



# 计算机科学

COMPUTER SCIENCE

## 基于神经架构搜索的点击率预测模型

帅剑波, 王金策, 黄飞虎, 彭舰

### 引用本文

帅剑波, 王金策, 黄飞虎, 彭舰. [基于神经架构搜索的点击率预测模型](#)[J]. 计算机科学, 2022, 49(7): 10-17.

SHUAI Jian-bo, WANG Jin-ce, HUANG Fei-hu, PENG Jian. [Click-Through Rate Prediction Model Based on Neural Architecture Search](#)[J]. Computer Science, 2022, 49(7): 10-17.

---

### 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

#### [基于概率元学习的矩阵补全预测融合算法](#)

Fusion Algorithm for Matrix Completion Prediction Based on Probabilistic Meta-learning

计算机科学, 2022, 49(7): 18-24. <https://doi.org/10.11896/jsjcx.210600126>

#### [一种改进的融合相似度和信任度的协同过滤算法](#)

Improved Collaborative Filtering Algorithm Combining Similarity and Trust

计算机科学, 2022, 49(6A): 238-241. <https://doi.org/10.11896/jsjcx.210400088>

#### [基于点割集图分割的矩阵变换与分解的推荐算法](#)

Matrix Transformation and Factorization Based on Graph Partitioning by Vertex Separator for Recommendation

计算机科学, 2022, 49(6A): 272-279. <https://doi.org/10.11896/jsjcx.210600159>

#### [基于遗憾探索的竞争网络强化学习智能推荐方法研究](#)

Study on Intelligent Recommendation Method of Dueling Network Reinforcement Learning Based on Regret Exploration

计算机科学, 2022, 49(6): 149-157. <https://doi.org/10.11896/jsjcx.210600226>

#### [基于注意力机制和门控网络相结合的混合推荐系统](#)

Hybrid Recommender System Based on Attention Mechanisms and Gating Network

计算机科学, 2022, 49(6): 158-164. <https://doi.org/10.11896/jsjcx.210500013>

# 基于神经架构搜索的点击率预测模型

帅剑波 王金策 黄飞虎 彭 舰

四川大学计算机学院 成都 610065

(122569217@qq.com)

**摘要** 点击率(Click-Through Rate,CTR)预测是推荐系统中一项重要的任务,其目标是预测用户点击一个广告或者商品的概率。特征嵌入和特征组合是影响预测性能的关键,因此很多点击率预测模型的思路也是针对这两个方面进行优化。但目前大部分工作仅关注其中一个方面,并且几乎所有的模型在进行特征组合时都没有对特征进行区分,同一个特征与其他特征组合时都使用相同的嵌入和组合方法,阻碍了模型性能的提升。为解决该问题,提出了 Auto-SEI(Automatic Super-field-aware Feature Embedding and Interacting)模型。该模型先将每个特征子域分配给一个特征超域,再根据分组得到特征的嵌入,然后为特征对选择合适的组合方法获取组合特征,最后进行预测。Auto-SEI 模型中,特征子域的划分和组合方法的选择被参数化为架构搜索问题,利用神经架构搜索(Neural Architecture Search,NAS)算法压缩搜索空间并进行选择。在 3 个真实的大规模数据集上进行了大量实验,结果表明 Auto-SEI 模型在点击率预测任务上具有优秀的性能。

**关键词:**神经架构搜索;点击率预测;特征嵌入;特征组合;推荐系统

**中图法分类号** TP391;TP183

## Click-Through Rate Prediction Model Based on Neural Architecture Search

SHUAI Jian-bo, WANG Jin-ce, HUANG Fei-hu and PENG Jian

College of Computer Science, Sichuan University, Chengdu 610065, China

**Abstract** Click-through rate(CTR) prediction is an important task in the recommendation system. Its goal is to predict the probability of a user clicking on an advertisement or item. Feature embedding and feature interacting are critical for prediction performance. Therefore, the ideas of many click-through rate prediction models are optimized based on these two aspects. However, most of the work only focus on one of the aspects, and almost all models do not distinguish features in feature interacting. The same embedding and interacting method are used when crossing the same feature with other features, which hinders the improvement of model performance. In order to solve this problem, the automatic super-field-aware feature embedding and interacting (Auto-SEI) model is proposed. Firstly, it assigns each sub-field to a super-field, and obtains the feature embedding according to the grouping, then selects appropriate interacting method for the feature pair to obtain the cross feature, and finally makes prediction. In Auto-SEI model, the division of sub-field and the selection of interacting methods are parameterized as an architecture search problem, and the neural architecture search(NAS) algorithm is used to compress the search space and make selections. A large number of experiments are conducted on three real large-scale data sets and the results show the excellent performance of the Auto-SEI model on the task of click-through rate prediction.

**Keywords** Neural Architecture Search, Click-Through rate prediction, Feature embedding, Feature interacting, Recommendation system

## 1 引言

近年来,点击率预测是学术界和工业界的研究热点,具有重要的研究意义和实际应用价值。例如,在推荐系统和在线广告应用中,每天有数以百万计的在线用户,即使点击率只有很小的提升,也可以带来巨额的收益<sup>[1-2]</sup>。

然而,CTR 预测是一项具有挑战性的任务,其难点主要体现在:1)输入特征的维度很大并且极度的稀疏<sup>[3]</sup>;2)高效地进行特征组合对于点击率的提升不可或缺<sup>[3-6]</sup>。特征嵌入和特征组合是影响预测性能的关键。特征嵌入:将所有 one-hot 稀疏向量表示的特征映射为维度相同的稠密向量表示;特征组合:为表达更加丰富和无偏的信息,选择合适的方法将原始

到稿日期:2021-06-02 返修日期:2021-10-19

基金项目:国家重点研发计划(2017YFB0202403);四川省重点研发计划(2020YFG0308,2019YFG0494,2020YFG0304,2017GZDZX0003)

This work was supported by the National Key R & D Program of China(2017YFB0202403) and Key R & D Program of Sichuan Province, China (2020YFG0308,2019YFG0494,2020YFG0304,2017GZDZX0003).

通信作者:彭舰(jianpeng@scu.edu.cn)

特征嵌入进行组合,得到高阶特征,以提升模型性能<sup>[7]</sup>。例如, $gender$ 是一个特征域,包含两个特征, $male$ 和 $female$ 。先将其变成稀疏的 one-hot 向量: $gender=female \rightarrow filed_{gender}=(0,1)$ 和 $gender=male \rightarrow filed_{gender}=(1,0)$ ;接着将其映射为相同维度的稠密向量 $\mathbf{V} \in R^d$ ,然后进行特征组合。如一个表示青年女性的组合特征可以使用哈达玛积来组合: $\mathbf{V}_{gender=female,age=20} = \mathbf{V}_{gender=female} \cdot \mathbf{V}_{age=20}, \mathbf{V}^* \in R^d$ 。

现有模型,如因子分解机( Factorization Machine, FM)<sup>[8]</sup>、域感知的因子分解机(Field-aware Factorization Machine, FFM)<sup>[9]</sup>、深度因子分解机(Deep Factorization Machine, DeepFM)<sup>[11]</sup>及深度交叉网络(Deep&Cross, D&C)<sup>[4]</sup>等,通过增强特征嵌入或者优化特征组合取得了预测性能的提升。这些方法的共同特点是只关注了影响性能的两个要素之一,或者增强特征嵌入,或者改进组合方法。从理论上来说,如果能够得到最优的嵌入或者找到最优的组合方法,那么另一个要素对性能的影响就微乎其微。但是,没有方法能够保证得到最优嵌入或者最优组合<sup>[10]</sup>,因此,增强特征嵌入,同时获取合适的组合方法,是提升点击率预测性能的关键。此外,几乎所有方法在进行嵌入和组合时都没有对特征进行明确区分,同一个特征与其他特征组合都使用相同的嵌入和组合方法,从而阻碍了模型性能的提升。

因此,本文提出了自动化超域感知的特征嵌入与组合模型来解决点击率预测的特征嵌入与组合问题。首先,模型将所有的特征子域,如年龄(Age)、性别(Gender)及广告形式(Advertise-type)等进行分组,每个分组都是一个特征超域;然后,根据划分出的超域加强特征嵌入和优化特征组合。对于特征嵌入的增强,Auto-SEI使用超域感知的嵌入,类似于在 FFM 中使用子域感知嵌入;对于特征组合,模型在两两超域之间从候选组合方法中自适应地选择最优组合方法。另外,Auto-SEI 模型还能在一定程度上解决梯度耦合的问题<sup>[2,11]</sup>。

超域的划分和方法选择的组合数太多,无法枚举。受交互函数搜索模型<sup>[12]</sup>和点击率模型中特征选择方法的启发<sup>[13-15]</sup>,本文的将组合问题转化为架构搜索问题,利用 NAS 方法进行数据驱动的超域划分和组合方法选择。

本文主要贡献如下:

(1)提出了 Auto-SEI 模型,在同一个特征和其他特征进行组合的时候,可以根据组合的特征来使用不同的嵌入和组合方法;

(2)将 NAS 方法应用于点击率预测模型中,实现了自动化及数据驱动的特征嵌入和组合;

(3)在 3 个真实的大规模数据集上进行了实验,验证了模型的有效性,并进行了细致的分析。

本文第 2 节介绍了近年来关于点击率预测模型和神经架构搜索的相关工作;第 3 节首先通过一个子模块描述了搜索算法,随后详细介绍了 Auto-SEI 模型的整个流程;第 4 节介绍了实验设置、对比算法以及对实验结果的分析;最后总结全文。

## 2 相关工作

### 2.1 点击率预测模型

点击率预测模型性能的提升,离不开将特征进行嵌入和进行高效的特征组合这两个方面。由于 FM 模型简洁高效,

后续研究中涌现出了如 FFM<sup>[9]</sup>、域感知概率嵌入网络(Field-aware Probabilistic Embedding Neural Network, FPENN)<sup>[16]</sup>、基于积的神经网络(Product-based Neural Networks, PNN)<sup>[11]</sup>及注意力因子分解机(Attentional Factorization Machine, AFM)<sup>[17]</sup>等 FM 的扩展模型。在这些模型中,FFW 和 FPENN 使用了不同类型的子域感知嵌入来对特征嵌入进行增强。AFM 和 PNN 则分别通过结合注意力机制<sup>[18]</sup>和子域感知的特征组合方法来优化组合特征构建及增强表达能力。除了基于 FM 的模型外,其他模型则是通过设计一种交互网络来增强嵌入或者构造组合特征。例如,文献[4]提出了 D&C 模型,使用了交互层(Cross Layer)来增强原始特征的表达能力;文献[19]提出了基于卷积神经网络的特征生成模型,利用卷积网络(Convolutional Neural Network, CNN)来获取局部特性,并将其组合成新的特征;文献[3]提出了自动特征组合学习模型,利用残差连接的多头自注意网络(Multi-heads Self-attention Network)优化特征的组合;文献[5]提出了极深因子分解机模型,利用压缩组合网络(Compressed Interaction Network, CIN)来高效获取向量级别的组合特征。由于深度神经网络(Deep Neural Network, DNN)已在其他数据科学领域(如计算机视觉和自然语言处理方向)成功应用,许多点击率预测模型也利用 DNN 来获取隐式的高阶组合特征,如深宽模型(Deep & Wide)<sup>[20]</sup>、神经因子分解机(Neural Factorization Machine, NFM)<sup>[21]</sup>等模型。这些模型中,Deep & Wide 需要人工构建组合特征,其他模型则自动进行特征组合。此外,文献[22]提出的操作感知的神经网络模型和文献[23]提出的自动交互网络框架生成模型也旨在解决特征组合问题,但这两种方法是在模型层面解决问题,而 Auto-SEI 模型是在特征域层面进行解决。

### 2.2 神经架构搜索

近年来, NAS 常被用于发现一个更高效的用于图像分类的卷积网络架构,或者用于自然语言处理的循环网络架构<sup>[24-26]</sup>。NAS 方法可以大致分为三大类:基于进化的方法、基于强化学习的方法和基于梯度的方法。本文主要使用基于梯度的方法,因为它比其他两种方法更高效。文献[25]提出了可微的架构搜索模型(Differentiable Architecture Search, DARTS),其是一种基于梯度的 NAS 方法。首先训练每一个操作(卷积操作使用的卷积核)被选择的权重(概率),然后选择具有最大权重(概率)的卷积核来组成最终的卷积网络。该方法在连续的权重编码和最终的离散架构之间存在着不可避免的误差。文献[24]提出基于近端迭代的神经架构搜索模型(NAS Method Based on Proximal Iterations, NASP),通过引入近端算法,强制架构搜索阶段只有单一卷积核被选择来解决不一致的问题。Auto-SEI 模型将超域的划分和特征组合方法的选择都参数化为架构搜索的问题,但是直接将 NASP 方法应用于所有的架构参数可能会导致架构不一致问题并最终得到一个次优解。因此在实际应用过程中,不同部分的架构参数训练会采用不同的 NAS 方法,但为了提高搜索效率,采用的都是基于梯度的方法。

## 3 Auto-SEI 模型

### 3.1 概述

文中主要使用的符号如表 1 所列。本节将详细介绍本文

提出的 Auto-SEI 模型。如图 1 所示,模型的主要工作可以分为 3 个部分:自动进行特征超域的划分(Automatic Grouping)、特征嵌入(Feature Embedding)和特征组合(Feature Interacting)。特征超域的划分,将  $m$  个特征子域划分成为  $g$  个特征超域,划分结果指导特征的嵌入和组合。特征嵌入,选择和 FFM 子域感知类似的嵌入方式,使用超域感知的嵌入,因此每个特征有多个嵌入(具体数量由超域个数  $g$  决定)。特征组合,模型针对来自不同超域的特征,为每个特征对选择最适合的组合函数。最后,模型的最终输出由 4 个子模块的输出共同决定。特征超域的划分和特征组合都被当作架构搜索的问题来处理,因此下文会先介绍搜索算法,接着介绍模型的工作流程。

表 1 符号说明

Table 1 Symbol descriptions

Symbols	Meaning
$n$	The number of features
$m$	The number of sub-fields
$k$	The embedding dimensions
$e_{i,j}$	The $j^{\text{th}}$ embedding of features in the $i^{\text{th}}$ sub-field
$g$	The number of super-field
$s_a$	The $a^{\text{th}}$ super-field(a sub-field set)
$o^{(a,b)}$	The feature interaction function between $s_a$ and $s_b$
$p_k^i$	The probability of sub-field $i$ selected by $s_k$
$e_{s_a}$	The embedding of the $a^{\text{th}}$ super-field
$w_{i,j}$	The weight of feature interaction between the $i^{\text{th}}$ and $j^{\text{th}}$ features

### 3.2 搜索算法

本节只关注模型第一部分如何进行分组和第三部分如何进行组合函数的选择,而忽略模型的其他部分,从而可以更清晰地介绍模型使用的搜索方法,如图 1 所示。

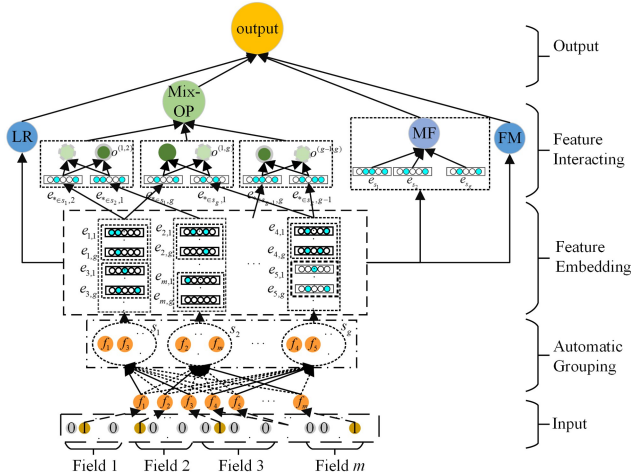


图 1 Auto-SEI 模型概览

Fig. 1 Overview of Auto-SEI

用 Mix-OP 标记的节点输出可以表示为:

$$\sum_{a=1}^g \sum_{b=a+1}^g \sum_{i \in s_a} \sum_{j \in s_b} w_{i,j} o^{(a,b)} \epsilon(e_i, e_j) \quad (1)$$

其中,  $\epsilon(\cdot)$  表示特征嵌入所使用的函数,  $o(\cdot)$  表示特征组合函数。  $\epsilon(\cdot)$  的作用是让进行嵌入的两个特征根据对方所属

超域来确定自己使用哪一个嵌入,  $o(\cdot)$  的作用是根据两个特征所属的超域选择一个函数进行特征组合。显然,函数  $\epsilon(\cdot)$  和  $o(\cdot)$  中的搜索空间都是离散的,在这两个离散空间中进行搜索需要消耗大量计算资源。受益于 DARTS<sup>[25]</sup> 和 NASP<sup>[24]</sup> 这类基于梯度的 NAS 方法,在放宽特征超域对特征子域的选择和特征超域之间对组合函数的选择后,可以将其表示为连续空间的一组向量。因此将  $\epsilon(\cdot)$  转变为:

$$\epsilon(e_i, e_j) = \left( \sum_{k=1}^g \mathcal{H}(p_k^i) e_{i,k}, \sum_{k=1}^g \mathcal{H}(p_k^j) e_{j,k} \right)$$

$$p_k^i = \frac{\alpha_k^i}{\sum_{c=1}^g \alpha_c^i} \quad (2)$$

$$\mathcal{H}(p_k^i) = \begin{cases} 1, & p_k^i \geq p_x^i, \forall x \in [1, g], x \neq k \\ 0, & \text{else} \end{cases}$$

转变后,特征子域被选择的情况就可以用向量  $\alpha^i \in \mathcal{R}^g$  来表示。特征嵌入函数  $\epsilon(\cdot)$  根据不同的  $\alpha$  会得到不同的嵌入。  $\mathcal{H}(\cdot)$  函数有两个作用:1)将连续的架构参数向量  $\alpha$  离散化得到一个唯一的选择;2)限制每一个特征子域仅属于某一个具体的特征超域,而不是依概率属于多个特征超域。类似的转变也应用于  $o(\cdot)$ :

$$\bar{o}^{(a,b)}(\cdot) = \sum_{o \in \mathcal{O}} \frac{\exp(\beta_o^{(a,b)})}{\sum_{o' \in \mathcal{O}} \exp(\beta_{o'}^{(a,b)})} o(\cdot) \quad (3)$$

其中,向量  $\beta^{(a,b)} \in \mathcal{R}^{|\mathcal{O}|}$  表示选择特征组合函数的权重,  $\mathcal{O}$  表示组合函数候选集。转换之后,在离散空间进行架构搜索的问题就变成了学习连续的架构变量。在 Auto-SEI 模型中,使用了  $\alpha$  和  $\beta$  两组架构变量,其中  $\alpha^i$  被用来表示特征子域  $f_j$  被每个特征超域选择的概率,  $\beta^{(a,b)}$  表示超域  $s_a$  和  $s_b$  之间使用每个特征组合函数的概率。因此,搜索阶段的目标可以表示为:

$$\min_{\alpha, \beta} \mathcal{L}_{\text{val}}(\omega^*(\alpha, \beta), (\alpha, \beta))$$

$$\text{s. t. } \omega^*(\alpha, \beta) = \arg \min_{\omega} \mathcal{L}_{\text{train}}(\omega, (\alpha, \beta)) \quad (4)$$

其中,  $\omega$  表示模型的所有非架构参数,  $\mathcal{L}_{\text{val}}$  和  $\mathcal{L}_{\text{train}}$  分别表示验证集损失和训练集损失。式(4)是一个双层优化问题。架构参数  $\alpha, \beta$  和非架构参数  $\omega$  共同影响验证集和训练集,因此在搜索阶段要找到一组最优的架构参数  $(\alpha_{\text{opt}}, \beta_{\text{opt}})$  来最小化验证集损失  $\mathcal{L}_{\text{val}}(\omega^*(\alpha, \beta), (\alpha, \beta))$ , 其中非架构参数  $\omega^*$  表示可以最小化训练集损失的非架构参数,即  $\omega^*(\alpha, \beta) = \arg \min_{\omega} \mathcal{L}_{\text{train}}(\omega, (\alpha, \beta))$ 。完成搜索之后,就可以根据连续的架构变量  $\alpha$  和  $\beta$  构造出离散的架构。例如,特征组合的函数选择表示为  $o^{(a,b)} = \arg \max_{o \in \mathcal{O}} \beta_o^{(a,b)}$ , 特征嵌入表示为  $\epsilon(\cdot) = (e_{i,k} = \arg \max_k \alpha_k^i, e_{j,k} = \arg \max_k \alpha_k^j)$ 。

### 3.3 重训练

完成搜索之后,下文从输入到输出来详细介绍重训练阶段。除了超域划分和超域对之间的特征组合函数已经确定外,搜索过程和重训练过程的其他部分是一致的。

#### 3.3.1 输入

点击率预测模型中的特征都是离散的(连续特征通常也会利用分桶方法映射为离散值),通常利用一个 one-hot 向量来表示这些特征。例如,一个数据实例 ( $City = \text{ChengDu}$ ,

<sup>1)</sup>  $o(e_i, e_j) = [o(e_{i_1}, e_{j_1}), o(e_{i_2}, e_{j_2}), \dots, o(e_{i_g}, e_{j_g})]$ , 为了简化模型,组合候选集  $\mathcal{O} = \{\max, \min, \text{plus}, \text{multiply}\}$

$Gender=Female, Weather=Rain$ )为:

$$(010\cdots0) \quad (10) \quad (010\cdots0) \quad (5)$$

$City=ChengDu, Gender=Female, Weather=Rain$

因此数据集中的每一个输入实例都可以表示为  $\mathbf{x}=[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$ , 其中  $\mathbf{x}_i \in \mathbb{R}^{l_i}$ ,  $l_i$  是特征子域  $i$  的特征的个数,  $\mathbf{x}_i[j] \in \{0, 1\}$ ,  $j \in [1, l_i]$ , 且  $\sum_{j=1}^{l_i} \mathbf{x}_i[j]=1$ .

### 3.3.2 嵌入

输入实例中的绝大多数特征都是高维稀疏的,导致输入实例也是高维稀疏的,因此无法直接在这样的输入上进行特征组合和点击率预测。最直接的方法就是要把它们嵌入成低维稠密的向量,比如:

$$\mathbf{e}_i = \mathbf{V}_i \mathbf{x}_i \quad (6)$$

其中,  $\mathbf{V}_i \in \mathbb{R}^{l_i \times k}$  是子域  $i$  的嵌入矩阵,  $\mathbf{x}_i$  是一个如式(5)中的 one-hot 向量。然而本文提出的 Auto-SEI 模型稍有不同,由于每个特征子域被划分到不同超域,两个子域特征进行组合时需根据对方的超域归属选择不同的嵌入。假设划分了  $g$  个超域,每个子域特征都会拥有  $g$  个嵌入。将式(6)转换为:

$$\mathbf{e}_{i,j} = \mathbf{V}_{i,j} \mathbf{x}_i, 1 \leq j \leq g \quad (7)$$

其中,  $\mathbf{V}_{i,j} \in \mathbb{R}^{l_i \times k}$  是第  $i$  个特征子域的第  $j$  个嵌入矩阵,  $\mathbf{e}_{i,j}$  是第  $i$  个特征子域在一个具体的输入实例中的第  $j$  个嵌入。特征嵌入之后,输入实例就可以表示为:

$$\mathbf{x} = [\underbrace{\mathbf{e}_{1,1}, \dots, \mathbf{e}_{1,g}}_{\text{embedding of field 1}}, \underbrace{\mathbf{e}_{2,1}, \dots, \mathbf{e}_{2,g}}_{\text{embedding of field 2}}, \dots, \underbrace{\mathbf{e}_{m,1}, \dots, \mathbf{e}_{m,g}}_{\text{embedding of field m}}] \quad (8)$$

### 3.3.3 特征组合

当所有的特征都映射为同一个低维空间的稠密向量之后,就可以在这个低维空间进行特征的组合和预测任务。Auto-SEI 模型可以进一步划分为 4 个子模块。

LR 子模块:一个线性回归模块,其输出在图 1 上用标记了 LR 的节点表示。该模块通常被用来提取原始特征嵌入的全局偏差。但在 Auto-SEI 模型中,原始特征不止一个嵌入,其嵌入数量等同于超域的数量,这表明需要一个聚合函数来聚合多个嵌入,之后再应用线性回归模型。因此,线性回归模块的输出可以表示为:

$$\Phi_{LR} = \tau_0 + \sum_{i=1}^m \tau_i \text{AGG}(\mathbf{e}_{i,1}, \dots, \mathbf{e}_{i,g}) \quad (9)$$

其中,  $\tau_0$  是偏差,  $\text{AGG}()$  表示聚合函数。本文的 Auto-SEI 模型选择平均聚合函数,即  $\text{AGG}(\mathbf{e}_{i,1}, \dots, \mathbf{e}_{i,g}) = (1/g) \sum_{j=1}^g \mathbf{e}_{i,j}$ 。其他聚合函数如相加或者拼接也可以应用,聚合函数对预测性能的影响将在以后的工作中探索。

FM 子模块:一个因子分解机模型,其输出在图 1 上用标记了 FM 的节点表示。该模块用于组合来自相同特征超域的子域特征。例如,第  $a$  个特征超域的特征组合为  $FM(s_a) = \sum_{i \in s_a} \sum_{j \in s_a} \mathbf{w}_{i,j}^T \mathbf{e}_{i,a} \odot \mathbf{e}_{j,a}$ , 其中  $\mathbf{w}_{i,j} \in \mathbb{R}^k$ ,  $(\mathbf{e}_{i,a} \odot \mathbf{e}_{j,a})[k] = \mathbf{e}_{i,a}[k] \mathbf{e}_{j,a}[k]$ 。最终,第 2 个子模块的输出可以表示为:

$$\Phi_{FM} = \sum_{a=1}^g FM(s_a) \quad (10)$$

Mix-OP 子模块:一个混合组合函数模块,其输出在图 1 中用标记了 Mix-OP 的节点表示。该模块用于组合来自不同特征超域的子域特征。当特征超域和它们之间的组合函数都确定之后,就可以通过改变式(1)来表示这个子模块的输出:

$$\Phi_{\text{Mix-OP}} = \sum_{a=1}^g \sum_{b=a+1}^g \sum_{i \in s_a} \sum_{j \in s_b} \mathbf{w}_{i,j}^T o^{(a,b)}(\mathbf{e}_{i,b}, \mathbf{e}_{j,a}) \quad (11)$$

其中,  $\mathbf{w}_{i,j}^T \in \mathbb{R}^k$ , 并且  $o^{(a,b)}$  仅代表一个被具体选择的组合函数:  $o^{(a,b)} = \arg \max_{o \in \mathcal{O}} \beta_o^{(a,b)}$ , 而非式(3)中的多个组合函数的混合结果  $\bar{o}^{(a,b)}$ 。为便于表示和计算的一致性<sup>1)</sup>,接下来使用  $\mathbf{w}_{i,j} = \mathbf{w}_{i,j}^1 \cup \mathbf{w}_{i,j}^2$  表示子域特征组合的所有权重。

MF 子模块:一个类似于协同过滤中的矩阵分解模块,其输出在图 1 中用 MF 标记的节点表示。可以把每个特征超域看作一个可用于预测的特征,该模块用于组合超域级别的特征。但在组合前,表示一个超域级别的特征要面临两个问题: 1)超域中有多个子域的特征,应用什么聚合函数来聚合它们? 2)每个子域特征也拥有多个嵌入,哪些子域嵌入用来聚合成超域嵌入?

出于简化模型的考虑,在表示第  $a$  个超域特征时使用子域特征的第  $a$  个嵌入  $\mathbf{e}_{s_a}$ , 然后使用平均聚合函数聚合内部子域特征来表示超域特征:  $\mathbf{e}_{s_a} = 1/|s_a| \sum_{i \in s_a} \mathbf{e}_{i,a}$ 。最终,第 4 个子模块输出表示为:

$$\Phi_{MF} = \sum_{a=1}^g \sum_{b=a+1}^g \mathbf{w}_{a,b} \mathbf{e}_{s_a} \odot \mathbf{e}_{s_b} \quad (12)$$

其中,  $\mathbf{w}_{a,b} \in \mathbb{R}^k$  表示超域特征  $a$  和超域特征  $b$  组合表示的权重。与 LR 子模块一样,其他的聚合函数也可以应用于这一个子模块的超域特征表示过程中。

### 3.3.4 输出

整个模型的输出由 4 个子模块的输出共同组成:  $\Phi_{LR}$ ,  $\Phi_{FM}$ ,  $\Phi_{\text{Mix-OP}}$  及  $\Phi_{MF}$ 。如 3.3.3 节所述,为了计算的一致性,将第二个子模块和第三个子模块一起表示:

$$\Phi_{FM} + \Phi_{\text{Mix-OP}} = \sum_{a=1}^g \sum_{b=a+1}^g \sum_{i \in s_a} \sum_{j \in s_b} \mathbf{w}_{i,j}^T o^{(a,b)}(\mathbf{e}_{i,b}, \mathbf{e}_{j,a}) \quad (13)$$

最终输出可以表示为:

$$\hat{y} = \sigma(\mathbf{W}_{\text{Auto-SEI}}^T [\Phi_{LR}, [\Phi_{FM} + \Phi_{\text{Mix-OP}}], \Phi_{MF}]) \quad (14)$$

其中,  $\mathbf{W}_{\text{Auto-SEI}} \in \mathbb{R}^3$  是一个权重向量,其作用是将所有的子模块输出联合起来;  $\sigma$  表示一个非线性转换函数:  $\sigma(x) = 1/1 + e^{-x}$ ;  $\hat{y}$  表示模型最终输出,即预测的点击概率。

### 3.3.5 训练

点击率预测的模型可以看作一个二分类的问题,可以使用交叉熵损失作为预测的损失函数,因此整个模型损失可以表示为:

$$\mathcal{L} = - \underbrace{\frac{1}{N} \sum_{i=1}^N y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)}_{\text{log loss}} + \lambda \|\mathbf{V}\| \quad (15)$$

其中,  $y_i \in \{0, 1\}$  表示真实的点击事件,  $\hat{y}_i \in (0, 1)$  表示预测用户点击的概率,  $N$  表示训练样本数,  $\mathbf{V}$  表示模型的嵌入参数,  $\lambda$  表示正则化系数。最终目标就是优化模型参数,使得损失  $\mathcal{L}$  达到最小。

#### 算法 1 Auto-SEI 模型训练流程

输入:训练集  $\mathcal{D}_{\text{train}}$ , 验证集  $\mathcal{D}_{\text{valid}}$ , 搜索轮数  $S$ , 重训练轮数  $R$ , 子域个数  $m$ , 超域个数  $g$ , 损失计算函数  $L$ , 参数更新函数  $\mathcal{G}$

输出:结构参数  $(\boldsymbol{\alpha}_{\text{opt}}, \boldsymbol{\beta}_{\text{opt}})$ , 非结构参数  $\boldsymbol{\omega}^*$

1. 初始化:  $\boldsymbol{\alpha} \sim \mathcal{N}(1, 0.01)$ ,  $\boldsymbol{\beta} \sim \mathcal{N}(1, 0.01)$ ,  $\boldsymbol{\omega} \sim \mathcal{N}(0, 0.01)$

2. for  $t=1$  to  $S$  do

Feature embedding function,  $\boldsymbol{\alpha} \rightarrow \boldsymbol{\epsilon}(\cdot)$  based on 式(2)

Feature interacting function:  $\beta \rightarrow o(\cdot)$  based on 式(3)

$$\omega = \mathcal{G}(\mathcal{L}_{\text{train}} = L((\alpha, \beta, \omega), \mathcal{D}_{\text{train}}))$$

$$(\alpha, \beta) = \mathcal{G}(\mathcal{L}_{\text{valid}} = L((\alpha, \beta, \omega), \mathcal{D}_{\text{valid}}))$$

$$3. (\alpha_{\text{opt}}, \beta_{\text{opt}}) = (\alpha, \beta)$$

$$4. \alpha_{\text{opt}}^j = \text{softmax}(\alpha_{\text{opt}}^j) \text{ for } j=1, \dots, m$$

$$5. \beta_{\text{opt}}^{(a,b)} = \text{softmax}(\beta_{\text{opt}}^{(a,b)}) \text{ for } 1 \leq a < b \leq g$$

$$6. \epsilon_{\text{opt}}(\cdot) = (\mathbf{e}_{i, k = \arg \max_k \alpha_k^i}, \alpha_i, k = \arg \max_k \alpha_k^i) \text{ based on } \alpha_{\text{opt}}$$

$$7. o_{\text{opt}}^{(a,b)} = \arg \max_{o \in \mathcal{O}} \beta_o^{(a,b)} \text{ based on } \beta_{\text{opt}}$$

$$8. \text{重新初始化: } \omega \sim \mathcal{N}(0, 0.01)$$

$$9. \text{for } t=1 \text{ to } R \text{ do}$$

Embedding function:  $\epsilon_{\text{opt}}(\cdot)$ , Interacting function:  $o_{\text{opt}}(\cdot)$

$$\omega = \mathcal{G}(\mathcal{L}_{\text{train}} = L((\alpha_{\text{opt}}, \beta_{\text{opt}}, \omega), \mathcal{D}_{\text{train}}))$$

$$\text{loss}_t = L((\alpha_{\text{opt}}, \beta_{\text{opt}}, \omega), \mathcal{D}_{\text{valid}})$$

if  $\text{loss}_t > \text{loss}_{t-1}$ : break

$$10. \omega^* = \omega$$

Auto-SEI 模型的训练流程如算法 1 所示,其训练时计算的消耗主要在损失计算和参数更新两个过程中。一个样本计算损失的复杂度为  $O(km^2 + gk)$ ,在训练集需计算  $|\mathcal{D}_{\text{train}}|$  次,在验证集需计算  $|\mathcal{D}_{\text{valid}}|$  次。步骤 9 每一轮都要在训练集和验证集上计算损失,并更新参数  $\omega$ ;步骤 2 每一轮也在训练集和验证集上计算损失,并更新参数  $\omega, \alpha, \beta$ ,但是由于  $\alpha, \beta$  参数量远远小于  $\omega$ ,并且  $\omega$  中的主要参数是嵌入参数  $\mathbf{V}$ ,忽略了其他参数,两个步骤参数更新的复杂度近似为  $\mathbf{V}$  参数量  $nk g$  的线性倍数  $O(nk g)$ 。综上所述,总的复杂度约为  $O((S+R) \cdot k(|\mathcal{D}_{\text{train}}| + |\mathcal{D}_{\text{valid}}|)(m^2 + g) + nk g)$ 。

## 4 实验

本节主要是基于 3 个公共数据集的实验结果及分析,分为以下 3 个部分:

(1) 本文提出的模型与其他模型相比,在点击率预测的任务上表现如何?

(2) 一些超参数如何影响模型的性能?

(3) 超越的划分是否有一定的可解释性?

### 4.1 实验设置

本文实验使用了 Criteo<sup>1)</sup>, Avazu<sup>2)</sup> 和 iPinYou<sup>3)</sup> 3 个数据集,统计信息如表 2 所列。使用 AUC(Area Under the ROC Curve) 和模型的交叉熵损失(Logloss)作为验证指标。其中, AUC 越高,交叉熵损失越低,表明模型的性能越好。参与对比的基准模型如下。

表 2 数据集统计信息

Table 2 Statistics of data sets

Dataset	Train samples	Valid samples	Sub-fields	features	Positiveratio
Criteo	86 883 012	12 733 031	39	1 178 909	0.50
Avazu	32 143 173	8 085 794	24	645 195	0.17
iPinYou	15 395 285	4 100 716	16	937 670	0.07

LR<sup>[3]</sup>: 利用原始特征进行逻辑回归分类,能够较好地学

习到全局的偏差。

FM<sup>[8]</sup>: 利用因子分解机对原始特征进行自动组合,可以有效利用二阶组合特征。

CrossNet<sup>[4]</sup>: 利用 CrossNet 网络加强原始特征嵌入,使得单个特征表达的信息更丰富。

CIN<sup>[5]</sup>: 利用 CIN 网络加强特征的组合方法,可以高效地组合和利用高阶特征。

模型对比时的参数设置如下:所有的模型嵌入维度固定为 32。根据原论文说明,对于 CrossNet 模型设置 *crosslayer* 为 2; CIN 模型的 *crosslayer* 设置为 3。对于学习率和正则化参数,采用网格搜索的方法进行选择,学习率搜索空间为  $\{0.001, 0.002, 0.005, 0.01, 0.02\}$ ,正则化参数搜索空间为  $\{1, 2, 5\} * \{1 \times 10^{-6}, 1 \times 10^{-5}, 1 \times 10^{-4}, 1 \times 10^{-3}\}$ 。另外,对每个模型都进行了早停策略,使用 Adam 优化器进行优化。对于 iPinYou 数据集,设置 *batchsize* 为 2048;其他两个数据集的 *batchsize* 设置为 10000。如果没有特别说明,对比模型的其他独有参数根据原文章进行设置,Auto-SEI 模型的超越个数在 iPinYou 数据集上的实验设置为 3,在其他两个数据集上的实验设置为 4。

### 4.2 对比实验

表 3 列出了模型之间的性能对比,可以得出以下几个结论:首先,所有进行了特征组合的模型性能都优于线性模型 LR,表明在 CTR 预测任务中原始特征表达能力的不足和组合特征的重要性;其次, CrossNet 模型和 CIN 模型分别增强了特征嵌入和优化了特征组合,在大部分情况下都优于 FM 模型,说明加强特征嵌入和优化组合方法的重要性(CrossNet 性能在某些数据集的表现不如 FM,原因是它增强的特征表示相当于原始特征的一个线性倍数,文献[5]中已经证明);最后, Auto-SEI 模型在 3 个数据集上的预测性能都优于对比模型,进一步说明同时优化特征嵌入和特征组合方法可以继续提升模型预测性能。另外, Auto-SEI 模型在 iPinYou 数据集上的提升最明显,说明其能够很好地在稀疏数据集上进行有效的学习。

表 3 预测性能

Table 3 Prediction performance

Model	Criteo		Avazu		iPinYou	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
LR	0.7801	0.5629	0.7643	0.3887	0.7305	0.006173
FM	0.7872	0.5555	0.7728	0.3850	0.7361	0.006039
CrossNet	0.7854	0.5572	0.7720	0.3853	0.7427	0.005707
CIN	0.7942	0.5478	0.7782	0.3819	0.7465	0.005701
Auto-SEI	0.7955	0.5465	0.7808	0.3799	0.7649	0.005620

图 2 给出了模型之间的响应时间对比,可以看出 Auto-SEI 模型在 iPinYou 和 Avazu 数据集上的响应时间都优于预测性能最好的对比模型 CIN,在 Criteo 数据集上也几乎达到了相同的响应速度。此外, LR, FM 和 CrossNet 3 个简单的模型在响应时间上有优势,但 CIN 和 Auto-SEI 的响应时间仍然符合要求,并且可以利用分布式的方法减小这种差距。

<sup>1)</sup> <http://labs.criteo.com/downloads/download-terabyte-click-logs/>

<sup>2)</sup> <http://www.kaggle.com/c/avazu-ctr-prediction>

<sup>3)</sup> <http://data.computational-advertising.org>

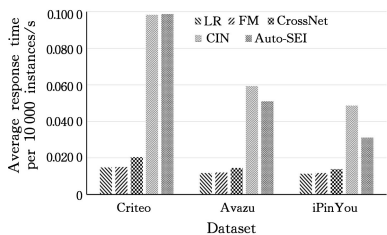
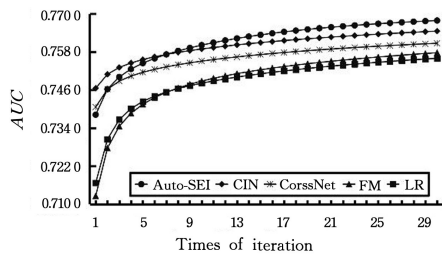


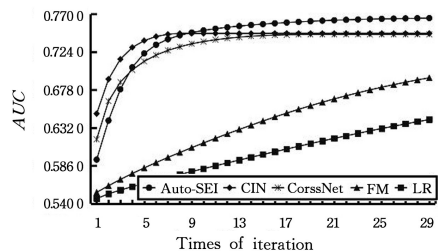
图2 响应时间

Fig. 2 Response time

图3和图4分别给出了模型AUC上升的趋势和交叉熵损失下降的趋势(模型在Criteo数据集上的表现和在Avazu数据集上的表现几乎一样,为了精简文章,未进行展示)。



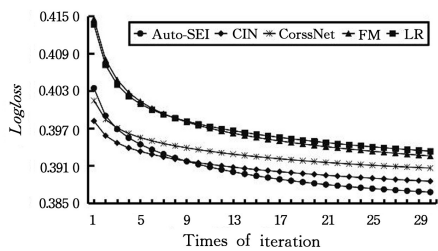
(a) AUC on Avazu dataset



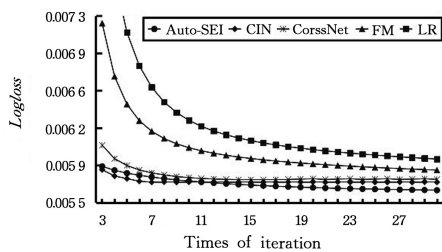
(b) AUC on iPinYou dataset

图3 AUC变化趋势

Fig. 3 Upward trend of AUC



(a) Logloss on Avazu dataset



(b) Logloss on iPinYou dataset

图4 Logloss变化趋势

Fig. 4 Downward trend of Logloss

可以得出如下结论:1)所有其他模型的收敛速度都比LR快,说明只利用原始特征很难有效学习;2)CIN是收敛速度最快的模型,但是在大约9轮迭代之后其收敛速度几乎

不再提升,最终被Auto-SEI模型超过;3)除了收敛速度略低于CIN,Auto-SEI模型相比其他模型拥有更快的收敛速度和更好的预测性能。

### 4.3 消融实验

本节通过消融实验来验证特征嵌入的加强和特征组合函数的优化都有利于模型性能的提升。因此,将原始Auto-SEI模型稍作改变,建立以下几个变体来进行对比实验。

(1)随机模型(Random):分组和特征组合函数的选择都是完全随机的。

(2)自然分组(N-type):将子域按照原本的自然类型进行划分,但组合函数仍然进行选择。

(3)单组合函数(S-):子域分组和原始模型一样,但特征组合函数只使用候选集中的一个。

表4 消融实验结果

Table 4 Results of ablation experiments

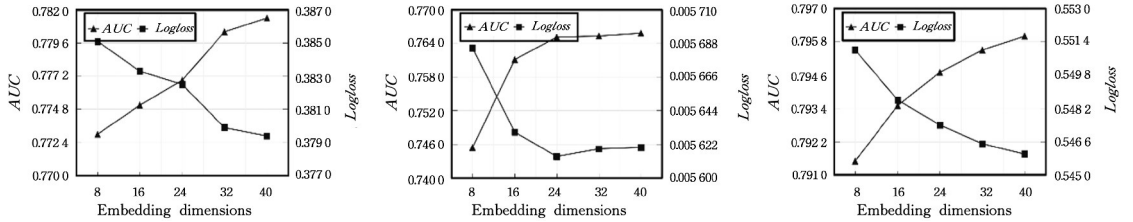
Variant	Avazu		iPinYou		Criteo	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
Random	0.7767	0.3822	0.7623	0.005626	0.7928	0.5495
N-type	—	—	0.7508	0.005664	—	—
S-Plus	0.7761	0.3827	0.7529	0.005658	0.7915	0.5510
S-Max	0.7779	0.3816	0.7549	0.005656	0.7934	0.5487
S-Min	0.7781	0.3814	0.7590	0.005642	0.7936	0.5485
Auto-SEI	0.7808	0.3799	0.7649	0.005620	0.7955	0.5465

所有的变体模型中超参数的设置都和原始模型保持一致。对比结果如表4所列,可以得出如下的结论:1)在iPinYou数据集上,随机变体模型性能优于其他的变体模型,因为iPinYou数据集预处理之后只有16个子域,是所有数据集中子域个数最少的,所以其受随机划分的影响更小;2)在单组合变体中,Max和Min变体模型比Plus变体模型性能更好(Multiply变体模型等同于FM模型,表3已列出对比结果),因为非线性组合函数通常能学习到更多的信息;3)Auto-SEI原始模型比其他的变体模型性能更好,说明增强特征嵌入和优化特征组合函数的方法都是有效的。

另外,自然变体模型在iPinYou数据集上表现最差。由于Avazu和Criteo数据集中的一部分子域名称做了匿名化处理,无法区分其代表的自然类型,因此没有进行自然分组变体的比较实验。

### 4.4 超参分析实验

Auto-SEI模型中有两个非常重要的超参数:特征嵌入的维度和超域的个数。图5和图6分别给出了这两个超参数对模型的影响。可以得出:1)模型性能随着嵌入维度的增大而提高,然而当嵌入维度超过24时,iPinYou数据集上交叉熵损失开始上升,这是因为iPinYou数据集相对更小且稀疏,更容易发生过拟合;2)超域个数越多,通常拥有更好的性能,但是也更容易引起模型过拟合,如在iPinYou数据集上,5个超域的性能已经开始出现下降;3)在Avazu和Criteo数据集上,更大的嵌入维度和更多的超域都能获得更好的性能,但是随着这两个参数的增大,模型的提升有所放缓,增加的参数却不变(总嵌入参数为 $nkg$ ,特征个数 $n$ 是固定的,参数量和嵌入维度 $k$ ,超域个数 $g$ 成正比),因此需要做一个平衡,在保证模型性能的同时保持其精简。



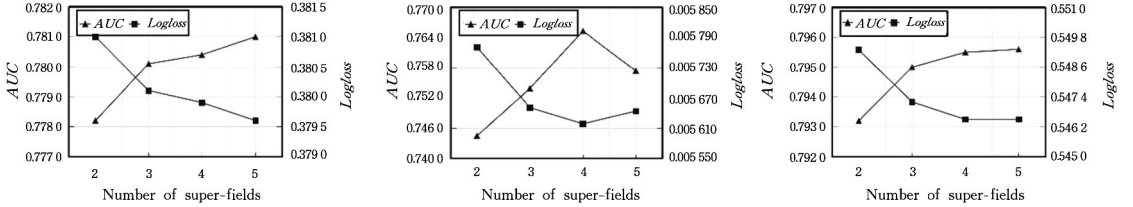
(a) Different embedding dimensions on Avazu

(b) Different embedding dimensions on iPinYou

(c) Different embedding dimensions on Criteo

图 5 嵌入维度对预测性能的影响

Fig. 5 Impact of embedding dimensions on prediction performance



(a) Different super-fields on Avazu

(b) Different super-fields on iPinYou

(c) Different super-fields on Criteo

图 6 超域个数对预测性能的影响

Fig. 6 Impact of super-fields on prediction performance

#### 4.5 可解释性实验

在预处理时,去除了 iPinYou 数据集的一些无用子域,其中用户子域只剩下用户代理(User Agent)和用户 IP 地址(Internet Protocol Address)。如果把剩下的子域分为两个超域,利用自然类型可以很顺利地完成任务;但是如果要分为 3 个超域,由于用户子域太少,模型往往很难获得较好的性能。在 4.3 节已经证明了利用自然分组的效果在所有变体模型中是最差的。另外,如果要分为更多的超域,利用自然类型进行划分也会越来越困难(因为 Avazu 和 Criteo 数据集对子域名称做了匿名化处理,所以未进行此实验)。

模型自动划分为 3 个超域的结果如图 7 所示。

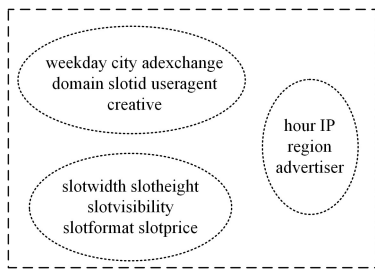


图 7 子域分组结果

Fig. 7 Results of sub-field division

由图 7 可知,除了 slotid,其余的产品子域都被划分到一个超域,另外两个超域由用户子域和一些上下文相关子域混合组成。可以看出,模型一方面能够感知到子域的自然类型,并且尽量将其分为一组,因为这些子域往往更加相似;另一方面,模型也会根据数据集本身的特点,将同一自然类型的子域划分到不同的超域中,以获得更好的预测性能。

**结束语** 点击率预测问题可以看作一个用户与项目之间的匹配问题,或者一个二分类问题,因此提升此类模型性能的一个方向就是加强原始特征的表达能力和进行高效的特征组

合。据此,本文提出了 Auto-SEI 模型,在嵌入和组合两方面对模型进行提升,并且引入神经架构搜索的方法提升模型的效率。模型在 3 个真实的大规模数据集上的实验结果表明,相比基准算法,本文模型的预测性能得到了提升,验证了模型的有效性。此外,本文还对模型进行了重要超参数的分析实验以及可解释性实验,对算法的特性进行了进一步的分析,为算法的实际应用提供了一些参考。

未来的研究工作将集中在以下两个方面:1)模型中两处使用了聚合函数,结合注意力机制来进行加强;2)利用一些辅助信息来指导模型更好地进行特征子域的划分和组合函数的选择,从而进一步提升模型的表达能力。

#### 参考文献

- [1] GUO H, TANG R, YE Y, et al. DeepFM: a factorization-machine based neural network for CTR prediction [C] // Proceedings of the 26th International Joint Conference on Artificial Intelligence. 2017: 1725-1731.
- [2] CHEN W, ZHAN L, CI Y, et al. FLEN: Leveraging Field for Scalable CTR Prediction [J]. arXiv: 1911. 04690, 2019.
- [3] SONG W, SHI C, XIAO Z, et al. AutoInt: Automatic feature interaction learning via self-attentive neural networks [C] // Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 1161-1170.
- [4] WANG R, FU B, FU G, et al. Deep & cross network for ad click predictions [C] // Proceedings of the ADKDD'17. 2017: 1-7.
- [5] LIAN J, ZHOU X, ZHANG F, et al. xdeepfm: Combining explicit and implicit feature interactions for recommender systems [C] // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 1754-1763.
- [6] PAN J, XU J, RUIZ A L, et al. Field-weighted factorization machines for click-through rate prediction in display advertising

- [C]//Proceedings of the 2018 World Wide Web Conference. 2018;1349-1357.
- [7] HUANG F, QIAO S, PENG J, et al. STPR: A personalized next point-of-interest recommendation model with spatio-temporal effects based on purpose ranking [J]. IEEE Transactions on Emerging Topics in Computing, 2019, 9(2): 994-1005.
- [8] RENDLE S. Factorization machines [C]//2010 IEEE International Conference on Data Mining. IEEE, 2010: 995-1000.
- [9] JUAN Y, ZHUANG Y, CHIN W S, et al. Field-aware factorization machines for CTR prediction [C]//Proceedings of the 10th ACM Conference on Recommender Systems. 2016: 43-50.
- [10] WANG X, HE X, WANG M, et al. Neural graph collaborative filtering [C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019: 165-174.
- [11] QU Y, FANG B, ZHANG W, et al. Product-based neural networks for user response prediction over multi-field categorical data [J]. ACM Transactions on Information Systems (TOIS), 2018, 37(1): 1-35.
- [12] YAO Q, CHEN X, KWOK J T, et al. Efficient neural interaction function search for collaborative filtering [C]//Proceedings of The Web Conference 2020. 2020: 1660-1670.
- [13] LIU B, XUE N, GUO H, et al. AutoGroup: Automatic feature grouping for modelling explicit high-order feature interactions in CTR prediction [C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020: 199-208.
- [14] LIU B, ZHU C, LI G, et al. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction [C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 2636-2645.
- [15] LUO Y, WANG M, ZHOU H, et al. Autocross: Automatic feature crossing for tabular data in real-world applications [C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 1936-1945.
- [16] LIU W, TANG R, LI J, et al. Field-aware probabilistic embedding neural network for ctr prediction [C]//Proceedings of the 12th ACM Conference on Recommender Systems. 2018: 412-416.
- [17] XIAO J, YE H, HE X, et al. Attentional factorization machines: Learning the weight of feature interactions via attention networks [J]. arXiv:1708.04617, 2017.
- [18] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.
- [19] LIU B, TANG R, CHEN Y, et al. Feature generation by convolutional neural network for click-through rate prediction [C]//The World Wide Web Conference. 2019: 1119-1129.
- [20] CHENG H T, KOC L, HARMSSEN J, et al. Wide & deep learning for recommender systems [C]//Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. 2016: 7-10.
- [21] HE X, CHUA T S. Neural factorization machines for sparse predictive analytics [C]//Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017: 355-364.
- [22] YANG Y, XU B, SHEN S, et al. Operation-aware Neural Networks for user response prediction [J]. Neural Networks, 2020, 121: 161-168.
- [23] SONG Q, CHENG D, ZHOU H, et al. Towards automated neural interaction discovery for click-through rate prediction [C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 945-955.
- [24] YAO Q, XU J, TU W W, et al. Efficient neural architecture search via proximal iterations [C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020: 6664-6671.
- [25] LIU H, SIMONYAN K, YANG Y. Darts: Differentiable architecture search [J]. arXiv:1806.09055, 2018.
- [26] XIE S, ZHENG H, LIU C, et al. SNAS: stochastic neural architecture search [J]. arXiv:1812.09926, 2018.



**SHUAI Jian-bo**, born in 1996, postgraduate, is a student member of China Computer Federation. His main research interests include recommendation algorithms and data mining.



**PENG Jian**, born in 1970, Ph.D, professor, Ph.D supervisor, is a outstanding member of China Computer Federation. His main research interests include big data and wireless sensor network.

(责任编辑:柯颖)