^{(i)}}\widetilde{\boldsymbol{v}}^{(i)}, \tag{1}$$

where the k-dimensional dense vector $\boldsymbol{v}^{(i)} \in \mathbb{R}^k$ is the representation of $\widetilde{\boldsymbol{v}}^{(i)}$. Finally, the item embedding $\boldsymbol{v} = [\boldsymbol{v}^{(1)}\ \boldsymbol{v}^{(2)}\ ...\ \boldsymbol{v}^{(n)}] \in \mathbb{R}^{n\times k}$ is generated by concatenating all fields of vectors. The generation of $\boldsymbol{u} = [\boldsymbol{u}^{(1)}\ \boldsymbol{u}^{(2)}\ ...\ \boldsymbol{u}^{(m)}] \in \mathbb{R}^{m\times k}$ is similar to that of $\boldsymbol{v}$.

3.4. Model Description

In this section, we discuss the ALIEN model's entire recommendation process and the main system components with the adopted technologies we introduced.

As illustrated in Figure 1, ALIEN integrates the modeling of attributes-driven and user behavior-driven latent information into one single architecture. The user behavior-driven latent information extraction layer (HISTORY) evolved from the deep interest network (DIN) model used in [8], and the model proposed in it is regarded as a baseline to compare with ALIEN model. The latent information digestion and prediction layer (PREDICTION) receives the outputs of the attributes-driven latent information extraction layer (USER $\times$ CANDIDATE) and the user behavior-driven latent information extraction layer (HISTORY), and generates the final prediction.

3.4.1. Attributes-Driven Latent Information Extraction Layer (USER $\times$ CANDIDATE)

In terms of building a comprehensive user profile, making good use of user and item attributes is an indispensable task. However, issues such as the enormous sparsity and high dimensions of one-hot feature vectors hinder the recommender systems from precisely modeling user's characteristics. In addition, extracting local dependencies and hierarchical structures among fields has been the challenging work. Qu et al. [32] tried to solve these issues by applying a product layer to capture

feature interactions from multi-field categorical data after the embedding layer, but the model has the limitation that it only deals with 1 and 2-order feature interactions, and lacks the ability to model higher-order feature interactions, i.e., the 3-order and higher ones. However, the complexity of n-order inner product grows exponentially with the order of interactions, which results from the exhaustive calculations of inter-fields feature interactions. To solve this issue, we follow [32] and made the following improvements in the attributes-driven latent information extraction layer (USER × CANDIDATE):

- We introduce a compressed interaction network (CIN) [33] to model higher order feature interactions, which solves the high complex and time-consuming issue by compressing the high-order interaction vectors to a fixed value.
- Taking both low and high-order feature interactions into consideration, we utilize synthesizer [37] to interpret the contribution of each feature interaction to the candidate item. To our best knowledge, our model is the first one proposed to apply synthesizer in the field of recommender systems.

The graph inside the dark-green box in Figure 1 illustrates the work flow of the attributes-driven latent information extraction layer (USER × CANDIDATE). In the following paragraphs, we state the functionality and the working procedure of each component in the attributes-driven latent information extraction layer (USER × CANDIDATE).

The generation of feature interactions. Let $\boldsymbol{f}^{u,v_c} = [\boldsymbol{u}^{(1)}, ..., \boldsymbol{u}^{(m)}, \boldsymbol{v}_c^{(1)}, ..., \boldsymbol{v}_c^{(n)}] \triangleq [\boldsymbol{f}_1^{u,v_c}, \boldsymbol{f}_2^{u,v_c}, ..., \boldsymbol{f}_{m+n}^{u,v_c}]$ be the concatenation of $\boldsymbol{u}$ and $\boldsymbol{v}_c$. There are two procedures of generating feature interactions, i.e., the generations of (1) low-order feature interactions, and (2) high-order feature interactions. We define low-order as 1-order and 2-order. The representation of 1-order feature interactions $\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c} = [\boldsymbol{f}_1^{u,v_c}, \boldsymbol{f}_2^{u,v_c}, ..., \boldsymbol{f}_{m+n}^{u,v_c}]$ is the concatenation of products between $\boldsymbol{f}_i^{u,v_c} \in \boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}$ and the constant signal "1." For 2-order feature interactions, there are $\frac{(m+n)(m+n-1)}{2}$ 2-order feature interactions. The representation of 2-order feature interactions $\boldsymbol{\mathcal{F}}_{(2)}^{u,v_c}$ is defined as:

$$\boldsymbol{\mathcal{F}}_{(2)}^{u,v_c} = [\boldsymbol{f}_1^{u,v_c} \circ \boldsymbol{f}_2^{u,v_c}, \boldsymbol{f}_1^{u,v_c} \circ \boldsymbol{f}_3^{u,v_c}, ..., \boldsymbol{f}_i^{u,v_c} \circ \boldsymbol{f}_j^{u,v_c}, ..., \boldsymbol{f}_{m+n-1}^{u,v_c} \circ \boldsymbol{f}_{m+n}^{u,v_c}], \tag{2}$$

where $\circ$ denotes the Hadamard product, and $i \in [1, m+n-1], j \in [i+1, m+n]$. In addition to low-order feature interactions, we utilize a compressed interaction network (CIN) approach to generate higher-order feature interactions, whose order is defined to be higher than two in this paper. In CIN, there are multiple layers, each of which has different feature vectors. The h-th feature vector of the l-th layer in CIN is:

$$\boldsymbol{X}_h^l = \sum_{i=1}^{H_{l-1}} \sum_{j=1}^{m+n} \boldsymbol{w}_{ij}^{l,h} (\boldsymbol{X}_i^{l-1} \circ \boldsymbol{X}_j^0), \tag{3}$$

where $1 \leq h \leq H_l$ and $\boldsymbol{X}_j^0 = \boldsymbol{f}_j^{u,v_c}$; and $\boldsymbol{w}_{ij}^{l,h} \in \mathbb{R}^{H_{l-1} \times (m+n)}$ is the parameter matrix for the h-th feature vector. H_{l-1} denotes the number of feature interactions in the l-th layer:

$$H_{l-1} = \frac{(m+n)!}{(m+n-l)!\, l!}, \tag{4}$$

where $1 \leq l \leq O$, and O denotes the highest order of feature interactions to be modeled in this paper. Equation (3) implies that the order of interactions increases with the growth of the layer depth of CIN. Take as an example the generation of 3-order feature interactions. From Equation (3), we find that the 3-order feature interactions are the output vectors of the second layer in CIN:

$$\boldsymbol{\mathcal{F}}_{(3)}^{u,v_c} = [\boldsymbol{X}_1^2, \boldsymbol{X}_2^2, \cdots, \boldsymbol{X}_{H_2}^2]. \tag{5}$$

Similarly, the higher-order feature interactions can be obtained:

$$\boldsymbol{\mathcal{F}}_{(i)}^{u,v_c} = [\boldsymbol{X}_1^{i-1}, \boldsymbol{X}_2^{i-1}, \cdots, \boldsymbol{X}_{H_{i-1}}^{i-1}], \tag{6}$$

where $1 \leq i \leq O$. We define $\ell_{\boldsymbol{\mathcal{F}}_{(i)}^{u,v_c}}$ as the number of i-order feature interactions:

$$\ell_{\boldsymbol{\mathcal{F}}_{(i)}^{u,v_c}} = \begin{cases} m+n, & i=1 \\ \frac{(m+n)(m+n-1)}{2}, & i=2 \\ H_{i-1}, & i \geq 3 \end{cases} \tag{7}$$

The complete representation vector of low and high-order feature interactions is:

$$\boldsymbol{\mathcal{F}}^{u,v_c} = [\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}, \boldsymbol{\mathcal{F}}_{(2)}^{u,v_c}, \boldsymbol{\mathcal{F}}_{(3)}^{u,v_c}, ..., \boldsymbol{\mathcal{F}}_{(O)}^{u,v_c}]. \tag{8}$$

Thus, the number of low-order and high-order feature interactions $\ell_{\boldsymbol{\mathcal{F}}^{u,v_c}}$ is:

$$\ell_{\boldsymbol{\mathcal{F}}^{u,v_c}} = \sum_{i=1}^{O} \ell_{\boldsymbol{\mathcal{F}}_{(i)}^{u,v_c}}. \tag{9}$$

From Equations (7) and (9), it is obvious that $\boldsymbol{\mathcal{F}}_{(i)}^{u,v_c} \in \mathbb{R}^{\ell_{\boldsymbol{\mathcal{F}}_{(i)}^{u,v_c}} \times k}$ and $\boldsymbol{\mathcal{F}}^{u,v_c} \in \mathbb{R}^{\ell_{\boldsymbol{\mathcal{F}}^{u,v_c}} \times k}$.

The synthesizer. Synthesizer is an extension of the self-attention mechanism. The advantage of the synthesizer mechanism is that it replaces the inner product $\boldsymbol{QK}^\top$ in the vanilla self-attention mechanism with the synthesizing function $\varphi(\cdot)$, which results in reduced computational complexity; parameter costs which are approximately 10% lower than those for the vanilla self-attention mechanism; and competitive performance with the vanilla one [37]. Among its several variants, we utilized a mixed version of synthesizer, i.e., synthesizer (random + vanilla), which is the mixture of synthesizer random and vanilla self-attention models. The architecture of synthesizer (random + vanilla) is illustrated in Figure 2. The objective of synthesizer in the attributes-driven latent information extraction layer (USER $\times$ CANDIDATE) is to learn the different influences of each feature interaction on the candidate item. Therefore, we achieve it by applying $\boldsymbol{\alpha}_{syn} = \varphi(\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}) \in \mathbb{R}^{\ell_{\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}} \times \ell_{\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}}}$ as the feature interaction-level attention score matrix for $\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}$. We discuss the influences of feature interactions in Section 5.3.1. $\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}$ is used as the input, and the output vector $\widehat{\boldsymbol{\mathcal{F}}}^{u,v_c}$ of synthesizer is defined as:

$$\widehat{\boldsymbol{\mathcal{F}}}^{u,v_c} = \boldsymbol{\alpha}_{syn} \psi_{value}(\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}) = \varphi(\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}) \psi_{value}(\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}), \tag{10}$$

where

$$\varphi(\cdot) = \begin{cases} \boldsymbol{W}_{syn}, & \textit{Random Mode} \\ \text{Softmax}_{syn}(\psi_{query}(\cdot)\psi_{key}(\cdot)^\top), & \textit{Vanilla Mode} \\ \text{Softmax}_{syn}(\boldsymbol{W}_{syn} + \psi_{query}(\cdot)\psi_{key}(\cdot)^\top), & \textit{Random + Vanilla Mode} \end{cases}, \tag{11}$$

and

$$\text{Softmax}_{syn}(\cdot) = \frac{\exp(\cdot)}{\sum_1^{\ell_\cdot} \exp(\cdot)}. \tag{12}$$

$\psi_{query}(\cdot), \psi_{key}(\cdot), \psi_{value}(\cdot) \in \mathbb{R}^{\ell_{\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}} \times \ell_{\boldsymbol{\mathcal{F}}_{(1)}^{u,v_c}}}$ are multi-layer perceptron (MLP) units that are analogous to $Q(Query)$, $K(Key)$ and $V(Value)$ in the vanilla self-attention model, respectively:

$$\psi_\cdot(\cdot) = \text{MLP}(\cdot) = \sigma(\boldsymbol{W}(\cdot) + \boldsymbol{b}), \tag{13}$$

where $\boldsymbol{W}$, $\boldsymbol{b}$ and $\sigma(\cdot)$ are the weight, bias and non-linear activation function, respectively. Dice is utilized as the activation function in our model:

$$\sigma(x) = \frac{1}{1 + e^{-\frac{x - E[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}}}}, \tag{14}$$

where $E[x]$ and $Var[x]$ are the mean and variance of input, respectively, and ϵ is set to 10^{-8} following [8]. Note that $\boldsymbol{W}_{syn} \in \mathbb{R}^{\ell_{\boldsymbol{\mathcal{F}}^{u,v_c}_{(1)}} \times \ell_{\boldsymbol{\mathcal{F}}^{u,v_c}_{(1)}}}$ is a randomly initialized matrix which is applied to replace $QK^\top$ in the synthesizer (random) method. Therefore, given Equations (10) and (11), $\widehat{\boldsymbol{\mathcal{F}}}^{u,v_c}$ in random + vanilla mode is:

$$\widehat{\boldsymbol{\mathcal{F}}}^{u,v_c} = \varphi(\boldsymbol{W}_{syn} + \psi_{query}(\boldsymbol{\mathcal{F}}^{u,v_c}_{(1)})[\psi_{key}(\boldsymbol{\mathcal{F}}^{u,v_c}_{(1)})]^\top)\psi_{value}(\boldsymbol{\mathcal{F}}^{u,v_c}_{(1)}), \tag{15}$$

where $\widehat{\boldsymbol{\mathcal{F}}}^{u,v_c} \in \mathbb{R}^{\ell_{\boldsymbol{\mathcal{F}}^{u,v_c}_{(1)}} \times k}$. Thus, $\widehat{\boldsymbol{\mathcal{F}}}^{u,v_c}$ and $\boldsymbol{\mathcal{F}}^{u,v_c}$ are provided as the outputs of the attributes-driven latent information extraction layer (USER $\times$ CANDIDATE).

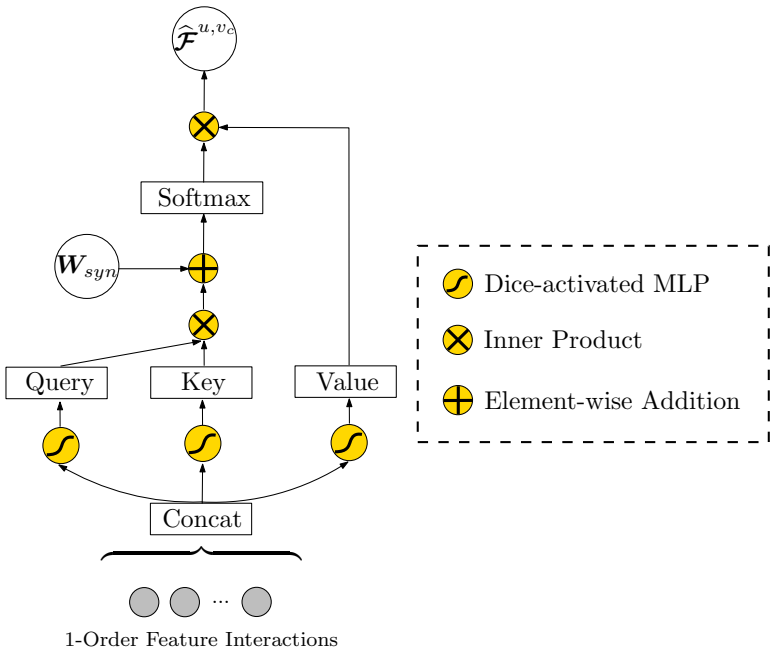

Figure 2. Synthesizer architecture.

3.4.2. User Behavior-Driven Latent Information Extraction Layer (HISTORY)

For the attributes-driven latent information extraction layer (USER $\times$ CANDIDATE), it learns the scope of the user's interest and tries to narrow it down by extracting latent information from feature interactions in the macro perspective. Additionally, a user's interests are diverse [8]. To precisely and comprehensively capture a user's diverse interests, similar to the activation unit applied in [8], we propose the user behavior-driven latent information extraction layer (HISTORY) to learn the latent information from user's historical behavior by modeling the user's history interactions, and investigating the different impact which each history item $v_i \in S_u^{v_c}$ has on the candidate item v_c. With these impacts, the intensity of u's interest in v_c can be measured, which plays a decisive role in determining the probability of clicking v_c. Differently from the self-attention-based synthesizer mechanism applied in the attributes-driven latent information extraction layer (USER $\times$ CANDIDATE), the attention mechanism utilized in this layer learns to assign a different attentive weight to each history item, to compute the importance of each item on the candidate item.

The graph inside the purple box in Figure 1 illustrates the architecture of the user behavior-driven latent information extraction layer (HISTORY). For each candidate item v_c of user u, $\boldsymbol{v}_c$ and $\boldsymbol{S}_u^{v_c}$ are used

as the inputs of the layer. The intensity of u's interest in v_c, i.e., $\mathcal{I}_u^{v_c}$, is calculated with a fully-connected neural network layer $\mathcal{M}$, as shown in Equation (16).

$$\mathcal{I}_u^{v_c} = f(\boldsymbol{v}_c, \boldsymbol{S}_u^{v_c}) = f(\boldsymbol{v}_c, \{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_{|\boldsymbol{S}_u^{v_c}|}\}) = \sum_{i=1}^{|\boldsymbol{S}_u^{v_c}|} \boldsymbol{w}_{ci}\boldsymbol{v}_i, \tag{16}$$

where

$$\boldsymbol{w}_{ci} = \frac{\exp(\boldsymbol{w}_{ci}^{'})}{\sum_1^{|\boldsymbol{S}_u^{v_c}|}\exp(\boldsymbol{w}_{ci}^{'})}, \tag{17}$$

$$\boldsymbol{w}_{ci}^{'} = \frac{\mathcal{M}(\boldsymbol{v}_c, \boldsymbol{v}_i)}{\sqrt{k}} = \frac{\mathrm{MLP}^L([\boldsymbol{v}_c; \boldsymbol{v}_i; \boldsymbol{v}_c - \boldsymbol{v}_i; \boldsymbol{v}_c \circ \boldsymbol{v}_i])}{\sqrt{k}} \tag{18}$$

and

$$\mathrm{MLP}^L(\cdot) = \mathrm{MLP}(\mathrm{MLP}(\cdots\mathrm{MLP}(\cdot))). \tag{19}$$

$\boldsymbol{w}_{ci}$ is treated as the correlation score between $\boldsymbol{v}_c$ and $\boldsymbol{v}_i$. We discuss the influence of $\boldsymbol{v}_i$ on $\boldsymbol{v}_c$ in Section 5.3.2. $[\boldsymbol{A}; \boldsymbol{B}]$ denotes the concatenation of vector $\boldsymbol{A}$ and vector $\boldsymbol{B}$. The working procedure of $\boldsymbol{w}_{ci}$ is illustrated in Figure 3. $\mathcal{I}_u^{v_c}$ is provided as the output of user behavior-driven latent information extraction layer (HISTORY).

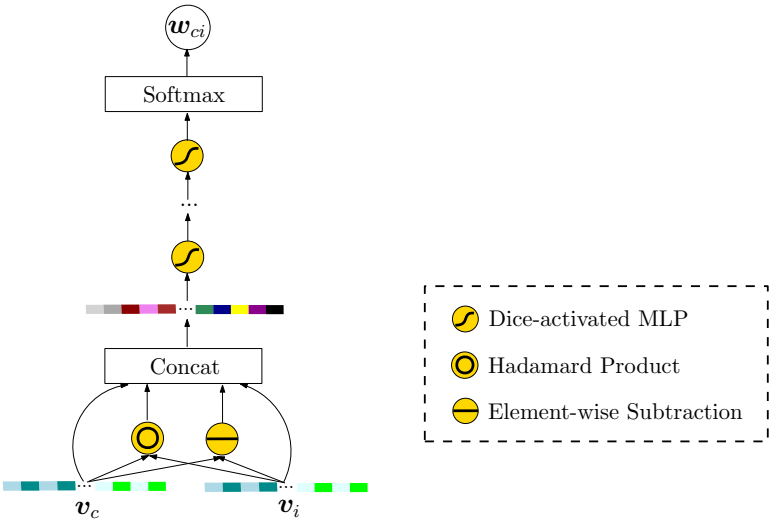

Figure 3. The working procedure of $\boldsymbol{w}_{ci}$.

3.4.3. Latent Information Digestion and Prediction Layer (PREDICTION)

The latent information digestion and prediction layer (PREDICTION) receives the outputs of the first two layers, i.e., $\widehat{\boldsymbol{\mathcal{F}}}^{u,v_c}$, $\boldsymbol{\mathcal{F}}^{u,v_c}$ and $\mathcal{I}_u^{v_c}$, and generates $\boldsymbol{\mathcal{P}}_u^{v_c}$, i.e., u's preference vector for v_c:

$$\boldsymbol{\mathcal{P}}_u^{v_c} = [\widehat{\boldsymbol{\mathcal{F}}}^{u,v_c}; \boldsymbol{\mathcal{F}}^{u,v_c}; \mathcal{I}_u^{v_c}]. \tag{20}$$

Next, $\boldsymbol{\mathcal{P}}_u^{v_c}$ is sent through a fully-connected layer, to calculate the probability of u clicking v_c:

$$prob(\boldsymbol{\mathcal{P}}_u^{v_c}) = \mathrm{MLP}^{L'}(\boldsymbol{\mathcal{P}}_u^{v_c}), \tag{21}$$

Finally, v_c's prediction score $\widehat{y}$ for u is:

$$\widehat{y} = \mathrm{sigmoid}(prob(\boldsymbol{\mathcal{P}}_u^{v_c})) = \mathrm{sigmoid}(\mathrm{MLP}^{L'}(\boldsymbol{\mathcal{P}}_u^{v_c})), \tag{22}$$

where $\widehat{y} \in \{0, 1\}$ and $\mathrm{sigmoid(x)} = \frac{1}{1+exp(-x)}$.

4. Experimental Setup

In this section, we present our experiments in detail, including datasets, baselines, evaluation metrics and hyperparameters. Experiments were conducted on two public datasets with user and item attributes and user behavior to investigate the effectiveness of our model.

4.1. Datasets

We experimented with two different datasets: MovieLens-1M (Movielens-1M dataset: https://grouplens.org/datasets/movielens/1m/) and Ad Display/Click Data on Taobao.com (Ad Display/Click Data on Taobao.com dataset: https://tianchi.aliyun.com/dataset/dataDetail?dataId=56). We name "Ad Display/Click Data on Taobao.com" dataset "Taobao" in the rest of the paper for short. The descriptive statistics for the datasets and attributes used as features are shown in Tables 2–4.

MovieLens-1M dataset. The MovieLens-1M dataset contains approximately one million explicit movie ratings (ranging from 1 to 5), users' demographic information and movies' basic information from the MovieLens website. We used both users' and items' attributes as the input features. To make it suitable for CTR prediction task, we followed [38] and transformed ratings into implicit feedback; each entry that was marked with 1 indicated that the user had rated the item positively, and we sampled negative examples from an unwatched set marked as 0 for each user, which had the same numbers as the rated ones. The threshold of positive ratings was set to 4. For the dataset splitting procedure, we split data based on userID; therefore, 3622 (60%), 1207 (20%) and 1207 (20%) users among 6036 users were randomly sampled into the training set (441,134 samples), validation set (149,128 samples) and test set (151,438 samples).

Taobao dataset. Taobao dataset contains over 26 million ad display/click logs from 8 days, users' and items' basic information from Taobao website. We used both users' and items' attributes as the input features. Samples whose "clk" field is 1 were treated as positive samples, otherwise as negative samples. Users with low activity, i.e., the ones who have less than 5 positive samples, were filtered from the dataset. Similar to the unwatched set generated for Movielens-1M, an unclicked set was sampled as negative samples for Taobao dataset, which had the same number with the positive ones. In addition to log data, attribute values of the numerical field, i.e., price, were normalized to the range [0, 1]. Same as the split method for Movielens-1M, we split Taobao dataset based on userID, thus 30,295 (60%), 10,097 (20%) and 10,097 (20%) users among 50,489 users were randomly sampled into training set (443,640 samples), validation set (148,264 samples) and test set (148,534 samples).

Table 2. Descriptive statistics for the datasets.

	MovieLens-1M	Taobao
Number of users	6036	50,489
Number of items	2347	376,764
Number of interactions	753,772	841,416
Median of interactions per user	78	14
Average number of interactions per user	125	17
Number of fields	6	13
Number of features	3486	460,995

Table 3. Attribute statistics for the Movielens-1M dataset.

Attribute Type (# Fields)	Field Name	Field Type	# Categories
User Attributes (4)	gender	Categorical	2
	age	Categorical	7
	occupation	Categorical	21
	zipcode	Categorical	3402
Item Attributes (2)	release_year	Categorical	36
	genre	Categorical	18

Table 4. Attribute statistics for the Taobao dataset.

Attribute Type (# Fields)	Field Name	Field Type	# Categories
User Attributes (8)	cms_segid	Categorical	95
	cms_group_id	Categorical	13
	final_gender_code	Categorical	2
	age_level	Categorical	7
	pvalue_level	Categorical	3
	shopping_level	Categorical	3
	occupation	Categorical	2
	new_user_class_level	Categorical	4
Item Attributes (5)	cate_id	Categorical	5273
	campaign_id	Categorical	233,398
	customer	Categorical	155,979
	brand	Categorical	66,215
	price	Numerical	1

4.2. Baselines

We compared our model with the following baseline algorithms.

IPNN [32]. IPNN is the PNN model with an inner product layer. PNN applies a product layer to capture 1-order and 2-order feature interactions from multi-field categorical data after the embedding layer. We used the same configurations in [32], except that the mini-batch size was set to 50.

OPNN [32]. OPNN is the PNN model with an outer product layer. We used the same configurations in [32], except that the mini-batch size was set to 50.

DeepFM [39]. DeepFM combines the power of factorization machines for recommendation and deep learning for feature learning in a neural network architecture. We used the same configurations in [39], except that the mini-batch size was set to 50.

DIN [8]. DIN uses a local activation unit to adaptively learn the representations of user interests from historical behaviors with respect to a certain item. We used the same configurations in [8], except that the mini-batch size was set to 50.

4.3. Evaluation Metrics and Hyperparameters

To evaluate the performance of each method, we employed two commonly used metrics AUC (area under ROC curve) [40] and ACC (accuracy). AUC is a widely used metric for evaluating classification problems. Reference [25] validated AUC as a good measurement in CTR estimation. Formally, the AUC of a classifier C is the probability that C ranks a randomly drawn positive sample x^+ higher than a randomly drawn negative sample x^-:

$$\mathrm{AUC}(C) = P\big[C(x^+) > C(x^-)\big]. \tag{23}$$

The second metric, ACC calculates the fraction of correctly classified samples:

$$\mathrm{ACC} = \frac{\#\ True\ Positives + \#\ True\ Negatives}{\#\ Samples}. \tag{24}$$

For hyperparameters, we experimented with different ones to find the best configurations of our method for each dataset. The configurations of our model are summarized in Table 5.

Table 5. Configurations for ALIEN.

	MovieLens-1M	**Taobao**
Embedding size	32	32
Learning rate	0.1	0.1
Dropout rate	0	0
$\text{MLP}^{L'}$ layer sizes	[800, 1]	[4500, 2300, 1200, 600, 1]
Highest order of feature interactions modeled	4	4
Mini-batch size	50	50

5. Results and Discussion

In this section, the performances of the proposed models and baselines are shown, according to the experimental settings we stated in the previous section. In addition, the superiority of the proposed models through comparisons of performance and the demonstration of their effectiveness in explainability are discussed.

5.1. Model Performance

Table 6 shows the performances of all methods under the metrics of AUC and ACC. Each experiment was repeated three times and the averaged results are reported. Relative scores are given compared to the strongest baselines, whose results are underlined. Note that a slightly higher AUC of 0.001 is regarded as significant for CTR predictions [23,39,41]. Therefore, for both datasets, ALIEN outperformed all baselines with large margins.

Table 6. AUC and ACC scores for the ALIEN models and the baselines for the test sets in Movielens-1M and Taobao datasets.

	Movielens-1M		**Taobao**	
	AUC	**ACC**	**AUC**	**ACC**
DIN	0.8808	0.7993	0.8223	0.7675
DeepFM	0.9257	0.8447	0.8665	0.7866
IPNN	0.9293	0.8536	0.8710	0.7880
OPNN	<u>0.9294</u>	<u>0.8543</u>	<u>0.8718</u>	<u>0.7916</u>
ALIEN	**0.9400 (+1.14%)**	**0.8670 (+1.49%)**	**0.8928 (+2.41%)**	**0.8114 (+2.50%)**

5.2. Comparison of Performances

OPNN had the strongest baseline on both the Movielens-1M and Taobao datasets. ALIEN outperformed OPNN with the improvements of 1.14% and 2.41% in AUC score; and 1.49% and 2.50% in ACC score for both the Movielens-1M and Taobao datasets, respectively. For both datasets, although DIN learns the representation of user interests from historical behavior with the candidate item, it has the worst performance without the modeling of feature interactions. For approaches which modeled low-order feature interactions, IPNN and OPNN held the lead compared with DeepFM on Movielens-1M, and the advantage of PNN's two variants against DeepFM became more obvious on the Taobao dataset. In terms of PNN's two variants, OPNN outperformed IPNN on both datasets. Despite being without the modeling of high-order feature interactions and user's historical behavior, they still provided competitive results. The good performance of ALIEN is attributed mainly to the traits that two types of latent information are taken into consideration simultaneously, i.e., attributes-driven and user-behavior-driven latent information, which are extracted from low and high-order feature interactions and the user's history items, respectively, to construct a comprehensive user profile and finally provide a more precise prediction. Please note that for the Movielens-1M dataset, as shown in Table 2, there were abundant interactions for each user but there were not enough fields of features, whereas for the Taobao dataset, interactions for each user were scarce but the number of feature fields was adequate. These results demonstrate that ALIEN can achieve a better performance when it is

short of features or the user's history interactions. In other words, ALIEN is capable of coping with severe situations.

Please note that in [8,39], the authors reported the experimental results of PNN, DeepFM and DIN on different datasets, i.e., Movielens-20M, Amazon, Alibaba, Criteo and Company [8,39], which were not utilized in this study. On Movielens-20M, Amazon and Alibaba datasets, DIN was reported to outperform IPNN and DeepFM, and DeepFM performed better than IPNN. On Criteo and Company datasets, DeepFM was reported to outperform the two variants of PNN, i.e., IPNN and OPNN. However, the analysis of the baselines' performances on these datasets is beyond the topic of this paper, since the valuable user and item attributes and the timestamps of each user's history interactions do not exist simultaneously in these five datasets, thereby making them fall short of the requirement for our experiment.

5.3. Effectiveness on the Explainability Problem (Case Study)

We conducted a real-world case study of user #7 and candidate item #861 in the Movielens-1M dataset, to present the capability of ALIEN to explain the recommendation results from two perspectives: (1) feature interactions, and (2) a user's historical behavior. The MovieLens-1M dataset was utilized as the dataset of this case study.

5.3.1. Influence of Feature Interactions

Figure 4 illustrates the relevance between different fields of attributes from the attention score $\boldsymbol{\alpha}_{syn}$ obtained by the synthesizer module in the attributes-driven latent information extraction layer (USER $\times$ CANDIDATE). From the red and black dashed rectangles, we can see that the user's age "35–44" and the movie's genre "action and thriller" contribute more than other fields of attributes and act as good ingredients when forming feature interactions with others. Specifically, the pair <age = "35–44", movie's genre = "action and thriller"> (i.e., the red square) is identified as the most influential feature interaction for the candidate item. It makes sense that middle-aged adults, especially men, are very likely to prefer action and thriller movies.

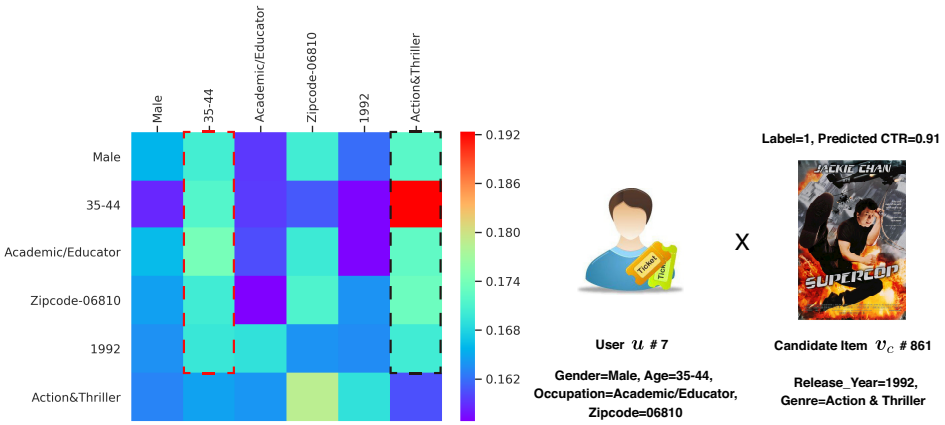

Figure 4. A heat map of attention weights for feature interactions on MovieLens-1M. The axes represent attribute fields of u and v_c (gender, age, occupation, zipcode, movie's release year, genre). Influential feature interactions are highlighted with bright colors (Label = 1, Predicted CTR = 0.91).

5.3.2. Influence of User's Historical Behavior

Figure 5 illustrates the relevance between different history items v_i and the candidate item v_c from the attention score $\boldsymbol{w}_{ci}$ obtained by the attention module in the user behavior-driven latent information extraction layer (HISTORY). The user discussed in Figure 5 is the same person discussed in Section 5.3.1. Note that the Movielens-1M dataset suffers a severe shortage of attributes, and there exists latent information which is not implied by the attributes. Therefore, generally, history items

which have the similar genre and release year as the candidate item are given a high relevance score. It is apparent from the figure that this male user loves action and thriller movies. The reason that v_{13} and v_{14} are weighted very low is due to the fact that he has watched two romance movies and is currently tired of that same genre. He now wants to get back to the action and thriller movies again and chooses to watch v_c.

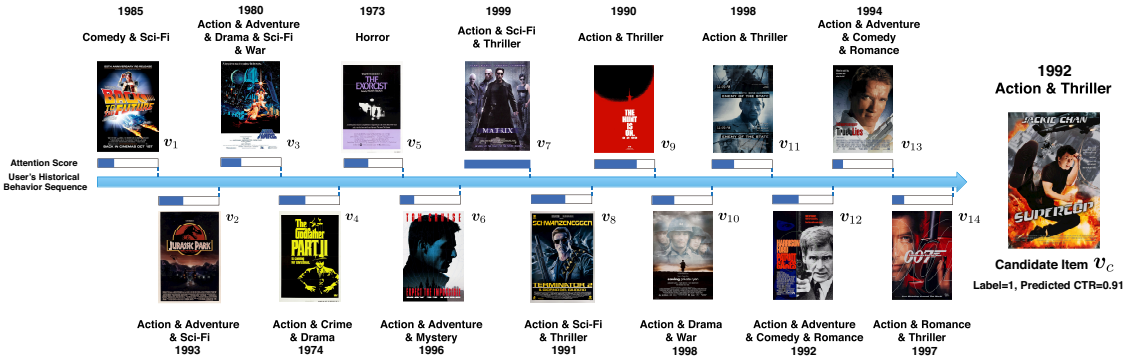

Figure 5. Illustration of the influence of user behavior on the candidate item. The genre and release year of each item are provided. Generally, history items with high relevance to candidate item get high attention scores.

6. Conclusions and Future Research

In this work, we developed a model titled the attention-based latent information extraction network (ALIEN) for CTR prediction. ALIEN is designed for common scenarios, such as the recommendation of movies on a video streaming website or products on an e-commerce shopping site, where the user and item attributes and the user's past clicking decisions are typically available. Based on the experimental results, ALIEN resolves the issues of: (1) modeling high-order feature interactions, and (2) the explainability of the prediction. To achieve these two objectives, ALIEN (i) constructs the low and high-order feature interactions from user and item attributes, via the vector inner-product approach combined with a compressed interaction network (CIN) module; (ii) extracts latent information from feature interactions and the user's history interactions with two attention-based layers to enhance the performance of CTR prediction; and simultaneously (iii) provides explainability by interpreting the contributions of different feature and history interactions. We have conducted experiments on two real-world datasets and demonstrated considerable improvements over strong baselines. Moreover, our proposed model can provide reasonable explanations, even when attributes are quite scarce. For future research, we aim to model other sources of attributes, particularly on a knowledge graph, to better characterize users and items and therefore make even more precise predictions.

Author Contributions: Conceptualization, R.H., S.M., M.S. and H.E.; methodology, R.H.; software, R.H.; validation, R.H., S.M., M.S., H.E. and Z.O.; formal analysis, R.H.; investigation, R.H.; resources, R.H., S.M., M.S. and H.E.; data curation, R.H.; writing—original draft preparation, R.H.; writing—review and editing, R.H., S.M., M.S., H.E. and Z.O.; visualization, R.H.; supervision, S.M., M.S., H.E. and Z.O.; project administration, M.S.; funding acquisition, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science and Technology Major Project (grant number 2018YFB1403003).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript (in alphabetical order):

ACC	Accuracy
ALIEN	Attention-Based Latent Information Extraction Network
AUC	Area Under ROC Curve
CF	Collaborative Filtering
CIN	Compressed Interaction Network
CTR	Click-Through Rate
DIN	Deep Interest Network
FM	Factorization Machines
IPNN	Inner Product-based Neural Network
MF	Matrix Factorization
MLP	Multi-Layer Perceptron
OPNN	Outer Product-based Neural Network
PNN	Product-based Neural Network
ROC	Receiver Operating Characteristic

References

1. Herlocker, J.L.; Konstan, J.A.; Terveen, L.; Riedl, J. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **2004**, *22*, 5–53. [CrossRef]
2. Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749. [CrossRef]
3. Hong, J.; Su, X.; Khoshgoftaar, T.M. A Survey of Collaborative Filtering Techniques. *Adv. Artif. Intell.* **2009**, *2009*, 421425. [CrossRef]
4. He, X.; Zhang, H.; Kan, M.; Chua, T. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pisa, Italy, 17–21 July 2016; pp. 549–558.
5. Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 426–434.
6. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
7. Chen, W.; Hsu, C.; Lai, Y.; Liu, V.; Yeh, M.; Lin, S. Attribute-Aware Recommender System Based on Collaborative Filtering: Survey and Classification. *Front. Big Data* **2019**, *2*, 49. [CrossRef]
8. Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; Gai, K. Deep Interest Network for Click-Through Rate Prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2018; pp. 1059–1068.
9. Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; Gai, K. Deep interest evolution network for click-through rate prediction. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 5941–5948.
10. Sun, Z.; Guo, Q.; Yang, J.; Fang, H.; Guo, G.; Zhang, J.; Burke, R. Research commentary on recommendations with side information: A survey and research directions. *Electron. Commer. Res. Appl.* **2019**, *37*, 100879. [CrossRef]
11. Adams, R.P.; Dahl, G.E.; Murray, I. Incorporating Side Information in Probabilistic Matrix Factorization with Gaussian Processes. In Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, 8–11 July 2010.
12. Zhao, F.; Guo, Y. Learning discriminative recommendation systems with side information. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3469–3475.
13. Guo, Y. Convex Co-Embedding for Matrix Completion with Predictive Side Information. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 1955–1961.

14. Kim, Y.M.; Choi, S. Scalable Variational Bayesian Matrix Factorization with Side Information. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, Reykjavik, Iceland, 22–25 April 2014; pp. 493–502.
15. Chen, J.; Zhang, H.; He, X.; Nie, L.; Liu, W.; Chua, T. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 335–344.
16. He, X.; Pan, J.; Jin, O.; Xu, T.; Liu, B.; Xu, T.; Shi, Y.; Atallah, A.J.; Herbrich, R.; Bowers, S.M.; et al. Practical Lessons from Predicting Clicks on Ads at Facebook. In Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, New York, NY, USA, 24 August 2014; pp. 5:1–5:9.
17. Juan, Y.; Zhuang, Y.; Chin, W.; Lin, C. Field-aware Factorization Machines for CTR Prediction. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 43–50.
18. Lee, K.C.; Orten, B.B.; Dasdan, A.; Li, W. Estimating conversion rate in display advertising from past performance data. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 768–776.
19. Rendle, S. Factorization Machines. In Proceedings of the 10th IEEE International Conference on Data Mining, Sydney, Australia, 14–17 December 2010; pp. 995–1000.
20. Song, W.; Shi, C.; Xiao, Z.; Duan, Z.; Xu, Y.; Zhang, M.; Tang, J. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1161–1170.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Thirty-first Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
22. Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; Chua, T. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6298–6306.
23. Cheng, H.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.D.; Aradhye, H.; Anderson, G.; Corrado, G.S.; Chai, W.; Ispir, M.; et al. Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 7–10.
24. Covington, P.; Adams, J.; Sargin, E. Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; pp. 191–198.
25. Graepel, T.; Candela, J.Q.; Borchert, T.; Herbrich, R. Web-Scale Bayesian Click-Through rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 13–20.
26. Mcmahan, H.B.; Holt, G.; Sculley, D.; Young, M.; Ebner, D.; Grady, J.P.; Nie, L.; Phillips, T.; Davydov, E.; Golovin, D.; et al. Ad click prediction: A view from the trenches. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 1222–1230.
27. Richardson, M.; Dominowska, E.; Ragno, R.J. Predicting clicks: Estimating the click-through rate for new ads. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 521–530.
28. Koehrsen, W. Why Automated Feature Engineering Will Change the Way You Do Machine Learning. Available online: https://towardsdatascience.com/why-automated-feature-engineering-will-change-the-way-you-do-machine-learning-5c15bf188b96 (accessed on 18 May 2020).
29. Rendle, S.; Gantner, Z.; Freudenthaler, C.; Schmidtthieme, L. Fast context-aware recommendations with factorization machines. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 25–29 July 2011; pp. 635–644.
30. Rendle, S.; Freudenthaler, C.; Schmidtthieme, L. Factorizing personalized Markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 811–820.
31. He, X.; Chua, T. Neural Factorization Machines for Sparse Predictive Analytics. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017; pp. 355–364.

32. Qu, Y.; Cai, H.; Ren, K.; Zhang, W.; Yu, Y.; Wen, Y.; Wang, J. Product-Based Neural Networks for User Response Prediction. In Proceedings of the IEEE 16th International Conference on Data Mining, Barcelona, Spain, 12–15 December 2016; pp. 1149–1154.
33. Lian, J.; Zhou, X.; Zhang, F.; Chen, Z.; Xie, X.; Sun, G. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2018; pp. 1754–1763.
34. Shan, Y.; Hoens, T.R.; Jiao, J.; Wang, H.; Yu, D.; Mao, J. Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 255–262.
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; Chua, T. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3119–3125.
37. Tay, Y.; Bahri, D.; Metzler, D.; Juan, D.C.; Zhao, Z.; Zheng, C. Synthesizer: Rethinking Self-Attention in Transformer Models. *arXiv* **2020**, arXiv:2005.00743.
38. Wang, H.; Zhang, F.; Zhao, M.; Li, W.; Xie, X.; Guo, M. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 2000–2010.
39. Guo, H.; Tang, R.; Ye, Y.; Li, Z.; He, X. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017.
40. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
41. Wang, R.; Fu, B.; Fu, G.; Wang, M. Deep & Cross Network for Ad Click Predictions. In Proceedings of the ADKDD'17, Halifax, NS, Canada, 13–17 August 2017; pp. 12:1–12:7.