计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201200189

一种可用于分类型属性数据的多变量回归森林

刘振宇[1]　宋晓莹[2]
1 东北大学计算机科学与工程学院　沈阳 110819
2 大连东软信息学院计算机学院　辽宁 大连 116023

摘　要　针对线性回归、SVR以及大部分多变量回归树等回归模型不能直接利用分类型属性进行回归分析的问题,提出了一种可联合多种类型属性的决策树结点划分方法。该方法通过定义样本集合在分类型属性上的中心以及样本到中心的距离,使得分类型属性也可以像数值型属性一样参与样本的聚类过程,从而形成样本集的划分。之后,文中又为由该方法产生的决策树选择了合适的集成方案,生成的集成器被称为聚类回归森林(CRF)。最后,在12个UCI公开数据集上对比CRF与其他9个回归模型的回归平均绝对误差(MAE)和均方根误差(RMSE),实验结果表明,CRF在10个回归模型中具有最好的表现。

关键词:决策树;多变量回归树;集成学习;随机森林;梯度提升
中图法分类号　TP393

Multivariate Regression Forest for Categorical Attribute Data

LIU Zhen-yu[1] and SONG Xiao-ying[2]
1 School of Computer Science and Engineering,Northeastern University,Shenyang 110819,China
2 School of Computer,Dalian Neusoft University of Information,Dalian,Liaoning 116023,China

Abstract　As categorical attributes cannot be utilized directly in some regression models like the linear regression,SVR and most multivariate regression trees,a multivariate split method dealing with multiple types of data is prompted in this paper. We define the centers of the sample sets on the categorical attributes and the distances from the samples to the centers in order that the categorical attributes can also participate in the clustering process like the numerical attributes. Then a reasonable ensemble scheme is selected for the decision trees generated by the method to get the ensemble called cluster regression forest(CRF). Finally,we use CRF and other 9 regression models to compare regression mean absolute error (MAE) and root mean square error (RMSE) on 12 UCI public data sets. The experimental results show that CRF has the best performance among the 10 regression models.

Keywords　Decision trees,Multi-variable regression trees,Ensemble learning,Random forest,Gradient boosting

1　引言

回归分析是机器学习、数据挖掘等研究的核心问题之一[1],被广泛应用于经济预测、工业控制、医疗诊断等诸多领域[2-3]。

在回归问题中,输入属性按照其是否存在有序性可分为数值型和分类型两类。如“体重”“年龄”等属性可采用实数或整数表示,属于数值型属性。而“学生专业”“籍贯”等属性,虽然可以使用整数枚举或字符串等表示,但由于其属性值之间的无序性,这些属性属于分类型属性。一些用于回归问题的算法,如线性回归、支持向量机、人工神经网络等,不能直接使用分类型属性建立模型。如果使用这些算法处理带有分类型属性的数据,则需要通过CRIMCOORD[4]、One-Hot等方法将分类型属性转换为一个或多个数值型属性。这种转换可能为回归问题带来新的偏差,从而降低生成模型的泛化能力。

决策树是一种常见的归纳式学习方法,可用于分类和回归问题。经典的决策树算法,如C4.5[5]和CART[6],在每次划分结点时仅仅使用一个属性,产生的分割超平面平行于其他坐标轴,因此这种决策树被称为单变量树或轴平行树。单变量决策树具备构建速度快、易解释、直接使用分类型属性等优点。

与轴平行决策树相对应的是斜决策树(属于多变量决策树)。斜决策树在划分每个结点时,线性联合多个数值型属性。在 p 个数值属性的空间中,结点中样本按照式(1)分到左右两个子结点中。

$$\sum_{l=1}^{p} a_l x_l \leqslant \theta \tag{1}$$

其中,a_l 表示第 l 个属性的系数,确定了连同 θ 在内的 $p+1$

到稿日期:2020-12-22　返修日期:2021-03-14　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61772101)
This work was supported by the National Natural Science Foundation of China(61772101).
通信作者:刘振宇(liuzhenyu@neusoft.edu.cn)

个系数就唯一确定了划分超平面。由于划分超平面的方向是任意的，在绝大多数情况下，斜划分比轴平行划分能够生成更小的决策树，同时获得更好的泛化性能。另外，在预测属性较少的情况下，多变量决策树比单变量决策树更加适合作为集成算法的基学习器。这是由于在同样的属性空间中，可供选择的斜划分超平面远远多于轴平行划分超平面，多变量联合划分能够给集成器带来更加多样的个体，从而改善集成器的整体表现。

在近几十年中，许多针对数值型数据的斜划分方法被提出，其中的一些方法是专为分类问题而设计，无法在回归问题中使用。CART-LC[6]（CART 的斜划分版本）是最早的斜决策树，它采用爬山法随机扰动各个特征在空间中的系数，以确定分割超平面。SADT[7]使用模拟退火算法搜索超平面，以避免陷入局部最优解。OC1[8]结合了 CART-LC 和 SADT 的思想。文献[6-8]中的方法需要尝试大量的候选超平面，因此构建速度较慢。为了弥补这个缺点，线性判别分析、主成分分析等技术被用于启发搜索适合的划分超平面。例如，FDT[9]是使用线性判别分析方法构建的斜决策树，但 FDT 仅能应用于二分类问题，对多分类问题及回归问题则不适用。文献[10]分别尝试费舍尔判别分析、稀疏线性判别分析以及岭回归 3 种方法来确定结点划分的超平面。文献[11]中的 HHCART 是 CART 的变种，它首先求得原始数据的特征向量，然后将数据映射至新的坐标空间，最后在新空间中搜索最佳轴平行划分超平面。

以上提及的多变量决策树和森林，虽然在面对数值型属性数据时比单变量决策树和森林有更好的表现，但由于无法对分类型属性进行线性组合，这些算法在处理带有分类型属性的数据之前，必须将分类型属性转换为数值型属性。文献[12]指出，其提出的斜树森林在分类型数据上的表现差于依靠单变量划分的随机森林[13]。另外，部分斜划分方法只能应用于分类问题，例如文献[9]中的方法，确定划分超平面时，需要利用训练数据的类别标签，这就使得这一类方法无法用于回归问题。

为解决分类型属性之间以及分类型属性与数值型属性之间无法联合在一起进行结点划分的问题，本文提出了一种基于聚类的结点划分方法。使用该方法构建的决策树，可以直接处理纯数值型数据、纯分类型数据以及混合型数据的回归问题。本文将此决策树作为基学习器，构建集成回归器，这个集成器被称为聚类回归森林（CRF）。实验结果表明，CRF 在各种类型的数据上均具备很好的泛化能力。

2 预备知识

在介绍提出的算法之前，本节简要回顾回归树及其集成算法。

2.1 回归树

预测目标的类型为离散值的机器学习任务被称为分类问题，预测目标的类型为连续值的任务被称为回归问题。决策树算法可以解决分类和回归问题，用于回归问题的决策树被称为回归树。普通回归树的叶结点表示一个预测值，而被称为模型树的特殊回归树的一类叶结点表示一个函数（通常是一个线性回归模型）。本文方法属于普通回归树。

CART[6]是最经典的回归树，它采用自顶向下、二路划分、递归的方式生长。每次划分时，选择能使当前结点不纯度下降最大的一个属性。当 CART 用于分类问题时，使用 GINI 指数来衡量结点的不纯度；当 CART 用于回归问题时，式(2)将被用来衡量结点的不纯度。

$$MSE(D)=\frac{1}{|D|}\sum_{i=1}^{|D|}(y_i-\bar{y})^2 \tag{2}$$

其中，$MSE(D)$表示训练集 D 中所有样本在目标属性上的均方误差，y_i 和$\bar{y}$分别表示第 i 个样本的目标属性值和 D 中所有样本的目标属性值的平均数。

CART 是二叉树，即每次划分时，被划分结点中的样本将被分到左右子结点中。如果划分依据的是数值型属性，需要选择一个切分点；如果划分依据的是分类型属性，则需要把所有属性值分成两组。本文方法除了在划分方式上不同外，在生长方式、停止条件等其他方面与 CART 类似。

2.2 决策树集成

集成学习通过将多个学习器进行结合，通常可以获得比单一学习器更优的泛化性能。集成器的性能不仅依赖于其使用的个体学习器的能力，也依赖于个体之间的多样性。由于决策树的不稳定性（训练集的变化易造成学得模型的变化），其成为多种集成学习框架中基学习器的理想候选者。

从个体学习器的创建次序角度来说，集成学习框架可分为“可并行”和“只串行”两类。Bagging[14]、随机子空间[15]以及随机森林是可并行产生个体的集成框架的代表。Bagging 依靠随机子采样（bootstrap 方法）来得到不同的训练样本子集，从而产生多样的个体；随机子空间则是通过随机选择属性子集来保障个体间的多样性；随机森林则可被认为是前面两种方法的结合。

在串行生成个体的集成框架中，得到广泛应用的是 Adaboost[16]和 GBDT（Gradient Boosting Decision Tree）[17]。Adaboost 采用增加被前一个个体学习器误判的样本的权重的方式来提升集成器的整体性能；而 GBDT（主要用于回归，也可以用于分类）使用前一轮生成的回归树对训练样本进行预测，并将预测值与实际目标值的差作为训练样本的新的目标值，用于下一棵回归树的构建。

以上提及的经典算法都是集成学习领域的代表，目前许多学者尝试将它们融合，从而获得结构更复杂、泛化能力更强的超级集成器。例如，Wang 等[18]使用遗传算法优化随机森林和 XGBoost（GBDT 的一个变体）的参数，并将二者融合；Qu 等[19]借鉴深度学习思想，将 GBDT 和 XGBoost 作为隐层，提出了一种深度梯度提升回归预测模型。本文的集成方案也将综合随机森林和 GBDT 的方法，但不如文献[18]和文献[19]中的集成器的结构复杂。

3 本文算法

本节首先以纯数值型属性数据为例介绍本文提出的结点划分方法，然后将其扩展到分类型数据和混合型数据，最后介绍回归森林（CRF）的构建。

3.1 加权聚类结点划分

前文曾提到，经典的回归树 CART 使用结点中样本输出

值的均方误差作为启发信息，在输入空间中搜索最优属性进行结点划分。而本文使用的划分方法不存在这个启发搜索的过程，该方法基于流形假设。流形假设指在输入空间中距离相近的点具有相似的输出值。经典的 k-means 聚类算法的优化目标是减小输入空间中样本到簇心距离的均方差，而结点划分的目标是减小样本输出值的均方误差。如果流形假设成立，则这两个优化目标虽然不完全相同，但基本一致。此时，聚类过程便可代替启发搜索的属性挑选过程。

为此，我们使用 2-means（$k=2$ 时的 k-means）算法划分结点中的样本。首先，我们找到当前结点中预测目标值最大和最小的两个样本，使用这两个样本作为初始的两个簇的中心，准备开始聚类迭代；然后，在每次迭代中，依据“到簇心最短距离”原则为每个样本确定其归属的簇，并在当前轮次分簇结束后更新两个簇心；迭代过程终止于两个簇心位置不再改变，或者达到预设的迭代次数。迭代终止后，两个簇心连线的中垂超平面就是我们为当前结点寻找到的划分超平面。

之前的讨论建立在流形假设成立的基础上，在数据的输入空间中存在大量与输出值无关的属性时，流形假设将不再成立，即输入空间中距离相近的点不一定具有相似的输出值。此时，简单地应用 k-means 算法进行结点划分将会产生偏离目标的结点划分结果，最终导致生成的决策树泛化能力差，结构复杂（结点多）。其原因是，经典的 k-means 算法是一种无监督学习算法，它不会考虑回归问题中各个输入属性与输出值的相关性，在计算样本到簇中心的距离时，平等地看待全部输入属性，过多与输出值无关的输入属性“淹没”了那些相关属性。解决这一问题的一种方法就是对属性加权。如果能够给予与回归目标强相关的输入属性较大的权重，则放大该属性对距离的贡献，否则给予较小的权重，减小弱相关和不相关属性对距离的贡献，这样可使 k-means 算法的优化目标与结点划分的目标趋向一致。

为此，在每次聚类迭代过程中，计算样本到簇心距离时，我们使用式(3)所示的加权欧氏距离。

$$dis\_n(\boldsymbol{x}_i,\boldsymbol{\mu}_j)=\sqrt{\sum_{l=1}^{p}w_l\,(x_{i,l}-\mu_{j,l})^2} \tag{3}$$

其中，$dis\_n(\boldsymbol{x}_i,\boldsymbol{\mu}_j)$ 表示样本 $\boldsymbol{x}_i$ 与簇心 $\boldsymbol{\mu}_j$ 的距离，p 表示属性的个数，w_l 表示第 l 个属性的权重，该权重按照式(4)计算获得。

$$w_l=|cov(\boldsymbol{A}_l,\boldsymbol{Y})|/\sqrt{var(\boldsymbol{A}_l)var(\boldsymbol{Y})} \tag{4}$$

其中，$cov(\boldsymbol{A}_l,\boldsymbol{Y})$ 表示属性 $\boldsymbol{A}_l$ 与输出 $\boldsymbol{Y}$ 的协方差，$var(\cdot)$ 表示方差。实际上，式(4)表示的权重就是输入属性 $\boldsymbol{A}_l$ 与输出 $\boldsymbol{Y}$ 的皮尔逊相关系数的绝对值，取值范围为 0～1。

综上，本文提出的结点划分方法可简单描述为属性加权和聚类两个步骤，具体流程如算法 1 所示。

算法 1 split

输入：当前结点的数据集 D，最大迭代次数 I_{max}（默认值为 6）

输出：将 D 划分为 C_1，C_2，各簇的中心 $\boldsymbol{\mu}_1$，$\boldsymbol{\mu}_2$，以及各属性的权重构成的向量 $\mathbf{w}$

1. 使用式(4)计算得到权重向量 $\mathbf{w}$
2. 在 $\mathbf{w}$ 中找到最大权重存入变量 w_{max}，剔除权值小于 $\beta.w_{max}$ 的属性，$\beta\in[0,1]$，默认值为 0.2
3. 使用具有最大和最小输出值的两个样本作为初始簇心 $\boldsymbol{\mu}_1$，$\boldsymbol{\mu}_2$
4. For 1 to I_{max} Do
5. 依照加权距离最近原则划分样本形成 C_1，C_2
6. 使用两簇各个属性上的均值重新计算 $\boldsymbol{\mu}_1$，$\boldsymbol{\mu}_2$
7. If 各簇心 $\boldsymbol{\mu}_1$，$\boldsymbol{\mu}_2$ 与前一轮相比无明显变化
8. Then Break For
9. End For
10. Return C_1，C_2，$\boldsymbol{\mu}_1$，$\boldsymbol{\mu}_2$ 以及 $\mathbf{w}$

3.2 分类型属性

3.1 节所阐述的结点划分方法可以直接应用在数值型属性上，若要对分类型属性采用类似的方法，则需要解决如何为分类型属性加权以及如何联合多个分类型属性进行聚类两个问题。

皮尔逊相关系数无法应用在分类型属性之上，这里本文采用与经典回归树进行属性选择类似的方法来为分类型属性加权。权重的计算式如式(5)所示：

$$w_l=1-\frac{\sum_{i=1}^{d}(|D_i|\cdot MSE(D_i))}{|D|\cdot MSE(D)} \tag{5}$$

其中，假设数据集 D 中样本在第 l 个分类型属性上共有 d 种不相同的取值，D_i 表示 D 的一个子集，其中包含的样本在该属性上取值相同。实际上，本文为分类型属性所使用的权重就是按照该属性的属性值分割样本集后，样本子集相对于原始样本集在输出值均方误差方面下降的比率。

如果要联合多个分类型属性为样本聚类，则需要定义分类型属性的簇心表示方法，以及样本到簇心距离的计算方法。k-modes 算法是 k-means 在分类型属性上的变体，它使用分类型属性的众数代替数值型属性的均值，计算样本到簇中心的距离时，分类属性上采用相同值取 0、不同值取 1 的方式。然而，这种方式计算出的距离不够精确，并且当某个属性出现多个众数时，不同的众数选择可能得到完全相反的结果。下面使用一个例子来说明这个问题：假设由 A 和 B 两个分类型属性描述的两个簇 C_1 和 C_2 都包含 10 个样本，样本的统计分布如表 1 所列。

表 1 属性值分布

Table 1 Attribute value distribution

	A	B
C_1	$a_1:9, a_2:1$	$b_1:4, b_2:4, b_3:2$
C_2	$a_1:4, a_2:3, a_3:3$	$b_1:4, b_3:4, b_4:2$

簇 C_1 中，90% 的样本在 A 上的值为 a_1，而 C_2 中，只有 40% 的样本在 A 上的值为 a_1，a_1 在两个簇中都是众数，这使得在判断某个样本距离 C_1 还是 C_2 更近时，属性 A 没有任何作用。属性 B 在 C_1 和 C_2 中分别存在两个众数。假设存在样本 $\boldsymbol{q}=(a_1,b_1)$，如果为 C_1 选择簇心 $\boldsymbol{\mu}_1=(a_1,b_1)$，为 C_2 选择簇心 $\boldsymbol{\mu}_2=(a_1,b_3)$，得到 $\boldsymbol{q}$ 与 $\boldsymbol{\mu}_1$ 的距离为 0，$\boldsymbol{q}$ 与 $\boldsymbol{\mu}_2$ 的距离为 1，得到 $\boldsymbol{q}$ 属于 C_1 的结论；如果为 C_1 选择簇心 $\boldsymbol{\mu}_1=(a_1,b_2)$，为 C_2 选择簇心 $\boldsymbol{\mu}_2=(a_1,b_1)$，得到 $\boldsymbol{q}$ 与 $\boldsymbol{\mu}_1$ 的距离为 1，q 与 $\boldsymbol{\mu}_2$ 的距离为 0，则得到 $\boldsymbol{q}$ 属于 C_2 的结论。

针对使用各个属性的众数表示簇心带来的计算距离不够精确、存在二义性等问题，本文使用每个分类属性值概率分布

的估计来定义簇心的表示方法及样本到簇心距离的计算方法，以供结点划分时使用。

定义 1 设包含 p 个分类型属性，n 个样本的数据集 D 分为 2 个簇，簇 C_j 中第 l 个属性 A_l 有 d 个不同取值 $\omega_1,\omega_2,\cdots,\omega_d$，令 C_{j,x_l} 表示第 l 个属性上取值为 x_l 的样本子集，$x_l\in\{\omega_1,\omega_2,\cdots,\omega_d\}$ 的条件概率被估计为：

$$P(x_l|j)=\frac{|C_{j,x_l}|}{|C_j|} \tag{6}$$

定义 $S_{j,l}$ 为 C_j 中关于 A_l 的各个属性值的汇总。

$$S_{j,l}=\{P(\omega_1|j),\ P(\omega_2|j),\cdots,P(\omega_d|j)\} \tag{7}$$

定义 2 使用 $S_{j,l}$ 代替众数，定义簇 C_j 的中心 $\boldsymbol{\mu}_j$ 为由各个属性的 $S_{j,l}$ 组成的向量。

$$\boldsymbol{\mu}_j=(S_{j,1},S_{j,2},\cdots,S_{j,p}) \tag{8}$$

定义 3 定义 $diff(A_l,\omega,S_{j,l})$ 为属性值 ω 与 $S_{j,l}$ 的距离。

$$diff(A_l,\omega,S_{j,l})=\begin{cases}1-P(\omega|j), & \text{if } \omega\in\{\omega_1,\omega_2,\cdots,\omega_d\}\\ 1, & \text{others}\end{cases} \tag{9}$$

定义 4 在已知属性权重向量 $\boldsymbol{w}$ 的情况下，样本 $\boldsymbol{x}_i$ 与簇心 $\boldsymbol{\mu}_j$ 的加权距离为：

$$dis\_c(\boldsymbol{x}_i,\boldsymbol{\mu}_j)=\sum_{l=1}^{p}(w_l\cdot diff(A_l,x_{i,l},S_{j,l})) \tag{10}$$

在之前的例子中，假设所有属性的权重均为 1，依据式(10)，样本 $\boldsymbol{q}=(a_1,b_1)$ 与表 1 中 C_1 和 C_2 的中心距离分别为 0.7=0.1+0.6 和 1.2=0.6+0.6，说明 $\boldsymbol{q}$ 与簇 C_1 更接近。

对于只包含分类型属性的数据，我们仅仅是将算法 1 的步骤 1 中计算权重的公式换成式(5)，第 3 和第 6 步中计算簇心的公式由计算均值改为计算式(8)，第 5 步中计算样本到簇心距离的计算公式由式(10)代替式(3)。对于混合类型的数据，簇心向量由两部分构成，一部分是数值属性的均值，一部分是式(7)所示的汇总信息。计算样本到簇心距离时，使用式(3)计算数值属性部分 dis_n，使用式(10)计算分类属性部分 dis_c，再使用式(11)将两部分结合。

$$dis(\boldsymbol{x}_i,\boldsymbol{\mu}_j)=(1-\gamma)\cdot dis\_n(\boldsymbol{x}_i,\boldsymbol{\mu}_j)+\gamma\cdot dis\_c(\boldsymbol{x}_i,\boldsymbol{\mu}_j) \tag{11}$$

其中，$dis\_n(\boldsymbol{x}_i,\boldsymbol{\mu}_j)$ 和 $dis\_c(\boldsymbol{x}_i,\boldsymbol{\mu}_j)$ 分别代表数值属性子集和分类属性子集的距离，γ 是 0 到 1 之间的一个实数，用于调整公式中两项的占比。由于我们为数值型属性和分类型属性设计了不同的加权方式，尽管两种权重的取值范围相同，但它们之间不存在可比性。例如，当某个数值型属性的权值与某个分类型属性的权值相等时，并不意味着它们在计算距离时的重要性也相同。为此，我们在确定系数 γ 时，以输出值均方误差下降率为评价指标，在 $\{0.1,0.2,\cdots,0.9\}$ 中选择使输出值均方误差下降最大的值。

文献[20]将本小节的方法应用于分类问题，在分类型数据及混合型数据上均取得了较好的分类效果。

3.3 划分算法的时间复杂度

本文提出的划分方法分为属性加权和聚类划分两个阶段。在属性加权阶段，为了给数值型和分类型属性加权，分别需要计算式(4)和式(5)，计算这两个公式的时间复杂度均是结点包含样本数量的线性函数。因此，加权阶段的时间复杂度为 $O(n\cdot p)$，n 是当前结点包含样本的数量，p 为属性的个数。在聚类划分阶段的每次迭代中，需要为每个样本分簇并顺带更新两个簇心的位置，这需要计算样本到簇心的距离，时间复杂度为 $O(n\cdot p)$。聚类迭代次数最多为 $I_{\max}$，因此聚类划分阶段在最坏情况下的时间复杂度为 $O(n\cdot p\cdot I_{\max})$。综合属性加权和聚类划分两个阶段，本文提出的划分方法的时间复杂度为 $O(n\cdot p\cdot I_{\max})$。

经典的轴平行划分算法，如 CART，划分结点的时间代价主要来自于属性选择。当候选属性集中存在 p_n 个数值型属性时，为了选择划分点，需要依据每个属性为样本排序，处理数值型属性的时间复杂度为 $O(p_n\cdot n\cdot lbn)$；当候选属性集中存在 p_c 个分类型属性时，简单地假设每个属性都有 d 个属性值，为了进行二路划分，需要为每个属性尝试 $2^{d-1}-1$ 种不同的属性值组合，那么处理分类型属性的时间复杂度为 $O(p_c\cdot n\cdot 2^d)$。两部分综合在一起的时间复杂度为 $O((p_n\cdot lbn+p_c\cdot 2^d)\cdot n)$。

实验中，本文将 $I_{\max}$ 设置为 6，因此当样本数量 n 较大或者分类型属性的属性值个数 d 较大时，本文的划分方法在时间代价上优于经典的 CART 算法。但这并不能说明本文的划分方法比轴平行划分在时间方面有绝对的优势，原因有两个方面：1)面对数值属性，轴平行方法主要进行关系运算，而本文方法主要进行浮点数的乘法甚至开方运算；2)面对分类型属性，二路划分并不是轴平行方法唯一的选择，例如可将分类型属性转换为数值型，或者同 C4.5 算法一样根据属性值个数进行多路划分。

综上，本文方法在时间复杂度方面与轴平行方法相近，但优于大部分多变量划分方法。

3.4 回归树的构建及集成

本文提出的回归树采用自顶向下的方式递归生长，具体构建过程如算法 2 所示。

算法 2 create_tree

输入：样本子集 D，最小父母数 minparent(默认值为 5)，最小均方差比例 minradio(默认值为 0.05)

输出：回归树

1. 使用 D 中的样本创建一个结点 node，计算样本输出值方差 yv，将 yv * minradio 存入变量 minyv。
2. Procedure grow(node)
3. If node 中样本的输出值均方误差小于 minyv 或者样本数小于 minparent Then
4. 标记 node 为叶结点，并将样本输出值的均值作为该结点的输出值。
5. Return node
6. End If
7. 随机产生属性子集 F，调用 split 函数划分结点，获得 C_1，C_2，$\boldsymbol{\mu}_1$，$\boldsymbol{\mu}_2$，以及 $\mathbf{w}$。
8. 保存 F，$\boldsymbol{\mu}_1$，$\boldsymbol{\mu}_2$ 和 $\mathbf{w}$ 到当前结点之中，供未来预测新样本时使用。
9. 使用 C_1，C_2 中的样本分别创建 $node_1$，$node_2$ 后调用 grow($node_1$) 和 grow($node_2$)。
10. End Procedure
11. Return 以 node 为根的回归树。

本文的集成方法有两个层次，内层先使用算法 2 创建的

回归树作为基学习器套用随机森林框架生成回归随机森林，外层使用内层产生的回归随机森林作为梯度提升集成方案的基学习器，创建最终的回归模型(CRF)。实际上，CRF 是一个回归树集成器的集成器。算法 3 和算法 4 分别展示这两个集成方法的流程。

算法 3 create_forest

输入：训练集 D，个体的数量 n_t(默认值为 20)

输出：创建的森林

1. 创建空森林 FO
2. For i=1 to n_t Do
3. 从 D 中使用 bootstrap 方法采样获得样本子集 D_i
4. 使用 D_i 调用 create_tree 获得 T_i，加入 FO。
5. End For
6. Return FO

算法 4 create_CRF

输入：训练集 D，个体的数量上限 n_f(默认值为 5)

输出：创建的 CRF

1. 创建空 CRF
2. For i=1 to n_f Do
3. 使用当前的 D 调用 create_forest 函数创建森林 FO_i，加入 CRF
4. 使用 FO_i 预测 D 中所有样本的输出值，并将样本原有输出值与预测值的差作为该样本新的输出值，供下一轮训练使用
5. If 当前 D 中的所有样本预测值都为 0
6. Then Break For
7. End If
8. End For
9. Return CRF

算法 3 生成的回归随机森林在预测未来数据的输出值时，使用森林中所有个体树预测值的平均值。而算法 4 生成的 CRF 在预测未来数据的输出值时，使用集成器中各个个体森林预测值的和。

4 实验

本文使用 C++语言实现了提出的回归模型 CRF，并将其与 sklearn 框架中常用的线性回归(LR)、支持向量回归(SVR)、最近邻(KNN)、CART、随机森林(RF)、Bagging、Adaboost、GBDT 以及文献[21]中的 NBFRTree 9 种回归模型在 12 个回归任务上进行了实验对比。NBFRTree 是一种多重划分的回归树，在可用划分属性较多时能够达到集成学习器的回归效果。实验使用的计算机配置为 Intel Core i7-6500U (2.6 GHz)16 GB 内存，Python 版本为 3.7.1，C++编译使用 TDM-GCC 4.9.2。

4.1 数据集和实验设置

表 2 列出了实验使用的 12 个数据集，这些数据集源于 UCI[22]，“属性数”一列以“数值型属性数+分类型属性数”的形式给出。

表 2 数据集

Table 2 Data sets

序号	数据集	样本数	属性数
1	abalone	4177	7+1
2	airfoil	1503	5+0
3	ccpp	9568	4+0
4	concrete	1030	8+0
5	flare	1389	0+10
6	forestfire	517	10+2
7	mpg	398	4+3
8	redwine	1599	11+0
9	servo	167	2+2
10	student	649	17+13
11	whitewine	4898	11+0
12	yacht	308	6+0

实验中，本文将 CRF 的所有参数均设置为前文算法描述中提及的默认值，例如算法 1 的聚类最大迭代次数为 6，算法 2 的最小父母数为 5，算法 3 的个体树数量为 20，算法 4 的最大个体数为 5 等。通过前期的一些实验发现，为 CRF 设置的这些参数默认值可以在所有数据集上获得最佳或接近最佳的回归表现。

为了保障公平性，我们将对比的 9 个回归模型中的 4 个集成器(RF，Bagging，Adaboost 和 GBDT)的个体数量也设置为 100(CRF 中回归树的数量是 20 * 5)。此外，LR、SVR 等模型不支持分类型属性，当这些模型处理带有分类型属性的数据时，我们使用 OneHot Encoder 技术对数据进行数值化转换。

4.2 与其他回归模型进行对比

实验中，本文对每个数据集都采用了 10 次 10 折交叉验证的方法，并且只报告测试集上的平均实验结果。表 3 和表 4 分别列出了 10 种算法在 12 个数据集上的平均绝对误差(MAE)和均方根误差(RMSE)，两个表格的最后两行分别是平均结果(平均 MAE 或平均 RMSE)和平均名次(平均名次是 10 种算法在每个数据集上表现排名的平均值)。另外，两个表格中每一行上的最好结果被加粗显示，由于结果只保留了两位小数，因此原本存在微小差异的结果看起来完全相同。

表 3 对比不同算法的 *MAE*

Table 3 Comparison of *MAE* for different algorithms

数据集	CRF	LR	SVR	KNN	CART	RF	Adaboost	Bagging	GBDT	NBFRTree
1	**1.49**	1.59	1.61	1.58	2.07	1.55	2.46	1.55	1.52	1.56
2	1.21	3.74	5.27	4.82	1.74	**1.19**	3.16	1.19	1.98	1.53
3	2.35	3.63	9.98	2.86	3.01	2.31	**4.47**	**2.31**	2.95	3.13
4	3.87	8.31	13.13	6.75	3.88	3.27	6.33	**3.27**	3.74	3.95
5	0.38	—	**0.33**	0.40	0.40	0.40	0.56	0.40	0.38	0.39
6	19.27	21.73	**12.81**	19.50	23.41	22.15	29.27	21.88	21.66	18.96
7	**1.99**	2.26	6.48	3.18	2.93	2.23	2.87	2.23	2.19	2.47
8	**0.38**	0.50	0.51	0.57	0.45	0.41	0.51	0.41	0.48	0.40
9	0.23	0.91	0.60	0.62	0.24	0.22	0.39	**0.22**	0.23	0.27
10	**1.97**	2.01	2.01	2.26	2.75	2.00	2.18	2.00	2.06	2.07
11	**0.37**	0.59	0.53	0.62	0.47	0.42	0.59	0.42	0.54	0.43
12	0.74	7.34	9.18	5.17	0.63	0.50	1.16	0.50	**0.39**	0.56
平均	**2.85**	4.78	5.20	4.03	3.50	3.05	4.49	3.03	3.18	2.98
名次	**2.50**	7.67	7.17	7.58	6.67	3.17	8.25	3.17	4.08	4.75

表 4 对比不同算法的 *RMSE*

Table 4 Comparison of *RMSE* for different algorithms

数据集	CRF	LR	SVR	KNN	CART	RF	Adaboost	Bagging	GBDT	NBFRTree
1	**2.15**	2.21	2.42	2.24	3.01	2.18	2.89	2.18	2.16	2.22
2	1.68	4.81	6.56	6.07	2.42	**1.66**	3.87	1.67	2.63	2.39
3	3.41	4.56	12.68	3.92	4.43	3.27	5.62	**3.27**	3.87	4.58
4	5.54	10.47	16.44	8.88	6.05	4.76	7.67	**4.76**	5.08	5.82
5	**0.72**	—	0.75	0.77	0.86	0.79	0.81	0.79	0.73	0.77
6	48.32	48.58	**46.59**	52.73	71.91	55.93	60.69	55.52	57.82	47.78
7	**2.86**	2.96	7.75	4.26	4.26	3.18	3.77	3.20	3.11	3.55
8	0.59	0.65	0.68	0.75	0.78	**0.57**	0.63	0.57	0.62	0.63
9	0.40	1.09	1.05	0.87	0.44	0.40	0.54	0.40	**0.38**	0.48
10	2.73	2.75	2.79	3.10	3.77	**2.71**	2.84	**2.71**	2.79	2.82
11	**0.59**	0.75	0.73	0.80	0.82	0.60	0.74	0.60	0.69	0.66
12	1.41	9.03	16.21	10.75	1.23	1.00	1.54	0.99	**0.71**	0.92
平均	**5.87**	7.99	9.56	7.93	8.33	6.42	7.63	6.39	6.71	6.05
名次	**2.58**	6.83	7.58	7.58	7.83	3.25	7.58	3.08	3.67	5.00

数据集 flare 中的 10 个预测属性全部都属于分类型，经过 OneHot Encoder 数值化转换后的数据过于稀疏，导致 LR 无法获得一个收敛的解。因此，在表 3 和表 4 的相应单元格中我们以短横线显示。计算 LR 模型的平均 *MAE* 和平均 *RMSE* 时，仅考虑了其他 11 个数据集，计算 10 种算法在数据集 flare 上的排名时，将 LR 排在第 10 位。

通过表 3 可以看到，CRF 在 5 个数据集上获得了最低的 *MAE*，再观察全部 12 个数据集的平均 *MAE*，*CRF* 在 10 种算法中也是最优的。在表 4 展现的 *RMSE* 方面，*CRF* 在 4 个数据集上获得了最好的成绩，平均 *RMSE* 也是最低的。

为了进一步检验所比较的 10 种算法在 12 个数据集上表现的差异，Friedman 检验被使用。在 10 种算法、12 个数据集的情况下，F_F 服从自由度(9,99)的 F 分布，显著性 0.05 的临界值 $F_{9,99}=1.98$。本文使用表 3 和表 4 最后一行的平均名次计算获得的统计检验值分别为 12.09 和 12.13，均大于 1.98。为此，所有算法表现相同的假设被拒绝。本文继续使用 Nemenyi 的方法进行后续检验，计算得到名次临界区间 CD=3.83。因此，在 *MAE* 方面，CRF 与 LR，SVR，KNN，CART 以及 Adaboost 存在显著差异，RF 和 Bagging 与 LR，SVR，KNN 以及 Adaboost 存在显著差异，GBDT 与 Adaboost 存在显著差异；在 *RMSE* 方面，CRF 与 LR，SVR，KNN，CART 以及 Adaboost 存在显著差异，RF，Bagging 和 GBDT 与 SVR，KNN，CART 以及 Adaboost 存在显著差异。综合来看，CRF，RF，Bagging 和 GBDT 这 4 个集成回归器的表现突出，NBFRTree 的表现次之，而集成器 Adaboost 的表现与 4 个个体回归器相差不大。这就是本文为提出的划分方法、选择随机森林和梯度提升两个框架进行集成的原因。

4.3 消融实验

本文在 k-means 和 k-modes 算法的基础上提出了加权聚类划分方法。其改变有两点：1)使用加权属性计算距离；2)使用详细的样本属性值分布来表示分类型属性的“均值”。本小节将通过实验来验证这两个改变的作用。实验使用 abalone 等 6 个带有分类型属性的数据集。在其他参数不变的情况下，尝试另外 3 种不同的划分方法，并与 CRF 在 MAE 方面进行对比，表 5 列出了对比结果。表 5 中的第 2 列是 CRF 在 6 个数据集上取得的 *MAE*(保留 3 位小数)。第 3 列是 CRF 去掉属性加权后得到的结果，第 4 列是第 3 列相对于第 2 列上涨的百分比。从平均结果来看，在不对属性加权直接聚类的情况下，6 个数据集的平均 *MAE* 大约上涨 15.23%。第 5 列和第 6 列展示的是未使用 3.2 节描述的方法，而是采用类似于 k-modes 算法的方式来处理分类型属性所获得的回归结果，此时平均 *MAE* 大约上涨 7.2%。第 7 列和第 8 列则是既未对属性加权，又没有使用 3.2 节的方法来处理分类型属性的回归结果，可以看到平均 *MAE* 大约上涨了 18.78%。

表 5 对比不同划分方法的 *MAE*

Table 5 Comparison of the *MAE* for different split methods

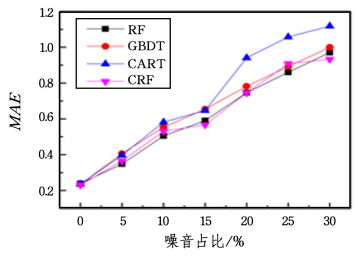
数据集	CRF	不加权		k-modes		不加权+k-modes	
		MAE	上涨%	*MAE*	上涨%	*MAE*	上涨%
abalone	1.488	1.613	8.33	1.600	7.46	1.615	8.46
flare	0.383	0.396	3.44	0.386	0.79	0.403	5.31
forestfire	19.274	21.624	12.19	22.658	17.55	22.787	18.22
mpg	1.994	2.048	2.70	2.067	3.68	2.158	8.24
servo	0.231	0.365	58.39	0.260	12.94	0.382	65.89
student	1.969	2.094	6.32	1.985	0.79	2.098	6.53
平均	4.223	4.690	15.23	4.826	7.20	4.907	18.78

4.4 鲁棒性测试

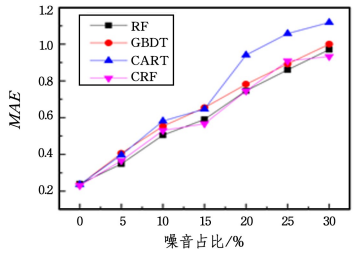
通常集成学习器比单一学习器具备更好的鲁棒性，而由于错误的不断积累，Boosting 方法生成的集成器的鲁棒性不如 Bagging 方法生成的集成器[13]。本节使用 servo 数据集对比 RF，GBDT，CART 以及 CRF 的抗噪能力。参与对比的回归器在 servo 数据集上的表现相近，取得的 MAE 都在 0.23～0.24 之间。另外，RF 属于 Bagging 的扩展，servo 数据集的属性仅有 4 个，两个模型在该数据集不同比例噪音下的表现几乎完全相同，为此本文仅展示了 RF 的相关数据。

实验分为两组，在第一组实验中，我们在输入属性上加入了 5～30%的噪音，加噪方法是随机地将属性值替换为与原值不等且又存在于数据集中的一个值。在第二组实验中，我们在输出属性上加入 5～30%的噪音，加噪方法是在输出属性的取值范围内以均匀分布方式产生随机值替换原值。

图 1 给出了 4 个回归模型在不同比例输入和输出噪音下，在 servo 数据集上取得的 MAE。可以看到，单树 CART 的抗噪能力最差，RF 最优，GBDT 界于两者之间。CRF 结合了 RF 和 GBDT 的集成策略，抗噪能力比 GBDT 稍强，但逊于完全依靠 Bagging 方法生成的 RF。



(a)输入噪音



(b)输出噪音

图 1 CRF 与其他算法的鲁棒性比较

Fig. 1 Robustness of CRF compared with other algorithms

结束语 回归是机器学习在实际应用领域中常见的任务。而在这些任务中,难免会出现带有分类型属性的数据。例如线性回归、SVR 以及部分多变量决策树等回归模型不能直接处理分类型属性,而分类型原始数据经过数值化转换后,可能会影响这些模型的表现。本文提出了一种可联合分类型属性的决策树结点划分方法,并将该划分方法产生的多变量回归树作为基学习器,应用随机森林和梯度提升框架进行了两级集成。实验结果表明,本文创建的回归集成器 CRF 在纯数值型数据、纯分类型数据以及混合型数据上都有不错的表现。

本文提出的划分方法以聚类算法 k-means 为基础,由于 k-means 本身易受离群点的影响,导致本文方法不适用于带有离群点的数据。离群点的存在,将使得本文算法计算得到的簇心发生严重偏离,导致生成决策树的结点过多,以至泛化能力下降。虽然使用中位数代替均值作为簇心可以降低离群点对划分效果的影响,但划分操作的时间代价会随之增加。在未来的工作中,我们将在数据预处理、离群数据识别以及使用其他聚类算法划分结点等方面进行探索,以进一步完善提出的回归森林。

参 考 文 献

[1] PAN J H, WANG Y H, WU W. Physical quantity regression method based on optimized BP neural network[J]. Computer Science, 2018, 45(12): 170-176.

[2] CHEN W, LI H, HOU E K, et al. GIS-based groundwater potential analysis using novel ensemble weights-of-evidence with logistic regression and functional tree models[J]. Science of the Total Environment, 2018, 634(9): 853-867.

[3] WANG N, LI Z, CHENG X Y. Reversible visible watermarking algorithm for medical image based on support vector regression [J]. Computer Science, 2019, 34(10): 2243-2248.

[4] LOH W Y, SHIH Y S. Split selection methods for classification trees[J]. Statistica Sinica, 1999, 7(4): 815-840.

[5] QUINLAN J R. C4. 5: programs for machine learning[J]. Machine Learning, 1994, 16(3): 235-240.

[6] BUNTINE W L. Learning classification trees[J]. Statistics & Computing, 1992, 2(2): 63-73.

[7] BUCY R S, DIESPOSTI R S. Decision tree design by simulated annealing[J]. ESAIM Mathematical Modelling and Numerical Analysis, 1993, 27(5): 515-534.

[8] MURTHY S K, KASIF S, SALZBERG S. A System for Induction of Oblique Decision Trees[J]. Journal of Artificial Intelligence Research, 1996, 2(1): 1-32.

[9] LÓPEZ-CHAU A, CERVANTES J, LÓPEZ-GARCÍA L, et al. Fisher's decision tree[J]. Expert Systems with Applications, 2013, 40(16): 6283-6291.

[10] HONG K S, OOI P L, YE C K, et al. Multivariate alternating decision trees[J]. Pattern Recognition, 2016, 50(C): 195-209.

[11] WICKRAMARACHCHI D C, ROBERTSON B L, REALE M, et al. HHCART: An Oblique Decision Tree[J]. Computational Statistics & Data Analysis, 2015, 96: 12-23.

[12] BJOERN H M, KELM B M, DANIEL N S, et al. On Oblique Random Forests[C]// Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 2011: 453-469.

[13] BREIMAN L. Random Forests [J]. Machine Learning, 2001, 45(1): 5-32.

[14] BREIMAN L. Bagging predictors[J]. Machine Learning, 1996, 24(2): 123-140.

[15] HO T K. The random subspace method for constructing decision forests[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8): 832-844.

[16] FREUND Y, SCHAPIRE R. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of Computing System, 1997, 55: 119-139.

[17] FRIEDMAN J H. Greedy function approximation: a gradient boosting machine[J]. The Annals of Statistics, 2001, 29(5): 1189-1232.

[18] WANG X H, ZHANG L, LI J Q, et al. Study on XGBoost Improved Method Based on Genetic Algorithm and Random Forest [J]. Computer Science, 2020, 47(S2): 454-458.

[19] QU W L, CHEN X Y, LI Y Y, et al. A regression prediction model of depth gradient boosting[J]. Computer Applications and Software, 2020, 37(9): 194-201.

[20] LIU Z Y, SONG X Y. An applicable multivariate decision tree algorithm for categorical attribute data[J]. Journal of Northeastern University (Natural Science), 2020, 41(11): 1521-1527.

[21] GENRIKHOV I E, DJUKOVA E V, ZHURAVLEV V I. On full regression decision trees[J]. Pattern Recognition and Image Analysis, 2017, 27(1): 1-7.

[22] LICHMAN M. UCI machine learning repository [EB/OL]. (2019-09-23) [2019-10-11]. http://archive.ics.uci.edu/ml/index.php.

LIU Zhen-yu, born in 1978, postgraduate, professor. His main research interests include machine learning and artificial intelligence.

(责任编辑:李亚辉)