面向机器学习的自动化特征工程研究综述

刘玉琳 白 杨* 崔 斌* 黎 洋
(北京大学信息科学技术学院 北京 100871)

摘 要 自动化特征工程是一种无须人为干预且高效的特征工程解决方案,是当前机器学习研究领域的热点问题之一。为了更好地帮助用户使用自动化特征工程方法构建高质量特征,在特征生成、特征选择、系统实现三个方面对当前自动化特征工程方法的技术特点以及存在的不足进行总结分析与归纳,并在已有分析的基础上,指出自动化特征工程的未来研究方向以及面临的问题和挑战。

关键词 自动化特征工程 特征生成 特征选择 系统实现 机器学习

中图分类号 TP3 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2025.01.001

A SURVEY ON AUTOMATED FEATURE ENGINEERING FOR MACHINE LEARNING

Liu Yulin Bai Yang* Cui Bin* Li Yang
(*School of Electronics Engineering and Computer Science*, *Peking University*, *Beijing 100871*, *China*)

Abstract Automated feature engineering is an efficient feature engineering solution that does not require human intervention, and is a popular research topic in machine learning research. In order to help the users employ automated feature engineering algorithms to create high-quality feature sets, this paper discussed the technics of current automated feature engineering frameworks from feature generation, feature selection, and system implementation, summarized, analyzed, and summarized the shortcomings of the existing methods. On the basis of existing analysis, we pointed out the future directions and challenges of automated feature engineering.

Keywords Automated feature engineering Feature generation Feature selection System implementation Machine learning

0 引 言

近年来,随着机器学习在各个领域的广泛应用,人们发现通过特征工程生成高质量的特征能够有效提升机器学习模型的性能[1]。然而,目前特征工程主要通过数据科学家手动生成并选择高质量特征,这个过程需要大量的重复迭代,会耗费较多的人力和时间成本[2]。与此同时,通过特征工程生成高质量的特征还需要特定业务领域的专业知识,这使得在某个领域内生成高质量特征的方法很难复用到其他领域[3]。此外,人工进行特征工程时,能够检索到的特征生成和特征选择的空间有限[4],会导致构建的特征集合不是全局最优的。因此以实现在不同领域的数据上都能够自动且高效地构建高质量特征集为目标的自动化特征工程研究成为当前特征工程研究的热点问题之一[5]。

自动化特征工程是指在有限的计算资源下,基于已有的特征集合,通过算法自动分析已有特征并构造新的特征,使得机器学习模型在新特征集合上具有最优的表现[6]。现有的自动特征工程方法包括:FeatureTools[7]、AutoCross[8]、AutoLearn[9]、Cognito[10]、Explorekit[11]、SAFE[12]、RLFE[13]、RNNFE[14]、LFE(Learning Feature Engineering)[15]、CNN[16]和 Autoencoder[17]等。这些方法采用了不同的特征生成、特征选择和系统实现方案。为了分析这些方法的优缺点,以便用户使用自动化特征工程方法生成最优的特征,本文对现有的自动化特征工程方法进行系统的分析,总结现有方法的技术特

收稿日期:2021-08-03。国家重点研发计划项目(2018YFB1004403);国家自然科学基金项目(61832001)。**刘玉琳**,硕士生,主研领域:机器学习,数据科学。**白杨**,博士。**崔斌**,教授。**黎洋**,博士生。

点,分析存在的不足,并指出自动化特征工程未来研究存在的问题和挑战。

1 自动化特征工程基本概念

自动化特征工程主要是基于已有特征集合,通过特征生成和特征选择自动构建一组新的特征集合,使得特定的机器学习算法性能在新特征集合上达到最优。

1.1 问题定义

常见的机器学习任务由三部分组成:(1) 数据集 $D=[X;Y]$,X 为特征集合,Y 为标签集;(2) 机器学习模型算法 A;(3) 模型性能的评估函数 L 和得分 f_{score}[18]。$X=\{f_1,f_2,\cdots,f_n\}$,表明 X 由 n 个特征组成。自动特征工程是在有限的计算资源下,以现有特征集合 X 为基础通过特征生成和特征选择自动构建一组新的特征集合 $\boldsymbol{X}'=\{f'_1,f'_2,\cdots,f'_m\}$,用于使算法 A 的性能评估得分 $f_{\text{score}}=\max(L(A,X',Y))$ 比算法 A 在原特征集合 X 上的性能评估得分 $f_{\text{score}}=\max(L(A,X,Y))$ 高,且达到最高。

1.2 特征生成

特征生成是特征工程中最重要的一个功能,用于根据原始特征生成新的特征,如式(1)所示。

$$f'=F(f) \tag{1}$$

式中:f 为原始特征,f′为生成的新特征,F 为生成新特征的方法。生成的特征可分为三类:(1) 能够提升模型性能且不冗余的特征;(2) 冗余特征;(3) 无法提升模型性能的特征。以下将第一类称为有效特征,将第三类称为无效特征[19]。

目前常用的特征生成方法主要包括基于函数映射的方法[11]、基于交叉运算的方法[8]、DFS(Deep Feature Synthesis)[7] 方法、基于特征相关性的方法[9]、基于 CNN 的方法[16]、基于 Autoencoder 的方法[17]。

1.2.1 基于函数映射的方法

基于函数映射的特征生成方法是指将原始特征通过函数映射到新的维度上进而生成新的特征。常用的函数有一元函数[11](如取平方、开方等)、二元函数[15](如求和、差、积、商、经纬度距离等)、多元函数[20](如主成分分析 PCA、独立成分分析 ICA、奇异值分解方法和神经网络压缩算法等)。

1.2.2 基于交叉运算的方法

基于交叉运算的特征生成方法是指通过对两个及以上特征进行交叉运算进而生成新的特征的方法[21]。例如,对类别特征的所有值进行交叉相乘。如特征 A 有两个类别 $\{A_1,A_2\}$,特征 B 有两个类别 $\{B_1,B_2\}$,A 和 B 之间进行交叉运算,新生成的特征为 $\{(A_1,B_1),(A_1,B_2),(A_2,B_1),(A_2,B_2)\}$。

1.2.3 DFS 方法

在工业界,数据是存储在多张实体表中的,比如用户的购物信息是存储在三张相关的表中的,包括用户画像表、物品信息表和购物流水表。数据科学家进行特征生成时是将三张实体表的数据进行关联生成新特征[22]。该过程可以通过 DFS[7] 方法自动实现。该方法基于表与表之间的关联关系生成特征,例如用户实体表与购物流水实体表的关系为一个用户可以有多个购物流水记录,基于这个关联关系,DFS 可以通过多张实体表生成用户购物流水表中所购物品价格的最大值、最小值和均值等新特征[23]。

1.2.4 基于特征相关性的方法

基于特征相关性的特征生成方法[9]是以两个特征组合为基础,通过计算以其中一个特征的取值来预测另一个特征的取值[24],该预测值与特征真实值的偏差可作为新生成的特征。

对于不同的类别,特征之间的关系往往是不一样的[25]。例如,在信用卡逾期数据中,对月收入与大额商品购物频次两个特征进行关联分析,在非逾期用户中,两者呈现正相关关系,但在逾期用户中并不呈现正相关关系。当以月收入特征预测大额商品购物频次特征时,非逾期用户有较好的预测效果,预测值相比真实值偏差较小,反之逾期用户的预测值相比真实值则偏差较大,因此以月收入特征预测大额商品购物频次的预测值与真实值的偏差可以作为新生成的特征用于预测用户信用卡的逾期风险。

1.2.5 基于卷积神经网络的方法

卷积神经网络(CNN)是面向图像数据分析的一种神经网络结构,其通过卷积核(例如 1×1、3×3、11×11 等)提取图像的局部细节信息,进而生成特征子图,并将生成的特征子图作为下一层神经网络的输入。基于 CNN 的特征生成过程是通过设置“填充”和“步长”来实现的,并通过池化层来决定是否将生成的特征输入到下一层。“填充”用于拓展输入图像,增加卷积核对图像边缘特征的提取能力。“步长”用于设置卷积核的移动间距,步长设置越小,对局部细节特征的识别越灵敏。池化层以上一层卷积核生成的特征子图作为输入,将其看作是不同特征单元的组合(例如,3×3 卷积核可看成 9 个特征单元的组合),池化层将

该特征单元组合与选择矩阵作乘积来决定是否使用对应的特征单元,常用的选择矩阵有取最大值、取平均值等。如文献[16]中通过训练 CNN 网络模型实现了一种提取青光眼特征的方法。该模型的前 5 层为 CNN 结构,通过模型训练得出前 5 层的参数,此时该模型将输入图像转化为生成青光眼局部细节的新特征,实现了自动生成青光眼特征的功能。又如文献[26]中训练了一个 21 层的 CNN 网络用于识别电脑病毒。该网络将病毒数据建模成灰度图像[27],将其输入到 CNN 网络中,以最后一层全连接层的输出作为病毒分类器的输入进行训练,进而实现电脑病毒的识别。完成训练时,CNN 网络对原始病毒数据生成了大量的病毒局部特征。此时 CNN 网络从输入层到全连接层之间的结构可独立作为一种自动化的特征生成的方法。

1.2.6　基于自动编码器的方法

自动编码器(Autoencoder)是一种无监督模型,其结构如图 1 所示。

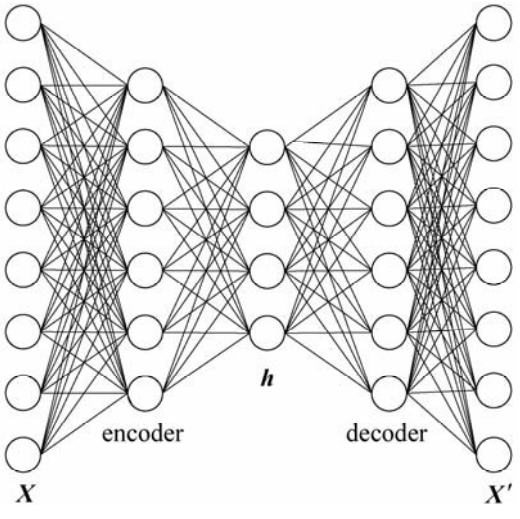


图 1　Autoencoder 结构

输入数据 X 经过编码层(encoder)对原始数据进行压缩。编码层的输出为隐藏层 h,h 同时作为解码层(decoder)的输入。解码层对 h 进行还原,输出结果为 X'。训练自动编码器使得 $X \approx X'$,当训练完成时,编码层将输入数据 X 压缩为 h,h 可视为自动编码器对 X 生成的特征,这种方法类似于通过主成分分析(PCA)实现降维功能。此时的编码层可视为一种特征生成方法。如文献[17]中提出了一种基于自动编码器提升机器学习模型性能的方法。机器学习模型在利用自动编码器生成的新特征后,模型性能得到了提升。又如文献[28]中自动编码器使用步态数据生成性特征,在使用新特征后,异常步态识别准确率同样得到了提升。

1.3　特征选择

特征生成阶段会生成大量的特征,包括有效特征、冗余特征和无效特征。从新生成的特征中选出有效特征,排除冗余和无效特征的过程称为特征选择。

常用的特征选择方法主要包括:封装型(Wrapper)、过滤型(Filter)、嵌入型(Embedded)、混合型[29]、基于梯度的方法[30]、基于迁移学习[11]的方法等。

1.3.1　封装、过滤、嵌入、混合型方法

封装型方法[31]是指以机器学习算法的性能为特征选择的评价标准,选择那些能够提升算法性能的特征。这种方法能够有效地选择最优的特征,但算法的多次训练会消耗大量的计算资源。

过滤型方法是用特定的评估指标来过滤掉评估得分低的特征的方法。可使用的评估指标有信息熵、信息增益等。以信息熵为评估指标为例,通过计算每一个特征的信息熵,过滤掉信息熵相对较小的特征,保留信息熵较大的特征[32]。过滤型方法只需要较少的计算资源就能实现特征选择,但经过选择后的特征对下游机器学习算法性能的提升不可控。

嵌入型方法是将特征选择与机器学习训练过程相结合,即在机器学习训练过程中自动进行特征选择。在原始特征集上该方法首先使用机器学习算法进行训练,然后计算每个特征的权重系数,根据权重系数从大到小进行特征选择。特征的权值系数代表了该特征对于算法的贡献度。例如在决策树模型中,可根据特征的重要性(Feature importance)进行特征选择。我们可以基于特征重要性的评估,删除特征重要性得分低的特征,保留得分高的特征。相对于过滤型法,嵌入型方法与模型效果相结合,能够有效提升模型的性能。

混合型方法(Hybrid 方法[29])是结合封装型和过滤型的方法。先基于过滤型方法对特征进行筛选,降低特征的维度,之后使用封装型方法进一步对特征进行高质量的筛选。该方法相比封装型方法资源损耗较小,但性能较差。相比过滤型方法性能较优,但资源消耗较大。

1.3.2　基于梯度的方法

基于梯度的特征选择方法[30]是使用梯度信息进行选择特征。该方法将离散的特征选择空间映射到连续的空间 X_f 上,并在 X_f 上随机选择一个点 x_f($x_f \in X_f$),x_f 的取值即代表了一种特征选择方案,之后计算该方案对模型性能的提升量 $y_e = F_e(x_f)$,其中,F_e 为文献[33]提出的一种评估模型性能提升量的函数,通过求 y_e 在 x_f 上的梯度来寻找 x_f 的更新方向,即特征选择的方向[34],进而实现特征选择。通过该方法选择

的特征,对于下游机器学习算法性能的提升优于过滤型方法,但需要消耗大量的计算资源。

1.3.3 基于迁移学习的方法

基于迁移学习的方法[11]是通过之前大量的机器学习任务提炼出的在不同数据集上能够切实提高模型性能的特征选择方法,并将其作为先验知识迁移到新的特征选择任务上,用于在新任务上实现特征选择。该方法避免了大量且无效的模型训练,能够加速特征选择过程。

2 自动化特征工程研究进展

现有的自动化特征工程方法通过不同的系统实现完成了自动化的特征生成和特征选择,本节对现有方法的系统实现进行分析并归类总结。

现有的自动特征工程的系统实现主要包括九类:只有特征生成(Expand)、先特征生成后特征选择(Expand-Reduce)、基于层次贪婪搜索(Hiearchrial Greedy Search)、基于迁移学习(Transfer Learning)、基于元学习(Meta-learning)[35]、基于循环神经网络(Recurrent Neural Network)、基于强化学习(Reinforcement Learning)的方法、面向图像数据的基于卷积神经网络(CNN)和基于自动编码的方法,具体如表 1 所示。

表 1 现有自动化特征工程方法及其系统实现方法

系统实现方法	自动化特征工程方法
只有特征生成	FeatureTools[7]
先特征生成后特征选择	AutoCross[8] AutoLearn[9] SAFE[12]
基于层次贪婪搜索	Cognito[10]
基于迁移学习	ExploreKit[11]
基于元学习	LFE[15]
基于循环神经网络	RNNFE[14]
基于强化学习	RLFE[13]
基于卷积神经网络	CNN[26]
基于自动编码器	Autoencoder[17]

2.1 只有特征生成

Featuretools[7]是基于 DFS 的自动化特征工程方法,可用于多表关联生成特征的场景。Featuretools 的输入包括三部分:(1) 描述实体集的数据表;(2) 描述实体集之间关系的关系对,即 DFS 中的前向(Forward)和后向(Backward)关系;(3) 搜索深度。

Featuretools 的特征生成过程是利用实体集之间的关系进行特征生成探索,比如基于用户实体和交易商品实体的购买关系,可生成“用户历史购物商品的平均价格”这一特征。当搜索深度达到预设的深度时,Featuretools 停止生成特征。

工业界的数据存储在多张表中,且表与表之间关系复杂。Featuretools 在工业级的数据上生成的特征相比人工生成的特征效率高出很多。然而,该方法没有特征选择功能,需要数据科学家额外开发代码进行特征选择。此外,实际应用场景中存在某些实体间的关系不满足 DFS 的前向关系和后向关系,此时 Featuretools 无法有效生成特征。

2.2 先特征生成后特征选择

先特征生成后特征选择的自动化特征工程方法包括 AutoCross、AutoLearn 和 SAFE。

2.2.1 AutoCross

AutoCross 在特征生成阶段采用基于交叉运算的方法,在特征选择阶段采用封装型的方法。

在基于交叉运算的特征生成过程中,是将两个特征的取值直接进行组合生成新的特征。然而这种特征组合方法是无穷的,遍历所有特征组合几乎不可能实现。因此 AutoCross 提出了一种基于 Beam Search[36]的自动化特征生成方法,用于提升交叉特征生成的效率。AutoCross 首先将交叉特征的生成过程转换为特征生成树[37]的拓展过程,树的根节点为原始的特征集。树的每个子节点的特征集由两部分组成:(1) 父节点的特征;(2) 父节点中两个备选特征进行交叉运算生成的新特征。父节点的所有子节点包含了父节点下所有可能的交叉特征。每一次迭代先遍历当前节点的所有子节点,并选出性能最优的子节点,之后以该节点作为下一层特征生成的父节点,继续生成特征。特征生成后,AutoCross 使用封装型方法进行特征选择[38]。当达到如下三个条件之一时停止搜索:(1) 运行时间达到预设值;(2) 模型的性能达到预设值;(3) 生成的特征数量达到预设值。

AutoCross 基于 Beam Search 的算法避免了遍历所有可能的交叉特征组合,提升了交叉特征的生成效率。此外,AutoCross 每次迭代只选择最优的子节点进行拓展,这会导致过早结束其他路径的搜索,可能会错过一些高质量的特征,使得最终选取的特征集合不是全局最优的[39]。

2.2.2 AutoLearn

AutoLearn 在特征生成阶段采用基于特征相关性的方法,在特征选择阶段采用过滤型的方法。

AutoLearn 的搜索过程包括以下几个阶段:(1) 数据预处理,计算每一个特征的信息增益,过滤掉信息增益较小的特征;(2) 计算两个特征之间的相关性,并排除掉独立的特征,用于减少需要进行回归训练的特征空间,提高效率;(3) 以两个相关的特征作为输入,训练回归模型,如果两个特征之间存在线性关系则使用 Ridge Regression (RR)回归模型,如果存在非线性关系则使用 Kernel Ridge Regression (KRR)回归模型[40];(4) 通过回归模型的输出能够生成两个新特征,一个是回归模型输出的预测值,另一个是输出的预测值与真实值之间的差;(5) 选择其中信息增益较大的新特征,实现特征选择。在遍历完全部两个特征组合后,停止特征生成和选择。

AutoLearn 只需生成少量特征,构建的特征集就能够明显提升模型的性能[9]。但是 AutoLearn 只使用特征相关性作为特征生成的条件,在大量特征间不存在相关性时,会降低该方法的有效性[41]。

2.2.3 SAFE

SAFE[12]是面向工业场景的自动化特征工程方法。由于特征组合空间的无限性,穷举搜索往往是无效的。SAFE 为了避免特征组合带来的特征空间的爆炸,使用 XGBoost[42]模型挖掘不同特征组合之间的关系,并使用信息增益率对生成的特征组合进行排序并删除信息增益率较低的特征组合,进而缩小特征组合的搜索空间。SAFE 在特征生成阶段包括:(1) 使用 XGBoost 方法在原始的特征集上进行训练,得到构建好的回归树模型[43];(2) 按照得到的回归树模型的同一路径进行特征生成;(3) 使用信息增益率对生成的特征组合进行排序并保留信息增益率较高的特征组合;(4) 将信息增益率高的特征组合和原始特征进行合并,得到新特征集。

由于得到的新特征集仍然具有较多的特征,SAFE 设计了一种新的信息值计算方法,对新特征集合中的每一个特征进行评估,删除低预测能力和冗余的特征,并选择重要的特征构建最优的特征集。SAFE 在特征选择阶段包括:(1) 设计一种新的信息值计算方法,用信息值过滤掉低预测能力的特征;(2) 基于皮尔逊相关系数去除冗余的特征[44];(3) 对于剩下的特征再用 XGBoost 模型进行训练,得到每一个特征的平均信息增益[45];(4) 将平均信息增益较高的特征组合形成最优的特征集。SAFE 重复迭代上述特征生成和特征选择过程,在达到训练时间或者占用存储达到阈值时停止算法搜索。

SAFE 在特征生成阶段能够过滤掉低预测能力和冗余的特征,并在特征选择阶段使用混合型方法进行筛选,降低了算法的时间复杂度与空间复杂度,使其成为一种高效的自动特征工程方法。但该方法在特征选择阶段存在删除掉有效特征的可能,且算法迭代次数以及树的数量和深度等超参数对生成特征效果的影响较大[45]。

2.3 基于层次贪婪搜索

Cognito[10]是一种基于层次贪婪搜索的自动化特征工程方法,该方法将特征生成问题转化成特征生成树上的节点搜索问题。Cognito 在每一层先使用函数映射方法进行特征生成,之后使用封装型方法进行特征选择。

Cognito 将特征工程问题转化成特征生成树上最优节点的搜索问题。特征生成树构造过程如下:(1) 以原始数据集作为根节点,计算模型在根节点上的性能;(2) 对根节点包含的特征,使用函数映射方法生成新的特征,作为根节点的子节点;(3) 遍历函数集中包含的所有函数映射方法,应用于根节点,生成根节点的所有子节点;(4) 以此类推,遍历所有层的父节点,生成子节点,节点代表应用函数映射后的数据集,边代表应用的映射函数;(5) 分别计算每个节点的模型性能(如准确率),记为节点的一个属性。

然而,将函数集中所有函数映射方法应用于每一个生成树节点会产生大量的特征,且随着生成树层数的增加呈指数级增长[46]。为了解决这一问题,Cognito 在每一次特征生成之后都会根据节点的性能进行特征选择,用于减少下一层的特征生成的基数。此外,不同的机器学习模型适用于不同的数据集[47],为了选择最优的模型和性能,每次特征选择时把所有机器学习模型都训练一遍将消耗巨大的计算资源。因此 Cognito 使用了三种方法减少这种资源的消耗:(1) 递增分配数据给模型训练,减少模型训练的计算量;(2) 用性能曲线的切线变化来估计模型最优的性能[48],丢弃预估性能不好的节点,减少运算次数;(3) 在特征生成初期就锁定某个模型,减少需要训练的模型个数。

Cognito 的系统实现支持深度优先、广度优先、先广度后深度的平衡搜索策略,达到如下条件之一时停止算法:(1) 树节点达到预设节点数量;(2) 树的深度达到预设深度;(3) 生成特征数量达到预设值。

Cognito 通过上述递增分配数据与切线评估最优性能的方法提高了自动特征工程的效率,但每一轮次的特征选择容易丢弃有价值的特征,可能会导致最终构建的特征集不是全局最优的。此外,每个节点都需要进行一次模型训练,会消耗大量的计算资源[49]。

2.4 基于迁移学习

ExploreKit[11]是一种基于迁移学习[50]的自动化特征工程方法,其采用函数映射的方法生成特征,并采用迁移学习的方法指导特征选择,在系统实现上采用基于贪婪搜索的方法。

ExploreKit 的执行过程如下:(1) 统计原始特征在多个维度上的表征值,如特征取值个数、特征类型、模型的 AUC 等指标;(2) 以在历史机器学习任务构建的一个候选特征排序模型作为先验知识,指导后续的特征选择;(3) 对于一组原始数据首先使用函数映射方法(包括一元、二元和多元函数)生成大量的新特征;(4) 计算新特征的表征值;(5) 根据(2)中训练好的模型对生成的新特征进行排序;(6) 筛选排名靠前的特征进行模型训练,用模型训练的结果选择新生成的特征,进而构建新特征集;(7) 重复迭代上述(3)到(6)过程,当运行时间达到预设值时算法停止。

ExploreKit 方法可以有效地提升下游机器学习的性能。但该方法的实现复杂度较高,资源消耗较大。

2.5 基于元学习

LFE[15]是一种基于元学习[16]的自动特征工程方法。LFE 设计了一种基于 MLP 的特征生成方法,其根据历史机器学习任务学习到的经验,对原始特征推荐最优的映射函数方法,进而生成有效的新特征,且不需要进行特征选择。

LFE 生成新特征的过程如下:(1) 将历史机器学习任务的特征数据使用统一的方法进行表征;(2) 基于历史机器学习任务生成标签,标签代表特定映射函数生成的特征是否是有效特征;(3) 基于(1)中的表征数据与(2)的标签数据为每个映射函数训练 MLP 模型;(4) 在新的特征工程任务中,LFE 将特征按(1)中方法进行表征,并作为(3)中已训练好的 MLP 模型的输入,进而依据 MLP 模型的结果选择有效的映射函数;(5) 基于 MLP 模型选择的映射函数生成新的特征。

LFE 表征特征 f 的方法如下:

$$\boldsymbol{R}_f = [\boldsymbol{Q}_f^1; \boldsymbol{Q}_f^2; \cdots; \boldsymbol{Q}_f^k] \tag{2}$$

式中:$\boldsymbol{Q}_f^i$ 代表第 i 个类别的特征值的表征。

V_k 代表特征 f 的第 k 类标签的取值集合,$[l_b, u_b]$ 为 V_k 的取值区间,将该区间划分为 r 个互不相交的均匀区间:

$$\{b_0, b_1, \cdots, b_{r-1}\}$$

通过式(3)计算每个区间的归一化数值:

$$I(b_j) = \frac{P(b_j)}{\sum_{0<m<r} P(b_m)} \tag{3}$$

式中:$P(b_j)$ 代表第 j 个区间的特征值个数。最终得到 $\boldsymbol{Q}_f^i$:

$$\boldsymbol{Q}_f^i = [I(b_0), I(b_1), \cdots, I(b_{r-1})]$$

假设一个 k 分类问题,将特征 f 划分为 b 个区间,则 $\boldsymbol{R}_f$ 为 $k \times b$ 固定大小的矩阵,LFE 以 $\boldsymbol{R}_f$ 为输入,训练每个映射函数的 MLP 分类器,结构如式(4)所示。

$$[p_{t\text{isuseful}}(f), p_{t\text{isnotuseful}}(f)] = \sigma_2(\boldsymbol{B}^{(2)} + \boldsymbol{W}^{(2)}(\sigma_1(\boldsymbol{B}^{(1)} + \boldsymbol{W}^{(1)}\boldsymbol{R}_f))) \tag{4}$$

式中:p_t 为标签,$\boldsymbol{W}$ 和 $\boldsymbol{B}$ 为模型的权重与偏置系数,σ 为激活函数。

传统的特征选择需要消耗大量的计算资源来进行模型验证,而 LFE 直接选择有效的映射函数来生成新特征,不需要进行特征选择,从而提高了特征工程的效率。但该方法在 MLP 训练阶段会消耗较大的计算资源,且已训练好的 MLP 在一些与历史数据集完全不相似的新数据上无法生成有效的新特征[15],因此 LFE 方法适用范围有限。

2.6 基于循环神经网络

RNNFE[14]是利用 RNN 结构[51]实现自动化特征工程的方法。比起生成最优的特征集,RNNFE 致力于为每个特征寻找最优的特征生成规则。RNNFE 为每一个原始特征生成一个独立的 RNN 网络,网络的结构代表对应的原始特征生成新特征的规则,网络的输出为新生成的特征[52]。

对于有 n 个原始特征的数据集,RNNFE 将生成 n 个独立的 RNN 网络。每个 RNN 网络内部由多个 time step 串联组成,上游 time step 的输出为下游的输入。每一个 time step 对输入特征执行一次函数映射生成新特征输出,最后一个 time step 的输出即为新生成的特征。假设 $\boldsymbol{f}$ 为原始特征,对应的 RNN 网络由 t 个 time step 组成,第 i 个 time step 使用的映射函数记为 F_i,则该 RNN 网络生成的新特征如式(5)所示。

$$\hat{\boldsymbol{f}} = F_t(\cdots F_i(\cdots F_2(F_1(\boldsymbol{f}))\cdots)\cdots) \tag{5}$$

RNNFE 并行地为每个特征生成 RNN 结构,生成过程如下:(1) 对原始特征生成第一个 time step;(2) 将该 RNN 结构的输出作为下游机器学习模型的输入进行模型训练;(3) 基于(2)中训练的模型评估性能的提升效果;(4) 基于(3)的评估结果采用策略梯度的方法[53]指导下一个 time step 的生成;(5) 重复步骤(2)到步骤(4)直到遇到终止函数或达到最大的 time step 时停止。

RNNFE 在特征生成阶段采用函数映射的方法,在特征选择阶段采用嵌入式的方法。RNNFE 设计了一个特殊的函数 delete,用于删除没有提升效果的特征,

在生成 RNN 结构的过程中起到特征选择的作用。

RNNFE 可以解决特征生成过程中的维度爆炸问题,提高了特征生成效率,同时可以实现高阶的特征生成规则。但因其生成方法对特征间的相关性信息感知较弱,无法生成描述特征相关性的特征,会造成生成的新特征有部分局限性,致使其适用范围有限。

2.7　基于强化学习

RLFE[13]将自动特征工程的特征生成空间定义为有向无环图,图中的每一个节点代表了一组生成特征,自动特征工程目的为在该图中找到最优的节点,使得下游的机器学习模型的性能达到最优。

如图 2 所示,在有向无环图中,$\boldsymbol{D}_0$为根节点,代表原始特征集,设$\boldsymbol{D}_i$为图 2 中的第 i 个节点,代表了其父节点经特征变换后生成的新特征,e_{ij}为连接$\boldsymbol{D}_i$和$\boldsymbol{D}_j$的边,代表一种映射函数使得特征集$\boldsymbol{D}_i$通过该映射函数生成了新特征$\boldsymbol{D}_j$。基于上述规则,以$\boldsymbol{D}_0$为根节点,拓展出一个有向无环图代表了所有的特征变换。

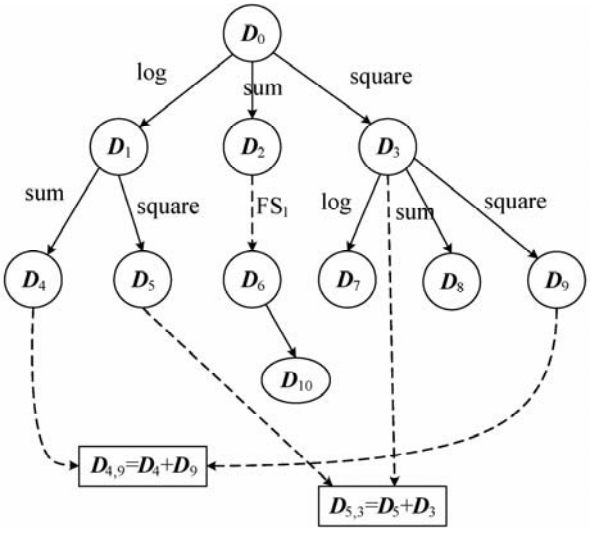


图 2　RLFE 的特征生成过程示例[13]

RLFE 从$\boldsymbol{D}_0$出发,基于强化学习[54]策略探索图中的节点,每探索一个节点就基于该节点的特征集执行一次机器学习训练,并评估模型性能,同时以该性能指导强化学习的下一次探索。RLFE 的每一次探索,实际上是在选择生成新特征的映射函数,并基于该函数生成新节点,即执行一次特征选择,探索到最优性能的节点时,该节点的特征集即为最终选择的特征。

RLFE 在探索过程中使用的探索策略是基于强化学习算法 Q-learning[55]学习的策略。RLFE 基于下游机器学习模型的性能以及执行的步数等设计了 Q-learning 的 Reward 函数,以图当前探索节点为 state,以可选的映射函数为 action。当探索达到时间限制或达到步数限制或达到图深度限制[55]时算法停止,并输出最优节点的特征集。

RLFE 是第一个将强化学习应用到自动特征工程的问题上,相比非强化学习方法,强化学习策略可以更高效地生成有效的特征。但是该方法复杂度较高,效果受图深度等参数影响较大。

2.8　基于卷积神经网络

基于卷积神经网络的方法是利用卷积神经网络的训练过程进行特征生成和特征选择的方法。卷积神经网络是面向图像数据的有监督机器学习方法,可将该网络分成三个依次串联的网络层:(1) 输入层;(2) 卷积网络层;(3) 输出层[26]。该网络训练的过程中,卷积网络层基于输入层的数据以及输出层的反馈调整自身的参数,不同的参数是对输入数据做不同的特征生成、变换和选择,因此参数的调整过程即是特征工程的过程,该过程同时实现特征生成和特征选择。训练完成时,卷积网络层可独立作为一种特征工程方法应用到相似的机器学习任务中。

该方法可以有效提取图像局部细节特征,在图像识别领域取得了较好的效果[16,26]。但该结构的网络层数,每层节点数目对效果影响大,需要多次调整,且生成的特征可解释性较差。

2.9　基于自动编码器

自动编码器(Autoencoder)是一个无监督的网络结构,主要包含了编码层(encoder)和解码层(decoder)两个模块。其中编码层在训练过程中完成特征工程。自动编码器的结构可分成 5 个依次串联的层:(1) 输入层;(2) 编码层;(3) 隐藏层 h;(4) 解码层;(5) 输出层。原始特征从(1)输入自动编码器,经(2)进行压缩,之后通过(3)传递给(4),经(4)还原后通过(5)输出,训练过程使(5)的输出尽可能等于(1)的输入。完成训练时,(2)将原始特征压缩成了新的特征,完成了特征工程的功能。此时可将(2)独立作为一个特征工程模块应用于机器学习任务中。

该方法的优点是无需数据标签,可以有效降低原始数据的特征维度和去除数据噪点,缺点是数据压缩效果受隐藏层的大小的影响较大,需要多次反复调整,且生成特征可解释性较差。

3　自动特征工程研究存在的难点

现有的自动特征工程系统使用了不同的特征生成和特征选择方法,如表 2 所示。

表 2 自动特征工程系统使用的特征生成与选择方法

自动化特征工程方法	特征生成方法	特征选择方法
FeatureTools[7]	DFS	无
AutoCross[8]	交叉特征	封装型
AutoLearn[9]	特征相似性	过滤型
SAFE[12]	函数映射	混合型
Cognito[10]	函数映射	封装型
ExploreKit[11]	函数映射	基于迁移学习的方法
LFE[15]	函数映射	无
RNNFE[14]	函数映射	嵌入型
RLFE[13]	函数映射	嵌入型
CNN	卷积核	嵌入型
Autoencoder	编码器	嵌入型

可见现有方法的特征生成方法大部分是基于函数映射的,然而特征选择方法却没有固定的模式。

表 1 和表 2 中所述现有自动化特征工程系统的特征生成、特征选择和系统实现方案都是为了解决自动化特征工程存在的难点问题而设计的,我们总结现有自动化特征工程存在的难点问题如下。

3.1 为了避免特征生成带来的维度爆炸,如何有效地生成特征

由于特征生成空间的无限性,特征生成过程中面临特征维度爆炸的问题。如交叉特征生成方法,其特征生成空间复杂度为 $O(2^d)$,其中 d 为特征的个数。当有 100 个原始特征时,如果对所有特征生成新的交叉特征,将生成约 $1.26e^{30}$个新特征。类似的问题也出现在其他特征生成方法上。现有解决该难点的方法有:(1) 设置限制条件,降低特征的生成数量,如 DFS 设置搜索深度来阻止复杂关系特征的生成,从而避免维度爆炸;(2) 预先判断有效性,减少无效特征和冗余特征的生成,如 SAFE 中通过预先计算特征的信息增益率来筛除信息增益率低的特征,避免其参与后续的特征生成造成维度爆炸;(3) 优化特征生成方法,如 AutoCross 中使用 Beam Search 算法来优化特征生成方法,解决了生成全部的交叉特征导致维度爆炸。

现有方法中并没有给出通用解决方法,因此在实际应用中仍需要根据不同计算条件和应用场景选择合适的特征生成方法。

3.2 如何从特征生成的新特征中快速、高效地选择有效特征

特征选择的目的是从特征生成的新特征中筛选有效特征,排除无效特征和冗余特征。在评估新特征的有效性时,使用下游机器学习模型的性能来评估是最优的解决方案,但这种模式却存在着资源消耗过大的难点问题。过滤型方法虽然解决了资源消耗大的难点,但筛选有效特征的效果不稳定,使得下游的模型性能不可控。混合型方法是封装型方法和过滤型方法的结合,减少了一部分的资源消耗,同时也牺牲了选择有效特征的一部分稳定性。此外,基于迁移学习的特征选择方法也是一种有效的解决方案,但仍需要较大的计算资源进行性能评估。

现有方法中并没有给出兼顾资源消耗和模型性能的通用方法,因此在实际应用中仍需要根据不同计算条件和应用场景选择合适的特征选择方法。

3.3 如何结合特征生成和特征选择,设计最优的结构,实现最优特征集合的构建

系统实现设计是通过整合特征生成和特征选择算法,实现特征工程的自动化。其设计过程存在的难点是如何结合特征生成和特征选择,设计最优的结构,实现最优特征集合的构建。现有解决该难点的方法有:(1) 传统的方法,其是将特征生成和特征选择视为两个独立的事件,先进行特征生成,后进行特征选择,如 AutoCross、AutoLearn 和 SAFE 的方法均是采用此模式。但这种模式需要评估生成特征的有效性,会消耗较大的计算资源。(2) 使用不同的搜索策略实现特征生成和特征选择,如 Cognito、RNNFE 和 RLFE 方法。但这种模式同样会消耗较大的计算资源。(3) 基于历史任务得到的先验知识指导特征生成和选择的方向,如 ExploreKit 和 LFE 方法,这种模式能够降低模型运算次数,提高搜索效率。但此类方法受限于历史任务的局限性,在一些新的数据上不一定适用。(4) 面向图像数据、将特征生成和特征选择结合在一起的 CNN 和 Autoencoder 方法,这两种方法能够有效地进行特征工程,但其训练过程需要反复迭代,需要消耗大量的计算资源,此外,通过 CNN 和 Autoencoder 生成的特征可解释性较差。

现有方法采用多种系统实现完成了特征生成和特征选择,但仍存在着资源消耗较大、特征可解释性差的问题。因此设计能够进一步降低资源消耗且特征解释性好的系统是仍待解决的问题。

4 自动化特征工程未来研究方向

4.1 通过并行计算提高特征工程搜索效率

特征生成过程中,大部分新特征的生成过程相互独立。另外,特征选择过程中,不同特征的有效性判断相互独立。因此可以使用并行计算的方法提高特征工

程的搜索效率。目前,有 RNNFE 在生成 RNN 结构的过程中采用了并行化的方法,并行生成不同特征的 RNN 结构,提高了整体的特征工程效率。因此如何在特征生成和特征选择过程中通过使用并行化方法提高特征工程效率是值得研究的一个方向。

4.2 更好地利用迁移学习和元学习进行特征生成和特征选择

迁移学习和元学习可以从历史的机器学习任务中学习总结经验,指导新任务中的特征生成和特征选择。如基于迁移学习的 ExploreKit 方法,在特征生成之后使用迁移学习指导特征选择,能够高效地选择出有效特征提升下游机器学习模型的性能。此外,基于元学习的 LFE 方法,从历史机器学习任务中学习生成有效特征的经验,在新的特征工程任务中指导特征生成,同样能够高效地生成有效的特征提升下游机器学习模型的性能。然而当历史机器学习任务与新任务存在差异时,采用迁移学习和元学习的方法容易过早舍弃生成有效特征的方法。因此如何从历史机器学习任务中提取先验知识,将迁移学习和元学习两种方法更好地应用到特征生成和特征选择上,使其能够在与历史任务不相似的数据中依然能够得到较好的新特征集,是值得深入研究的一个方向。

4.3 将特征工程和模型超参数优化结合起来一起提高性能

特征工程的目的是生成有利于提高下游机器学习模型性能的特征。当前的自动特征工程在评估下游机器学习模型性能的时候,都是基于 Baseline 的超参数进行评估,但该超参数对于任务应用的场景而言未必是最优的超参数,这将导致生成的特征在 Baseline 的超参数下不是最优的特征。因此将特征工程和模型超参数优化结合,一起提高机器学习模型性能是值得研究的一个方向。

4.4 增加 Autoencoder 和 CNN 网络生成特征的可解释性

在神经网络的实际应用过程中,如果生成的图像特征可以提高模型的效果则表明该特征是一个有效的特征,但并不代表该特征是有意义的,能被最终使用。比如有一组狼和狗的图像分类任务,假设输入模型训练的图像中,所有标注为狼的图像都是雪地背景,所有标注狗的图像都是非雪地背景,在 CNN 和 Autoencoder 的特征工程中会生成判断背景是否是雪地的新特征,该特征可以有效提高分类的准确性,但很明显该特征并不能应用于实际的狼和狗的分类任务。Autoencoder 和 CNN 网络在一些场景的特征工程具有非常优秀的效果,但因为神经网络的整个特征工程是一个黑盒,导致其生成的特征可解释性较差,并不能确定生成的特征是有意义的,这成为 CNN 和 Autoencoder 实际应用的一个阻力。因此如何增加 CNN 和 Autoencoder 生成特征的可解释性是一个值得研究的方向。

5 结　语

本文系统性介绍自动特征工程的研究现状。首先介绍现有的特征生成和选择方法,然后分类介绍自动特征工程的研究进展,在此基础上进行归纳总结,指出自动特征工程面临的难点以及未来的研究方向,为读者了解和使用自动特征工程提供参考。

展望未来,相信自动化特征工程可以降低机器学习的使用门槛、缩短机器学习的任务周期,进一步提升机器学习的产能。

参 考 文 献

[1] Zhu H, Jin J, Tan C, et al. Optimized cost per click in-taobao display advertising[C]//23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017:2191 - 2200.

[2] Domingos P. A few useful things to know about machine learning[J]. Communications of the ACM, 2012, 55(10): 78 - 87.

[3] Breiman L. Statistical modeling: The two cultures (with comments and a rejoinder by the author)[J]. Statistical Science, 2001, 16(3):199 - 231.

[4] Yao Q, Wang M, Chen Y, et al. Taking human out of learning applications: A survey on automated machine learning[EB]. arXiv:1810.13306, 2018.

[5] Wind D. Concepts in predictive machine learning[D]. Technical University of Denmark, 2014.

[6] He X, Zhao K, Chu X. AutoML: A survey of the State-of-the-Art[J]. Knowledge-Based Systems, 2021, 212:106622.

[7] Kanter J, Veeramachaneni K. Deep feature synthesis: Towards automating data science endeavors[C]//2015 IEEE-International Conference on Data Science and Advanced Analytics, 2015:1 - 10.

[8] Luo Y, Wang M, Zhou H, et al. Autocross: Automatic feature crossing for tabular data in real-world applications[C]//25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019:1936 - 1945.

[9] Kaul A, Maheshwary S, Pudi V. Autolearn—Automated feature generation and selection[C]//2017 IEEE International Conference on Data Mining, 2017:217 - 226.

[10] Khurana U, Turaga D, Samulowitz H, et al. Cognito: Auto-

mated feature engineering for supervised learning[C]//2016 IEEE 16th International Conference on Data Mining Workshops,2016:1304 - 1307.

[11] Katz G, Shin E, Song D. ExploreKit: Automatic feature generation and selection[C]//2016 IEEE 16th International Conference on Data Mining,2016:979 - 984.

[12] Shi Q, Zhang Y L, Li L, et al. Safe: Scalable automatic feature engineering framework for industrial tasks[C]//2020 IEEE 36th International Conference on Data Engineering, 2020:1645 - 1656.

[13] Khurana U, Samulowitz H, Turaga D. Feature engineering for predictive modeling using reinforcement learning[C]// AAAI Conference on Artificial Intelligence, 2018: 3407 - 3414.

[14] Chen X, Lin Q, Luo C, et al. Neural feature search: A neural architecture for automated feature engineering[C]// 2019 IEEE International Conference on Data Mining,2019: 71 - 80.

[15] Nargesian F, Samulowitz H, Khurana U, et al. Learning feature engineering for classification[C]//26th International Joint Conference on Artificial Intelligence, 2017: 2529 - 2535.

[16] Chen X, Xu Y, Yan S, et al. Automatic feature learning for glaucoma detection based on deep learning[C]//International Conference on Medical Image Computing and Computer-Assisted Intervention,2015:669 - 677.

[17] Fan C, Sun Y, Zhao Y, et al. Deep learning-based feature engineering methods for improved building energy prediction [J]. Applied Energy,2019,240:35 - 45.

[18] Cho K,Merriënboer V, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[EB]. arXiv:1406.1078,2014.

[19] Yeung H W F, Li J, Chung YY. Improved performance of face recognition using CNN with constrained triplet loss layer [C]//2017 International Joint Conference on Neural Networks,2017:1948 - 1955.

[20] 张东方,陈海燕,王建东.半监督特征选择综述[J].计算机应用研究,2021,38(2):321 - 329.

[21] Dor O, Reich Y. Strengthening learning algorithms by feature discovery[J]. Information Sciences,2012,189:176 - 190.

[22] Da K. A method for stochastic optimization[EB]. arXiv: 1412.6980,2014.

[23] Lam H, Thiebaut J, Sinn M, et al. One button machine for automating feature engineering in relational databases[EB]. arXiv:1706.00327,2017.

[24] Kong W, Valiant G. Estimating learnability in the sublinear data regime[EB]. arXiv:1805.01626,2018.

[25] Jun K, Lee D, Lee K, et al. Feature extraction using an RNN autoencoder for skeleton-based abnormal gait recognition[J]. IEEE Access,2020,8:19196 - 19207.

[26] Xiao G, Li J, Chen Y, et al. MalFCS: An effective malware classification framework with automated feature extraction based on deep convolutional neural networks[J]. Journal of Parallel and Distributed Computing,2020,141:49 - 58.

[27] Nataraj L, Karthikeyan S, Jacob G, et al. Malware images: Visualization and automatic classification[C]//8th International Symposium on Visualization for Cyber Security,2011: 1 - 7.

[28] Akadi A E, Amine A, Ouardighi A E, et al. A two-stage gene selection scheme utilizing MRMR filter and GA wrapper [J]. Knowledge and Information Systems,2011,26(3):487 - 500.

[29] Sheth R, Fusi N. Featuregradients: Scalable feature selection via discrete relaxation[EB]. arXiv:1908.10382,2019.

[30] Kohavi R, John G. Wrappers for feature subset selection [J]. Artificial Intelligence,1997,97(1 - 2):273 - 324.

[31] Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics[J]. Bioinformatics, 2007, 23 (19):2507 - 2517.

[32] Kong W, Valiant G. Estimating learnability in the sublinear data regime[EB]. arXiv:1805.01626,2018.

[33] Zhang Y L, Li L, Zhou J, et al. Anomaly detection with partially observed anomalies[C]//The Web Conference, 2018:639 - 646.

[34] Markovitch S, Rosenstein D. Feature generation using general constructor functions[J]. Machine Learning, 2002, 49 (1):59 - 98.

[35] Vanschoren J. Meta-learning[M]//Automated Machine Learning. Springer,2019:35 - 61.

[36] Zhang Y L, Zhou J, Zheng W, et al. Distributed deep forest and its application to automatic detection of cash-out fraud [J]. ACM Transactions on Intelligent Systems and Technology,2019,10(5):1 - 19.

[37] Dor O, Reich Y. Strengthening learning algorithms by feature discovery[J]. Information Sciences, 2012,189:176 - 190.

[38] Pang M, Ting K M, Zhao P, et al. Improving deep forest by confidence screening[C]//2018 IEEE International Conference on Data Mining,2018:1194 - 1199.

[39] Alajmi M,Awedat K, Essa A, et al. Efficient face recognition using regularized adaptive non-local sparse coding[J]. IEEE Access,2019,7:10653 - 10662.

[40] Zöller M A, Huber M F. Survey on automated machine learning[EB]. arXiv:1904.12054,2019.

[41] Chen T,Guestrin C. XGBoost: A scalable tree boosting system[C]//22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,2016:785 - 794.

(下转第 40 页)

测比社交媒体数据预测更能提前获取时尚趋势。本文将时尚秀场数据与神经网络相结合,为时尚趋势预测提供新的思路。实验结果表明,模型在 FIT 和 GeoStyle 两个数据集上都表现良好,且展示了单个时尚元素趋势与谷歌趋势之间的比较案例,为解决时尚趋势预测提供了简单可行的方案。

参 考 文 献

[1] Li X C, Wang X, He X N, et al. Hierarchical fashion graph network for personalized outfit recommendation[C]//43rd International ACM SIGIR Conference on Research and Development in Information Retrieval,2020:159 - 168.

[2] 王安琪,刘骊,付晓东,等. 面向个性化服装推荐的判断优化模型[J]. 计算机工程与应用,2018,54(11):1456 - 1465.

[3] 苏卓,柯司博,王若梅,等. 深度多模态融合服装风格检索[J]. 中国图象图形学报,2021,26(4):857 - 871.

[4] Ma Y S, Ding Y J, Yang X, et al. Knowledge enhanced neural fashion trend forecasting[C]//International Conference on Multimedia Retrieval,2020:82 - 90.

[5] 赵黎,杨连贺,黄新. 采用多蜂群协同演化算法的服装流行色预测[J]. 纺织学报,2018,39(3):137 - 142.

[6] Al-Halah Z, Stiefelhagen R, Grauman K. Fashion forward: Forecasting visual style in fashion[C]//IEEE International Conference on Computer Vision,2017:388 - 397.

[7] Vogue. Fashion show—Vogue, fashion magazine[EB/OL]. [2022 - 05 - 16]. http://www.vogue.com.cn/.

[8] Kim E, Fiore A M, Payne A, et al. Fashion trends: Analysis and forecasting[M]. London: Bloomsbury Publishing,2021.

[9] Wen R, Torkkola K, Narayanaswamy B, et al. A multi-horizon quantile recurrent forecaster[EB]. arXiv:1711.11053, 2017.

[10] Mall U, Matzen K, Hariharan B, et al. Geostyle: Discovering fashion trends and events[C]//IEEE International Conference on Computer Vision,2019:411 - 420.

[11] Matzen K, Bala K, Snavely N. Streetstyle: Exploring worldwide clothing styles from millions of photos[EB]. arXiv: 1706.01869,2017.

[12] Gilbert Walker. On periodicity in series of related terms[J]. Proceedings of the Royal Society of London, Series A: Containing Papers of a Mathematical and Physical Character, 1931,131(818):1905 - 1934.

[13] Slutzky E. The summation of random causes as the source of cyclic processes[J]. Econometrica: Journal of the Econometric Society,1937,12:105 - 146.

[14] Box G E P, Jenkins G M. Some recent advances in forecasting and control[J]. Journal of the Royal Statistical Society, 1968,17(2):91 - 109.

[15] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation,1997,9(8):1735 - 1780.

[16] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[EB]. arXiv:1409.0473,2014.

[17] Fan C Y, Zhang Y Z, Pan Y, et al. Multi-horizon time series forecasting with temporal attention learning[C]//25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,2019:2527 - 2535.

(上接第 10 页)

[42] Fang W, Zhou J, Li X, et al. Unpack local model interpretation forGBDT[C]//International Conference on Database Systems for Advanced Applications,2018:764 - 775.

[43] Song W, Shi C, Xiao Z, et al. AutoInt: Automatic feature interaction learning via self-attentive neural networks[C]// 28th ACM International Conference on Information and Knowledge Management,2019:1161 - 1170.

[44] Zhang Y, Li L. Interpretable MTL from heterogeneous domains using boosted tree[C]//28th ACM International Conference on Information and Knowledge Management,2019: 2053 - 2056.

[45] Fan W, Zhong E, Peng J, et al. Generalized and heuristic-free feature construction for improved accuracy[C]//2010 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics,2010:629 - 640.

[46] Ho T. The random subspace method for constructing decision forests[J]. IEEETransactions on Pattern Analysis and Machine Intelligence,1998,20(8):832 - 844.

[47] Sabharwal A,Samulowitz H, Tesauro G. Selecting near-optimal learners via incremental data allocation[C]//30th AAAI Conference on Artificial Intelligence,2016:2007 - 2015.

[48] Hutter F, Kotthoff L, Vanschoren J. Automated machine learning: Methods, systems, challenges[M]. Springer,2019.

[49] Torrey L, Shavlik J. Transfer learning[M]//Handbook ofResearch on Machine Learning Applications and Trends: Algorithms,Methods,and Techniques. IGI Global,2010:242 - 264.

[50] Mirhoseini A, Goldie A, Pham H, et al. A hierarchical model for device placement[C]//International Conference on Learning Representations,2018:1 - 11.

[51] Medsker L, Jain L. Recurrent neural networks[J]. Design and Applications,2001,5:64 - 67.

[52] Ramachandran P, Zoph B, Le Q. Searching for activation functions[EB]. arXiv:1710.05941,2017.

[53] Irodova M, Sloan R. Reinforcement learning and function approximation[C]//18th International Florida Artificial Intelligence Research Society Conference,2005:455 - 460.

[54] Watkins C, Dayan P. Q-learning[J]. MachineLearning,1992, 8:279 - 292.

[55] Li L, Jamieson K, DeSalvo G, et al. Efficient hyperparameter optimization and infinitely many armed bandits[EB]. arXiv:1603.06560v1,2016.