我用特征工程+LR超过了 xDeepFM!

阿泽的学习笔记 前天

以下文章来源于kaggle竞赛宝典 ，作者Data Magic

kaggle竞赛宝典

数据竞赛加分骚操作 & 数据分析方法 & 实践机器学习 & Kaggle + 天池 + 其他

之前对于特征工程的了解知之甚少，后来和杰少，峰少等朋友聊完之后，也自己跑了一些竞赛，深受启发，之前一直认为特征工程是艺术，但现在我个人更倾向于认为它是一门技术，它与模型相辅相成，特征工程要做的事情就是帮助模型，模型预测不好的地方，那么我们人为的用经验或者构建的特征来帮助它，使得模型能把自己做不好的地方能做好。**所以特征工程师95%的技术+5%的艺术（很多真的太难想到了）**。

最近刚好读了一篇我个人也较为感兴趣的自动化特征的文章(第四范式的)，文章通过自动化特征的方式结合LR模型使得LR的效果超过了xDeepFM等最新的神经网络效果。虽然不可思议，但是在某种意义上确实是可以做到的，只要特征足够强。好了，闲话不多说，我们直接阅读一下论文吧。

AutoCross: Automatic Feature Crossing for Tabular Data in Real-World Applications

背景

Motivation

交叉特征通过原始特征的cross乘积的向量化构建得到：

$$c_{i,j,...,k} = vec(f_i \otimes f_j \otimes \ldots \otimes f_k)$$

其中f_i是二元特征向量(hash trick之后套one-hot), $vec(\odot)$ 向量化一个张量，一个交叉特征也是一个二元特征向量。如果一个交叉特征使用三个或者更多的原始特征，我们就用高阶的交叉特征来表示它。

基于search的生成方法使用明显的搜索策略来构建特征集合的有用特征，大量的此类方法专注于数值特征并且没有产出交叉特征。

一方面，基于搜索的特征生成方法采用显式搜索策略来构造有用的特征或特征集。许多这样的方法集中于数值特征，而不产出交叉特征。对于现有的特征交叉方法，它们没有被设计成执行高阶特征交叉，因此效率低下。

Table 1: Comparison between AutoCross and other feature generation methods for tabular data.

Method	High-order Feature Cross	Simplicity	Fast Inference	Interpretability
Search-based methods (e.g., [5, 34])	×	medium	√	√
Implicit deep-learning-based methods (e.g., [33, 42])	×	low	×	×
Explicit deep-learning-based methods (e.g., [26, 37])	×	low	×	√
AutoCross	√	high	√	√

方案

在这一块，我们详细地看一下AutoCross的算法。

问题定义

我们假设所有的原始特征都是类别的，数据被表示为Multi-field的类别形式，其中每个field是一个从encoding得到的二元向量，给定训练数据$\mathcal{D}_{TR}$, 我们将其划分为子训练集$\mathcal{D}_{tr}$以及一个验证集合$\mathcal{D}_{vld}$,我们用特征集合$\mathcal{S}$表示$\mathcal{D}_{tr}$，用学习算法$\mathcal{L}$学习一个模型$\mathcal{L}(\mathcal{D}_{tr}, \mathcal{S})$, 我们使用相同的特征集合验证集$\mathcal{D}_{vld}$并计算一个metric，最终我们就尝试最优化：$\mathcal{E}(\mathcal{L}(\mathcal{D}_{tr}, \mathcal{S}), \mathcal{D}_{vld}, \mathcal{S})$

我们定义特征交叉问题为：

$$max_{\mathcal{S} \subseteq A(\mathcal{F})} \mathcal{E}(\mathcal{L}(\mathcal{D}_{tr}, \mathcal{S}), \mathcal{D}_{vld}, \mathcal{S})$$

其中

- $\mathcal{F}$为原始特征集合$\mathcal{D}_{TR}$;
- $A(\mathcal{F})$为原始特征集合和从$\mathcal{F}$中产出的可能的交叉特征;

特征集合产出

$$card(A(\mathcal{F})) = \sum_{i=1}^{d} C(d,k) = 2^d - 1$$

所有可能特征集合的数目为：2^{2^d-1}, 这个几乎是无法接受的。所以此处我们使用迭代构建局部最优特征子集的方式来贪心挖掘特征。

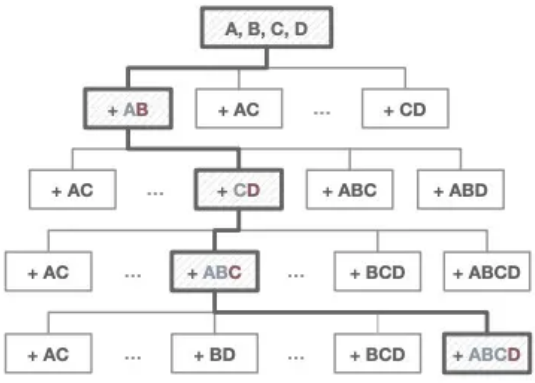

Figure 3: An illustration of the search space and beam search strategy employed in AutoCross. In beam search, only the best node (bold stroke) at each level is expanded. We use two colors to indicate the two features that are used to construct the new cross feature.

我们将两个特征A和B的交叉表示为AB, 成对特征的交叉就会产出更加高阶的特征。新的空间$\mathcal{T}$考虑$A(\mathcal{F})$中所有可能的特征交叉，同时忽略它的部分子集，搜索一个特征集合等价于从根节点$\mathcal{T}$到一个特定节点识别出一条路径。

这可以通过不断加入交叉特征到一个维护的特征集合中，但是，$\mathcal{T}$的大小是$P((d^2/2)^k)$,其中k是生成交叉特征的最大数。所以枚举出所有可能的解也是非常昂贵的。此处我们使用beam search的策略来解决该问题。

beam search的思想：在搜索过程中只扩展最有前途的节点。首先生成根节点的所有子节点，评估其对应的特征集，然后选择性能最好的节点进行下一次访问。在接下来的过程中，我们扩展当前节点并访问其最有希望的子节点。当过程终止时，我们在一个被认为是解决方案的节点处结束.

通过beam search，我们只性需要考虑$O(kd^2)$的节点。

特征集的评估

为了提升模型的评估效率，本文提出了field-wise LR以及连续的mini-batch GD档案。

1. field-wise LR

此处有两种假设：

1. 我们做了两个近似值。首先，我们使用经过小批量梯度下降训练的logistic回归（LR）来评估候选特征集，并用相应的性能来近似实际跟踪的学习算法L的性能。我们选择logistic回归作为一种广义线性模型，是大规模机器学习中应用最广泛的模型。它具有简单、可扩展、推理速度快、可解释性强等特点；
2. 在模型训练的时候，我们仅仅学习新增加的交叉特征的权重，其他的权重则被固定。所以训练时“filed-wise”的。举例来说，我们有一个特征集合$\mathcal{S}^* = \{A, B, C, D\}$, 我们希望对候选集$\mathcal{S} = \{A, B, C, D, AB\}$进行评估，在训练的时候只有AB的权重会被更新，我们用$x = [x_s^T, x_c^T]^T$进行表示,x_s表示之前所有特征，x_c是新增加的交叉特征；他们对应的权重为:$w = [w_s^T, w_c^T]^T$, LR会做下面的预测：

$$p(y = 1|x) = s(w^T x) = s(w_s^T x_s + w_c^T x_c) = s(w_c^T x_c + b_{sum})$$

其中$s(\cdot)$为sigmoid函数。更新的框架如下：

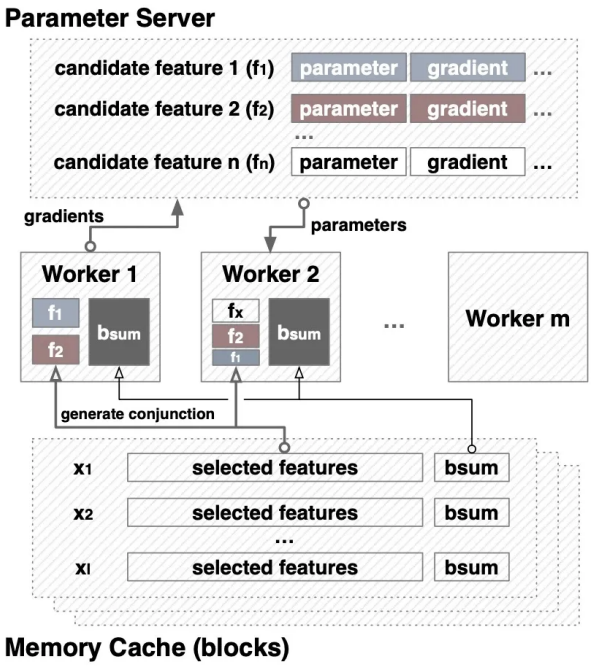

Figure 4: Illustration of field-wise logistic regression for feature evaluation based on a parameter server architecture.

它的优势如下：

- 存储，workers只需要存储x_c和b_{sum};
- 计算速度：快速，因为我们只需要更新w_c;

2. Successive Mini-batch Gradient Descent

Algorithm 2 Successive Mini-batch Gradient Descent (SMBGD).

Require: set of candidate feature sets $\mathbb{S} = \{\mathcal{S}_i\}_{i=1}^{n}$, training data equally divided into $N \geq \sum_{k=0}^{\lceil \log_2 n \rceil - 1} 2^k$ data blocks.

Ensure: best candidate $\mathcal{S}'$.

1: **for** $k = 0, 1, \cdots, \lceil \log_2 n \rceil - 1$ **do**
2: use additional 2^k data blocks to update the field-wise LR models of all $\mathcal{S} \in \mathbb{S}$, with warm-starting;
3: evaluate the models of all $\mathcal{S}$'s with validation AUC;
4: keep the top half of candidates in $\mathbb{S}$: $\mathbb{S} \leftarrow$ `top_half`($\mathbb{S}$) (rounding down);
5: break if $\mathbb{S}$ contains only one element;
6: **end for**
7: **return** $\mathcal{S}'$ (the singleton element of $\mathbb{S}$).

预处理

在数据预处理处，我们使用离散化的策略对数据进行预处理方便后续的特征交叉。为了使离散化过程自动化，避免对专家的依赖，提出了一种多粒度离散化方法，详细地可以参考下图：

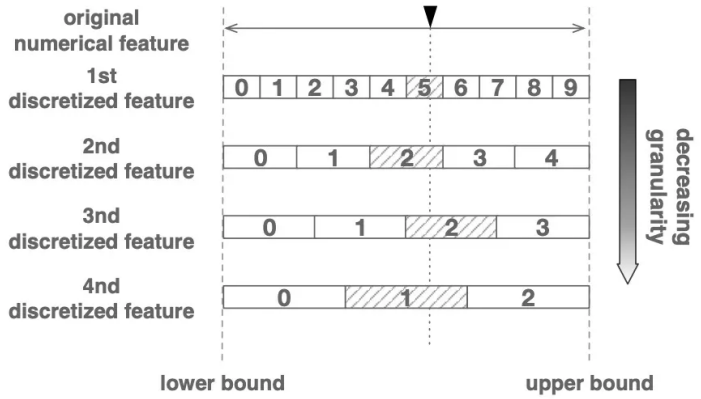

Figure 5: An illustration of multi-granularity discretization. Shade indicates the value taken by each discretized feature.

终止

AutoCross使用了三种终止条件：

- 运行时条件：用户可以设置AutoCross的最大运行时间。当时间流逝时，AutoCross终止输出当前解决方案。另外，用户可以随时中断该过程并得到时间的结果；
- 性能条件：在生成新的特征集后，用LR模型的所有特征进行训练。如果与前一组相比，验证性能下降，则终止验证过程；
- 最大特征数：用户可以给出一个最大交叉特征数，当达到该数目时自动交叉停止；

实验

1. 效果比较

Table 3: Experimental results (test AUC) on benchmark and real-world business datasets.

Benchmark Datasets					
Method	Bank	Adult	Credit	Employee	Criteo
LR (base)	0.9400	0.9169	0.8292	0.8655	0.7855
AC+LR	**0.9455**	**0.9280**	**0.8567**	**0.8942**	0.8034
AC+W&D	0.9420	**0.9260**	**0.8623**	**0.9033**	**0.8068**
CMI+LR	**0.9431**	0.9153	0.8336	0.8901	0.7844
Deep	0.9418	0.9130	0.8369	0.8745	0.7985
xDeepFM	0.9419	0.9131	0.8358	0.8746	**0.8059**
Real-World Business Datasets					
Method	Data1	Data2	Data3	Data4	Data5
LR (base)	0.8368	0.8356	0.6960	0.6117	0.5992
AC+LR	**0.8545**	**0.8536**	**0.7065**	**0.6276**	0.6393
AC+W&D	**0.8531**	**0.8552**	**0.7026**	**0.6260**	**0.6547**
Deep	0.8479	0.8463	0.6936	0.6207	0.6509
xDeepFM	0.8504	0.8515	0.6936	0.6241	**0.6514**

- AC+LR和AC+W&D都比LR（base）有显著的改善，AC+W&D也显著提高了deep模型的性能。

- 这些结果表明，通过生成交叉特征，AutoCross可以使数据更具信息性和区分性，并提高学习性能。AutoCross取得的有希望的结果也证明了filed-wise LR识别有用交叉特征的能力

2. 高阶特征的影响

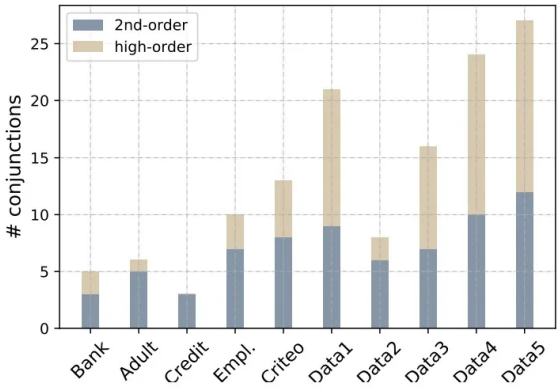

Figure 6: The number of second/high-order cross features generated for each dataset.

Table 5: Test AUC improvement: second v.s. high order features on benchmark datasets.

v.s. LR(base)	Bank	Adult	Credit	Employee	Criteo	Average
CMI+LR	0.330%	-0.175%	0.531%	2.842%	-0.140%	0.678%
AC+LR	0.585%	1.211%	3.316%	3.316%	2.279%	2.141%

- 高阶特征能为模型带来非常不错的提升。

3. 模型的速度和预测时间

Table 6: Cross feature generation time (unit: hour).

Benchmark Datasets				
Bank	Adult	Credit	Employee	Criteo
0.0267	0.0357	0.3144	0.0507	3.0817
Real-World Business Datasets				
Data1	Data2	Data3	Data4	Data5
0.9327	0.7973	1.5206	2.7572	5.1861

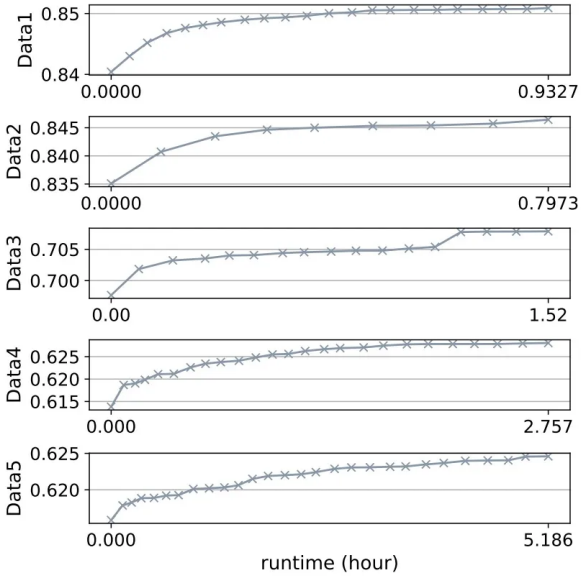

Figure 7: Validation AUC curves in real-business datasets.

- 模型的特征交叉速度很快;
- 模型的预测速度相较于神经网络也快了很多;

结论

本文介绍了AutoCross，一种在实际应用中用于表格数据的自动特征交叉方法。它捕捉分类特征之间的有用交互作用，并提高学习算法的预测能力。它利用beam-search来高效地构造交叉特征，从而可以考虑到高阶特征交叉，而这是现有研究中尚未涉及到的。此外，本文提出了连续的小批量梯度下降和多粒度离散化方法，在保持高度简单性的前提下，进一步提高了效率和效率。所有的算法都是为分布式计算而设计的，用于处理现实世界中的大数据。实验结果表明，AutoCross可以显著增强表格数据的学习能力，优于其他基于搜索和基于深度学习的针对同一主题的特征生成方法。

参考文献

1. AutoCross: Automatic Feature Crossing for Tabular Data in Real-World Applications: https://arxiv.org/pdf/1904.12857.pdf