

# 基于强化学习的特征工程算法研究\*

谢斌<sup>1,2</sup>,林珊玲<sup>2,3</sup>,林志贤<sup>1,2</sup>,郭太良<sup>1,2</sup>

(1.福州大学 物理与信息工程学院,福建 福州 350116;

2.中国福建光电信息科学与技术创新实验室,福建 福州 350116;3.福州大学 先进制造学院,福建 泉州 362200)

**摘要:**特征工程可以自动地处理和生成那些判别性高的特征,而无需人为的操作。特征工程在机器学习中是不可避免的一环,也是至关重要的一环。提出一种基于强化学习(RL)的方法,将特征工程作为一个马尔可夫决策过程(MDP),在上限置信区间算法(UCT)的基础上提出一个近似的方法求解二分类数值数据的特征工程问题,来自动获得最佳的变换策略。在5个公开的数据集上验证所提出方法的有效性,FScore平均提高了9.032%,同时与其他用有限元变换进行特征工程的方法进行比较。该方法确实可以得到判别性高的特征,提高模型的学习能力,得到更高的精度。

**关键词:**特征工程;强化学习;机器学习

中图分类号: TP391

文献标识码: A

DOI:10.16157/j.issn.0258-7998.201173

中文引用格式: 谢斌,林珊玲,林志贤,等. 基于强化学习的特征工程算法研究[J]. 电子技术应用,2021,47(7):29–32,43.

英文引用格式: Xie Bin,Lin Shanling,Lin Zhixian,et al. Research on feature engineering algorithm based on reinforcement learning[J]. Application of Electronic Technique,2021,47(7):29–32,43.

## Research on feature engineering algorithm based on reinforcement learning

Xie Bin<sup>1,2</sup>,Lin Shanling<sup>2,3</sup>,Lin Zhixian<sup>1,2</sup>,Guo Tailiang<sup>1,2</sup>

(1.School of Physics and Information Engineering,Fuzhou University,Fuzhou 350116,China;

2.China Fujian Optoelectronic Information Science and Technology Innovation Laboratory,Fuzhou 350116,China;

3.School of Advanced Manufacturing, Fuzhou University, Quanzhou 362200, China)

**Abstract:** Feature engineering can automatically process and generate those highly discriminative features without human operation. Feature engineering is an inevitable and crucial part of machine learning. The article proposes a method based on reinforcement learning(RL), taking feature engineering as a Markov decision process(MDP), and proposes an approximate method based on the upper limit confidence interval algorithm(UCT) to solve the feature engineering of binary numerical data problem to automatically obtain the best transformation strategy. The effectiveness of the proposed method is verified on five public data sets. The FScore of the five public data sets is improved by an average of 9.032%. It is also compared with other papers that use finite element transformation for feature engineering. This method can indeed obtain highly discriminative features, improve the learning ability of the model, and obtain higher accuracy.

**Key words:** feature engineering; reinforcement learning; machine learning

## 0 引言

机器学习广泛应用于人们的日常生活中,其中预测分析广泛应用于多个领域的决策,包括欺诈检测<sup>[1-2]</sup>、在线广告<sup>[3-4]</sup>、风险管理、市场营销等。预测模型是采用监督学习算法来进行预测,通过历史数据进行训练分类或者回归模型来预测未知的结果,以起到决策的作用。数据的表示方法对于模型的准确度十分重要,原始的数据空间往往难以表达数据。因此,在模型构建之前对数据

进行适当的处理及转换是必不可少的。

特征工程的主要目的就是改变预测建模的特征以更好地适应算法的训练,通过生成那些判别性高的特征来提高模型训练的准确度。在现实中,特征工程是由数据科学家手动和根据领域知识来进行的,这一过程往往是十分繁琐且耗时的<sup>[5]</sup>,而且很容易产生错误和偏差。

本文提出一种基于强化学习(RL)的方法进行特征工程,通过强化学习的相关方法从原始的数据中学习有

\* 基金项目:国家重点研发计划资助项目(2016YFB0401503);广东省科技重大专项资助项目(2016B090906001);福建省自然科学基金资助项目(2020J01468)

效的策略。探索有效策略的过程无需人为的操作,通过算法自动学习得到。

## 1 相关工作

现有的特征工程的方法分为两种,一种使用数值变换方法进行特征工程。数据科学机(DSM)<sup>[6]</sup>提出和开发了深度特征综合算法,自动地为那些有关系的数据集生成特征;ExploreKit<sup>[7]</sup>通过原始特征中的信息来组合生成大量的候选特征,并提出了一种新颖的基于机器学习的特征选择方法;NARGESIAN U<sup>[8]</sup>提出训练一组神经网络,来预测对分类性能产生积极影响的转换。KHURANA U<sup>[9]</sup>通过对过去的示例进行强化学习,得出一种高效的探索策略;Cognit<sup>[10]</sup>系统以分层且非穷举的方式探索各种特征构造选择,通过贪婪的探索策略最大化模型的准确性;AutoCross<sup>[11]</sup>是用在表数据上的数据特征组合的方法,利用集束搜索策略构建有效的特征组合。

另一种是通过深度学习的方法来学习有用的特征。深度学习的方法在视频、图像以及语音等数据上取得巨大的成功,但是深度学习的方法往往需要大量的数据以避免过拟合,且其学到的特征往往不可解释。

## 2 强化学习进行特征工程

本文的第一个贡献是将特征工程作为一个马尔可夫决策过程(MDP),令 $D_0$ 代表原始的特征空间,则状态空间 $S$ 就是经过一系列数字变换后形成的特征空间的集合。动作空间 $A$ 是一系列的数值变换方法(如 $\sin$ 、 $\cos$ 、 $\log$ 等)的集合。当前特征空间 $D$ 经过特征变换 $a$ 后,新的特征空间 $D^*=a(D), a \in A$ 。

特征工程的主要目的是对原始数据进行一系列处理以获得最高的精度,所以MDP过程中最终状态的奖励就是最终的特征空间与原始的特征空间的精度的差值,用 $\text{Acc}(D)$ 来表示,令 $\Pi^*$ 作为在每一个状态选择动作的MDP策略, $D_{II}$ 代表从原始特征空间跟随 $\Pi^*$ 策略生成最终的特征空间。因此, $\Pi^*$ 的作用就是使得 $\text{Acc}(D_{II})$ 最大化,即在最终的状态空间获得最大的精度。

$$\Pi^* = \operatorname{argmax} \text{Acc}(D_{II}) \quad (1)$$

根据Bellman的最优化原理<sup>[12]</sup>,将最优值函数 $V^*$ 定义为:

$$V^*(D) = \begin{cases} \text{Acc}(D) & D \text{ 为最终状态} \\ \max_{a \in A} V^*(a(D)) & \text{其他} \end{cases} \quad (2)$$

所以最佳的策略 $\Pi^*(D)$ 可以定义为:

$$\Pi^*(D) = \operatorname{argmax}_{a \in A} V^*(a(D)) \quad (3)$$

虽然上述的 $\Pi^*$ 可以得到最优的特征空间,但是状态空间 $S$ 的数量是指数级别的,处理起来将会十分困难。因此,本文第二个贡献是在上限置信区间算法(UCT)<sup>[13]</sup>的基础上提出一个近似的方法,使其可以用于进行二分类数值数据的特征工程,进而获得最优策略的近似值。

## 3 上限置信区间算法(UCT)

UCT是一种蒙特卡洛树搜索算法,可以权衡勘探与利用,以探索最有希望的区域。算法分为4个步骤:

(1)选择:从根节点开始,使用树策略即UCB公式来挑选一个子节点,直到挑选到叶子节点。

(2)扩展:对选定的叶子节点,将叶子节点采取非试探性的动作可以达到的节点加入该叶子节点的子节点中。

(3)模拟:从当前选择的节点,根据预演策略来进行动作的选择直到模拟结束,得到的是一次蒙特卡洛实验并获得奖励值。

(4)回溯:用模拟过程得到的奖励值来更新当前节点及其所有的父亲节点的状态。

树策略(UCB)<sup>[14]</sup>公式为:

$$a^* = \operatorname{argmax}_{a \in A} \{ u_{F,a} + \sqrt{\frac{C_e \ln(T_F)}{t_{F,a}}} \} \quad (4)$$

其中, $T_F$ 是节点 $F$ 的访问次数, $A$ 是可供选择的动作空间, $u_{F,a}$ 表示节点 $F$ 选择动作 $a$ 时获得的平均奖励, $t_{F,a}$ 表示节点 $F$ 选择动作 $a$ 的次数。参数 $C_e$ 可用来控制勘探强度。

## 4 基于UCT的特征工程

本文将特征工程视作蒙特卡洛树搜索过程,图1为整体的结构示意图,图中只画出了3层的结构图,动作空间为 $\log$ 、 $\cos$ 、 $\sqrt{r}$ 。图中椭圆代表一个节点,每一个节点代表着不同版本的数据。节点与节点之间的连线代表选择的动作,状态空间就是这些节点的集合。其中 $D_0$ 是原始的特征空间也就是根节点,从根节点出发,每进行一次特征变换就会得到一个子节点。这样随着层数的不断增加,探索空间也呈现爆炸性增长。因此,遍历所有的节点是非常昂贵且不现实的,因此本文在UCT的基础上提出求解最佳策略的方法,以此来权衡探索与利用。

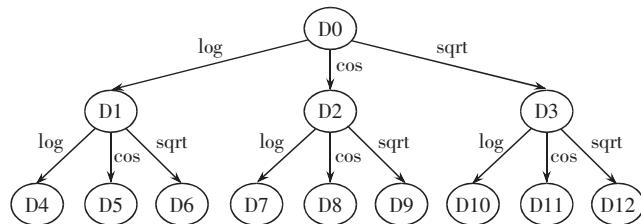


图1 整体结构示意图

本文在UCT基础上提出一种近似方法,使其可以适用于特征工程。在UCT中是根据平均收益来选择子节点。这在解决特征工程的问题并不是很适用,平均收益高的节点可以说明当前特征空间的特征是相对较好的,可能对于某些特征只有经过特定的变换才能获得较好的精度,但是在探索的过程中由于进行其他一些降低精度的变换,拉低了这个节点的奖励值,从而导致最终没有选择这个节点。因此,本文提出方法中节点中不仅维护了平均奖励、访问的次数,还维护了模拟过程中得到

的最大的收益。在选择子节点时,根据子节点的最大收益值来进行选择。本文设定的实验结束的条件是达到最大的层数或者选择了停止动作。在达到最大的层数之前可能就已经获得了最佳的结果,停止动作的设立是为了避免这种情况发生。

接下来的部分是对本文在 UCT 算法基础上提出的近似方法的介绍以及奖励函数、特征变换、模拟策略的描述。

#### 4.1 奖励函数的设定

蒙特卡洛搜索要求每次迭代都要计算奖励值。奖励函数的主要目的就是判断变换的有效性,因此将最终的特征空间与原始特征空间的精度的差值作为奖励。设  $A_c$  为特征空间的精度,本文计算的精度为 FScore。 $D$  为最终的特征空间, $D_0$  为原始的特征空间,则奖励  $R$  可以表示为:

$$R = A_c(D) - A_c(D_0) \quad (5)$$

#### 4.2 特征变换的设定(transform)

在进行特征变换时,设  $A$  为动作空间, $a \in A$  为当前选择的动作, $D^i$  为当前的特征空间,则新的特征空间  $D^*$  可以表示为:

$$D^* = D^i \cup a_{a \in A}(D^i) \quad (6)$$

特征变换都是对适合于当前选择的动作的特征进行变换。

在每次特征变换后通过特征筛选将特征的数量与原始特征空间保持一致,这样既可以使用判别性高的特征来创建高阶特征,又可以有效避免特征爆炸的问题。

#### 4.3 模拟策略(evaluate)

本文设定的模拟策略为贪婪的策略,设  $A$  为动作空间, $D^*$  为当前的特征空间。模拟策略  $\Pi^*$  可以表示为:

$$\Pi^*(D^*) = \underset{a \in A}{\operatorname{argmax}} A_c(D^* \cup a_{a \in A}(D^i)) \quad (7)$$

在 UCT 算法中模拟过程的节点往往是不会保存下来的,但是本文最终是根据子节点的最大收益来选择子节点。为了确保可以找到模拟过程中获得最大收益的路径,因此将模拟过程达到的节点全都保存下来。由于本文设定的是贪婪的模拟策略,需要昂贵计算代价,因此将模拟过程的节点保存下来在一定程度上会加快整体的计算速度。

##### (1) 算法 1 选择动作(get\_act)

输入 :  $(D_0, \text{Node}_0, K, B_{\max})$

输出 : action

- 1) for  $k \leftarrow 1$  to  $K$  do
- 2)  $D_* = D_0, \text{Node} = \text{Node}_0, \text{deep} = 0$
- 3) while true
- 4) if Node.is\_leaf() // 判断是否是叶子节点
- 5) break
- 6)  $a^* = \underset{a \in A}{\operatorname{argmax}} \{ u_{F,a} + \sqrt{\frac{C_e \ln(T_F)}{t_{F,a}}} \}$

//根据 UCB 挑选一个 action

- 7) Node = select( $a^*$ ) //选择
- 8) deep += 1 //当前层数加 1
- 9)  $D_* = \text{transform}(D_*, a^*)$  //特征变换
- 10) if deep <  $B_{\max}$  //判断是否达到最大层数
- 11) Node.expend() //扩展
- 12) leaf\_v = evaluate( $D_*$ ) //模拟
- 13) Node.update(leaf\_v) //回溯
- 14) if end
- 15) break
- 16) action = argmax(Node<sub>0</sub>.children.best\_score) //根据最大收益选择动作

##### (2) 算法 2 本文近似算法

输入 :  $(D_0, \text{Node}_0, K, B_{\max}, A)$

输出 : score

- 1)  $D_* = D_0, \text{Node} = \text{Node}_0$
- 2) for  $i \leftarrow 1$  to  $B_{\max}$  do
- 3)  $a^* = \text{get\_act}(D_*, \text{Node}, K, B_{\max}, i)$
- 4) if ( $a^* == \text{stop}$ )
- 5) break
- 6)  $D_* = \text{transform}(D_*, a^*)$
- 7) Node = select( $a^*$ )
- 8) score = get\_score( $D_*$ )

#### 4.4 方法概述

算法 1 展示了在每一层选择子节点的算法。通过 UCB 公式来权衡探索与利用,以此来搜索最有希望的区域,同时在这过程中保存获得最大收益。最终通过最大的收益来选择子节点。这里判断循环结束的条件就是连续超过 10 次选择相同的子节点或者达到最大的迭代次数。算法 2 展示了本文提出的近似算法。在每一层通过算法 1 来选择子节点,直到到达最终层或者挑选到停止动作。

#### 5 实验和结果

本实验在 5 个公开数据集上进行,验证本文提出方法的有效性。表 1 列出了本文使用的 5 个二分类数据集的名称、来源、记录数和特征的数量。

表 1 本文使用 5 个公开数据集的来源、记录数以及特征数量

数据集	来源	记录数	特征数量
Pima Indian	UCIrvine	768	8
German Credit	UCIrvine	1 000	30
SpectF	UCIrvine	266	44
SVMguide3	LibSVM	1 243	21
Messidor_features	UCIrvine	1 151	19

所有的实验结果都是在 5 折交叉验证下得到的,本文使用的模型是随机森林,模型的参数全部都是默认的参数。对于所有的数据集  $c_e$  全部设为 1.0。蒙特卡洛搜

索树的最大层数设置为 10 层,动作空间有:对数、开三次方、平方、开平方、标准化、归一化、 $S$  函数、正弦、余弦、双曲函数、分桶、相加、相乘、停止 14 种变换方法。在这些变换方法中相加和相乘的操作是对特征空间中的两两特征进行相加与相乘,停止操作就是不进行任何操作,返回原来的特征空间。

在实验中,比较了 5 个方面的精度:(1)原始数据集;(2)本文的方法;(3)将所有特征变换方法都用在原始的特征空间后进行特征筛选;(4)随机选择动作进行特征变换,重复进行 10 次;(5)选择使当前特征空间获得最高精度的特征变换,重复进行 10 次。比较结果如表 2 所示。

表 2 各个数据集在五个方面的精度(FScore)比较

数据集	(1)	(2)	(3)	(4)	(5)
Pima Indian Diabets	0.687 5	0.747 1	0.714 3	0.698 1	0.734 2
German Credit	0.675 4	0.711 3	0.682 4	0.657 8	0.692 7
SpectF	0.650 1	0.827 6	0.627 9	0.715 7	0.759 7
SVMGuide3	0.702 3	0.795 5	0.734 9	0.76	0.770 2
Messidor_features	0.678 1	0.763 5	0.675 6	0.654	0.678 1

本文还与其他两篇有限元变换的特征工程文献<sup>[9-10]</sup>进行比较。这两篇文献和本文类似,都是使用数值变换来进行特征工程。文献[9]通过线性逼近的 Q\_learning<sup>[15]</sup>来从过去的示例来学习策略,而文献[10]则是使用制定好的策略。本文通过 UCT 框架探索最佳的策略。本文使用 FScore 提升量来进行比较,表 3 中精度提升量以文献[9]为基准。

表 3 本文方法与文献[9]、[10]FScore 提升量的比较

数据集	本文	文献[9]	文献[10]
Pima Indian Diabets	5.96	3.5	1.1
German Credit	3.59	6.3	0.1
SpectF	17.75	10.2	9.4
SVMGuide3	9.32	3.6	2.6
Messidor_features	8.54	6.1	7.1

从表 2 可以看出,本文提出的方法在 5 个数据集原本的 FScore 上平均提高了 9.032%,在与其他变换方式进行比较时也是最好的。从表 3 可以看出,本文提出的方法在除了 German Credit 的数据集外,提升的精度是最高的。图 2 展示了 5 个数据集的 FScore 与变换次数之间的关系,图中有些数据集变换不足 10 次的原因是选择停止动作导致实验提前结束。对于搜索到最大层数的数据集的曲线走向整体上是朝上的,即使中间有较大的起伏,但是在最终层总能获得最高的精度。随着最大层数的加深,精度可能会进一步提升。最终层的特征的数量与原始特征空间的特征数量保持一致,在相同的特征数量下精度却得到了大幅度提升,说明本文提出的方法可以得到判别性高的特征,提高模型的学习能力,又能避

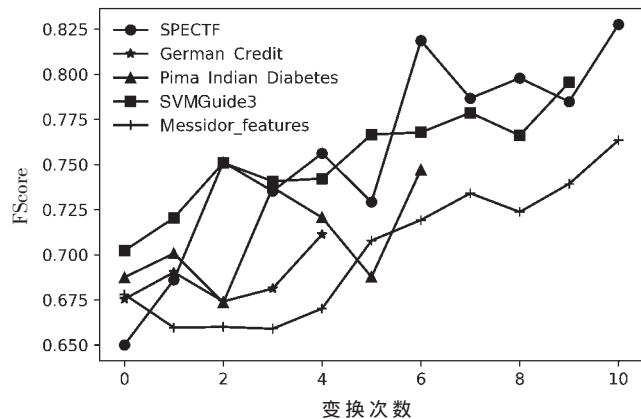


图 2 精度(FScore)与变换次数之间的关系

免特征爆炸的问题,达到特征工程的目的。

## 6 结论

本文中提出以一个新颖的思路将特征工程视作马尔可夫决策过程(MDP),在 UCT 基础上提出一种特征工程的最佳策略的求解方法。本文提出的方法可以自动找到那些具有高判别性的特征,同时又避免了特征爆炸的问题,在 5 个数据集上均取得了较好的效果。本文提出的方法目前只实现了二分类的数值数据的特征工程,还未支持多分类以及回归数据。由于采用的是贪婪模拟策略,所需要的计算代价比较昂贵,也有可能陷入局部最优的情况。未来可以通过深度蒙特卡洛的方法作为特征工程求解的方法,将深度学习的方法与蒙特卡洛的方法相结合来实现特征工程。

## 参考文献

- [1] BOLTON R J, HAND D J. Statistical fraud detection : a review[J]. Statistical Science , 2002 , 17(3) : 235–255.
- [2] WANG S. A comprehensive survey of data mining-based accounting-fraud detection research[C]. Intelligent Computation Technology and Automation. IEEE , 2010 : 50–53.
- [3] EVANS D S. The online advertising industry : economics, evolution, and privacy[J]. Journal of Economic Perspectives , 2009 , 23(3) : 37–60.
- [4] ZEFF R L, ARONSON B. Advertising on the internet[M]. Advertising on the Internet. John Wiley & Sons , Inc. , 1999.
- [5] WANG R , FU B , FU G , et al. Deep & cross network for ad click predictions[C]. Proceedings of the ADKDD'17. ACM , 2017.
- [6] KANTER J M, VEERAMACHANENI K. Deep feature synthesis: towards automating data science endeavors[C]. 2015 IEEE International Conference on Data Science and Advanced Analytics(DSAA). IEEE , 2015.
- [7] KATZ, G , CHUL, E , SHIN R , et al. Explorekit : automatic feature generation and selection[C]. IEEE International Conference on Data Mining. IEEE , 2017 : 979–984.

(下转第 43 页)



## 版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所