中国科学: 信息科学

SCIENTIA SINICA Informationis

论文

缓解随机一致性的基尼指数与决策树方法

王婕婷[1], 李飞江[1], 李珏[1], 钱宇华[1,2*], 梁吉业[2]

1. 山西大学大数据科学与产业研究院, 太原030006
2. 山西大学计算智能与中文信息处理教育部重点实验室, 太原030006
* 通信作者. E-mail: jinchengqyh@sxu.edu.cn

科技创新2030-重大项目(批准号: No.2021ZD0112400)、国家自然科学基金重点项目(批准号: 62136005)、国家自然科学基金青年基金(批准号:62106132)、山西省科技重大专项(批准号:202201020101006)、山西省基础研究计划(批准号: 20210302124271, 202103021223026).

摘要 决策树模型具有较强的可解释性, 是随机森林、深度森林等机器学习方法的基础. 如何选择节点的分割属性与分割值是决策树算法的关键问题, 对树的泛化能力、深度、平衡程度等重要性能产生影响. 传统属性选择准则的定义大多基于凹函数, 使得决策树算法存在多值偏向问题, 即倾向于选择取值种类多的属性作为节点分割属性. 已有研究表明缓解随机一致性的评价准则能够降低分类偏差与类簇个数偏向. 本文将基于标准化框架缓解基尼指数的随机一致性, 以此缓解其多值偏向问题. 通过人造数据集验证, 标准基尼指数能够缓解基尼指数的多值偏向问题, 并且选择出具有决策信息的属性. 通过十二个基准数据集与两个图像数据集的实验验证, 基于标准基尼指数的决策树算法比现有缓解多值偏向的决策树算法具有较高的泛化性能.

关键词 基尼指数, 多值偏向, 决策树, 随机一致性

1 引言

决策树方法是经典的机器学习方法[1], 该方法采用自上而下的分层策略对样本进行划分, 每一次划分按照选取的属性将特征空间分割成不同的区域, 目标是使得同一区域内的样本属于同一类. 决策树方法具有诸多优点, 如模型能够生成直观可理解的If-Then 规则, 具有较强的可解释性, 并且适用于处理兼具数值属性与语义属性的分类问题. 此外, 由于决策树的内部节点每次选择有利于当前样本划分的属性, 因此可以区分不同类别所需的特征, 并且树内部节点单独计算每个特征的重要度, 对高维数据带来的维数灾难问题具有一定的抵御能力. 基于决策树模型, 研究者提出随机森林、深度森林等泛化性能更高的学习方法, 在大规模数据、多模态数据、非平衡数据上取得了成功的应用.

如何选择节点的分割属性与分割值是决策树方法的核心问题. 现有方法大多使用基尼指数等不纯度降低函数作为节点选择属性准则. 这些不纯度函数实质上是某个单峰凹函数的和的负值, 由于两个变量线性组合的凹函数值大于两个变量凹函数值的线性组合($f(tx_1+(1-t)x_2) \geqslant tf(x_1)+(1-t)f(x_2)$),

引用格式: 王婕婷, 李飞江等. 缓解随机一致性的基尼指数与决策树方法. 中国科学: 信息科学, 在审文章
Wang J T, Li F J, et al. Gini Index and decision tree method with mitigating random consistency (in Chinese). Sci Sin Inform, for review

这使得不纯度降低函数存在多值偏向问题, 即倾向于选择取值个数多的属性. 一般而言, 均匀分布的随机变量或受噪音干扰的低质变量会存在较多的取值. 多值偏向问题导致决策树算法在属性选择时存在偏差, 选择出取值种类较多而对类别决定程度不高的属性变量, 这将会影响树模型的泛化能力与噪音鲁棒性等性能.

属性重要度可通过特征属性与决策属性之间的一致程度刻画. 在一致程度刻画中, 存在部分由变量随机性或分布特性导致的一致性. 例如, 在考试成绩评价中, 选择题的正确答案为真实标签, 学生答题结果为预测标签. 假设十道题的答案为(A, B, A, C, C, C, C, D, A, B), 学生随机猜测的答案为(C, C, C, C, C, C, C, C, C, C)与(A, B, A, B, A, B, A, B, A, B). 这样的答案可获得百分之四十与百分之五十的准确率. 这两种随机猜测的答案分布与真实答案的分布较为近似, 从而产生了较高的一致性. 本文将这种由于变量分布或随机性导致的一致性称为随机一致性.

在聚类性能评价中, 兰德指数与信息增益被验证具有类簇不鲁棒性, 即评价值随着类簇个数的增多而明显增大, 显然, 基尼指数的多值偏向问题与类簇评价准则的类簇不鲁棒性为同一个问题. 即两变量的一致性刻画过程中, 一致性度量值受到随机变量取值个数的影响. 如前所述, 这样的由于变量取值或分布影响一致性的问题可通过缓解随机一致性来解决. Gates等人[2, 3]已验证消除随机一致性的兰德指数与信息增益具有较高的类簇个数鲁棒性, 即评价值不随着类簇个数的增多而明显增多.

综上, 本文将通过缓解基尼指数的随机一致性来缓解其多值偏向问题, 以提升基尼指数选择特征的质量及决策树算法的泛化性能. 本文贡献主要体现在以下三方面:

(1) 证明置换集合中特征向量与标签向量形成的列联表元素服从超几何分布, 并在超几何分布下给出基尼指数期望与方差的表达式;

(2) 定义缓解多值偏向的标准基尼指数, 在人造数据集上验证其特征选择的有效性;

(3) 提出基于标准基尼指数的决策树算法, 并通过实验验证其泛化性能.

本文在第2节给出一类凹函数并从理论上分析其多值偏向问题. 在第3节给出置换集合标准基尼指数的期望与方差, 并定义标准基尼指数. 第4节在人造数据集上验证标准基尼指数缓解多值偏向的有效性, 第5节在基准数据集与图像数据集上验证基于标准基尼指数的决策树算法的泛化性能. 最后总结全文.

1.1 相关工作

如前所述, 决策树算法中的特征重要度评价准则中存在多值偏向问题. 已有研究成果表明消除随机一致性的方法可缓解两变量一致性评价的多值偏向问题[2]. 因此, 本文将基于缓解随机一致性的方法缓解基尼指数的多值偏向问题, 这包括决策树算法、决策树重要度评价准则, 多值偏向问题, 随机一致性四个方面的工作, 本节将从上述四个方面进行回顾.

1.1.1 决策树算法

决策树算法是统计机器学习领域的代表性算法, 自提出以来, 关于决策树算法的剪枝、集成、增量式学习、模型深度化等策略相继被提出, 决策树模型在代价敏感学习、非平衡问题、公平性学习等场景下被广泛使用[4]. 决策树学习算法的典型代表算法为ID3[5], C4.5[6]和Cart算法[7]. 这些算法的区别在于树结构与特征选择标准. ID3与C4.5采用多叉树结构, 为离散属性的每个取值建立一个分枝, 二者分别使用信息增益与信息增益比选择节点特征. Cart算法构建二叉树, 通过切割阈值将连续属性二值化, 使用基尼指数评价特征重要的程度.

为适应不同的任务类型与数据特性, 较为高阶的决策树算法相继被提出. 为了处理有序分类问题,

Hu等人[8]提出有序信息熵及相应的决策树算法, 该算法表现出较好的泛化性与鲁棒性. 为了处理非平衡数据, Demirovic等人[9]提出优化F_1分数等非线性评价指标的决策树方法. 为了提升决策树模型的类别公平性, Aghaei[10]等人基于混合整数线性规划的正则化方法来权衡决策树中的预测质量和公平性. 此外, 决策树模型具有较强的可扩展性. 容易与线性模型[11]、深度学习[12, 13]等其他模型结合, 生成较为复杂的决策树模型, 用于处理非线性数据、非结构化数据等复杂数据类型.

决策树模型的不足之处在于决策树算法对训练数据具有较强的依赖性, 导致决策树模型具有不稳定性, 容易对训练数据产生过拟合的现象. 当训练数据发生微小改变时, 决策树的树模型产生较大的变化. 决策树模型的不稳定性限制了其泛化性能. 这一问题可通过Bagging[14], 随机森林[15, 16], 正交旋转森林[17], Boosting[18], 特征子空间构建[19]等集成方法进行改善. 这些方法通过对原始数据集的样本进行抽样、赋权, 或对特征进行扰动、约简、加权组合等方式得到多个训练数据集, 在这些训练数据集上得到的多个决策树模型具有一定的互补性、差异性, 通过集成这些模型提升对新数据的泛化性.

可见, 作为经典的机器学习算法, 决策树模型具有较强的任务适应性与模型可扩展性, 并且作为具有较强可解释性的方法, 取得了较广泛的关注与应用.

1.1.2 特征重要度评价准则

对于特征重要度评价准则的研究, 主要从评估指标的定义、抽样分布估计及准则最优化特性三个方面进行回顾. 在评估指标定义方面, Giorgi等人[20]列出多种离散情形下基尼指数的定义, 并给出基于基尼指数的统计推断方法. 基于不同网格划分所得信息熵的最大值, Reshef等人[21]提出MIC评价指标, 可用来探索连续变量的非线性关系. Serrurier等人[22]基于概率置信区间上界定义了信息熵评价准则, 该准则与早停机制或事后剪枝效果相当. Li等人[23]基于信息熵定义了样本稳定性函数, 用于区分稳定的类簇核心样本与不稳定的边缘样本, 并提出相应的聚类方法.

抽样分布估计是指评价准则的经验估计与真实值之间的差距以及评价准则的概率分布形式的研究. Bhargava等人[24]在多项分布下给出基尼指数抽样分布的概率值表, 并给出其期望与方差的计算公式. Roulston等人[25]与Ramos等人[26]分别基于二阶与高阶泰勒展式给出观测信息熵与真实信息熵之间的差距, 并给出观测信息熵方差的一种估计方式.

准则优化特性是指属性评价准则倾向于选择何种特性的属性及分割点. Raileanu等人[27]从理论上比较了基尼指数与信息增益两种评价准则, 形式化地列出二者评判结果一致与不一致的情况, 并得出大多数情形下二者评判是一致的结论. Shih等人[28]将信息熵、基尼指数、卡方统计量等统计量统一表示为离散概率分布幂散度加权和的形式, 得到大多数准则(除基尼指数外)满足互斥优先性, 即优先选择可使左右节点互斥的属性. Breiman等人[29]理论验证了基尼指数倾向于将最大类放到一个节点, 其余类一个节点的分割, 信息熵倾向于节点大小平衡的分割.

综上所述, 决策树作为一种具有较强可解释性的分类算法, 其属性评价准则的经验估计性质, 优化特性及概率分布形式的研究仍具有较大的发展空间.

1.1.3 多值偏向问题及其缓解方法

为了缓解特征属性的多值偏向问题, 研究者提出引入特征属性或决策属性的置换集合来衡量多取值带来的重要度. 置换集合是指在某种特定约束条件下, 通过改变属性在样本上的排列位置形成新的属性向量. 这些属性向量由于特定约束与待评估属性向量形成某种意义下的共性向量, 然后这些属性向量及其评估值分别形成某种分布, 使用分布下的某个数字特征来衡量多取值带来的重要度.

通常, 属性向量常用的分布有均匀分布, 评估值常用的分布有伽马分布、高斯分布, 属性向量与决

策属性交叉表的元素常用的分布有超几何分布等. 现有方法大多采用置换集合上重要度的分布p值进行决策或从原始重要度中减去置换集合的平均重要度的策略. 这两种策略均需要求出重要度的均值, p值决策的策略进一步需要重要度的方差与抽样分布函数. 对于评估准则分布参数(均值与方差)与分布函数的计算, 现阶段存在着蒙特卡洛方法模拟计算、近似分布替代、精确计算的方法.

在蒙特卡洛方法模拟计算方面, Sandri等人[30]通过一组不包含决策信息的伪变量的平均重要度来近似未知偏差, 该方法与纯一致性度量思想相同, 不同之处在于该方法定义的消除偏差的重要度函数不包含归一化项, 且伪变量是满足条件的部分变量而非所有可能变量. Wright等人[31]通过置换特征属性在样本上排列的方式来定义由于属性结构(偏向)引起的重要程度, 并以减去这部分重要度的结果作为真正的重要度. Nembrini等人[32]采用评估准则的镜面经验累积分布估计p值, 该方法不适用于取值非负的重要度评价准则. Romano等人[33]基于自助采样法定义MIC评价指标的均值, 基于此定义缓解随机一致性的AMIC评价指标, 实验分析表明AMIC赋予均匀独立的随机变量对相应的零值.

在近似分布替代方面, Alin等人[34]在列联表元素服从多项分布的假设下计算基尼指数的期望与方差, 接着使用伽马分布近似基尼指数的抽样分布, 最后使用伽马分布的的p值作为重要度评价准则. 然而, 在分类问题中且假设特征的边缘分布固定时, 从理论上来看列联表元素应服从超几何分布. 此外, 使用p值进行重要度判断存在混淆有信息属性与无信息属性的风险, 且需要较多的样本拟合重要度准则的概率分布. Altmann等人[36]提出PIMP算法, 该算法对决策变量进行置换排列, 得到特征的多个重要程度值, 然后假设这些值服从高斯分布、对数高斯或伽马分布等, 分布的参数通过最大似然估计, 分布的选择通过Kolmogorov – Smirnov检验完成, 最终通过p值的对数变换对属性的重要程度进行判断.

在精确计算方面, Romano等人[35]在文献 [33]的基础上定义了更先进的标准互信息指标, 该指标通过从互信息中减掉均值除以标准差, 验证了这种定义方法可以降低多值偏向和小样本偏向, 并给出其使用条件为样本个数小于5倍的列联表元素个数. 然而, 这种定义方法涉及到计算评价指标的方差, 计算复杂度较高. Romano等人[33] 基于Alin等人[34]得到的基尼指数的抽样分布与Cantelli不等式给出基尼指数的分位数的上界, 并基于此构建随机森林, 实验表明该评价准则可获得更高的准确度.

综上所述, 基于蒙特卡洛方法与近似分布替代方面的工作大多基于评估准则的p值定义属性重要度. 而基于p值判断重要度的方法需要近似统计量(评估准则)的抽样分布及确定分布参数, 复杂度较高, 而且p值随着统计量的变换存在陡然变化的可能, 可能影响对属性信息程度的判断[37]. 基于精确计算的方法需要确定统计量的概率分布. 对于基尼指数而言, 其置换集合的分布应为超几何分布而非多项分布, 本文在超几何分布下定义标准基尼指数, 以缓解基尼指数引起的多值偏向问题, 以期构建具有更好泛化性的决策树模型.

1.1.4 随机一致性

在机器学习任务中, 样本相似度的计算、模型选择与评价等过程都需要对两个随机变量的一致性进行评判准则. 常用的一致性准则有基于距离的、基于统计相关系数的、基于信息论的. 这些准则大多直接计算向量之间的一致程度. 而由于变量的随机性、样本的有限性, 这些准则包含了随机一致性. 随机一致性是指由于随机变量分布的随机性导致的一致性.

在消除随机一致性方面, Wang等人[38]在纯一致性框架下缓解纯准确度指标中的随机一致性, 定义了纯准确度指标, 并理论验证其相较于准确度具有学习可替代性(优化纯准确度可得到较高的准确度值), 低偏差性(偏差是指两类错分率之间的绝对差值)、高辨识性(纯准确度指标可识别大部分被准确度赋予相同分数的基分类器对). Wang等人[39]对比了纯准确度值与Fmeasure值, 得出具有较高的纯准确度值分类器具有更高的准确度值和更低的小类概率, 并构建了基于纯准确度学习的更紧的泛化误差

上界. 文献 [38], [39], [40]分别提出优化纯准确度值的选择性集成算法、plug-in算法及支持向量机.

在消除信息熵与兰德指数的随机一致性方面, Romano等人[33] 定义了缓解随机一致性的AMIC评价指标, 该指标对于均匀噪音具有鲁棒性. Gates等人[2]在固定类簇个数、固定类簇大小、所有类簇三种假设及单边固定或双边固定两种情形下, 给出期望信息熵与期望兰德指数的定义, 得出不同的随机模型导致的评价结果不同. Vinh等人[3]分析对比了标准互信息和调整互信息的度量性质、随机零基准线性质, 并在两种类簇评价任务中实验验证了调整互信息不易受类簇个数的影响.

综上, 在分类任务中, 缓解采用固定边缘分布定义的随机一致性有助于构建低偏差的模型; 在聚类任务中, 缓解随机一致性有助于构建不受类簇个数影响的评价指标; 在关联分析中, 缓解随机一致性有助于构建不受均匀噪音影响的评价指标. 众所周知, 决策树算法存在多值偏向问题, 本文将基于随机一致性缓解该问题.

现对上述四个方面的研究进行整体层面的总结和分析. 关于决策树算法的研究包含树结构、特征重要度评价准则、适应新任务、处理新数据、模型深度化等方面. 其中, 特征重要度评价准则对于决策树算法规则的构建至关重要. 这方面的研究包含评估指标的定义、抽样分布估计、准则优化特性、多值偏向问题等层面的研究.

目前, 关于特征重要度评价准则的多值偏向问题主要通过求重要度的p值或分位数, 然后用这两种值代替重要度进行特征评估的方式进行解决. 求重要度的p值或分位数的方法涉及到求重要度的均值与方差, 一般通过蒙特卡洛计算、近似分布替代、精确计算的方法得到均值与方差. 然而, 蒙特卡洛计算方法存在着时间成本高、近似分布替代与精确计算存在着近似分布偏差的问题.

因缓解随机一致性的方法已被验证可缓解变量分布带来的一致性. 本文的核心研究问题为基于缓解随机一致性缓解基尼指数的多值偏向问题, 这个方式同样涉及到基尼指数的均值与方差的计算, 本文通过更近似的分布与精确计算的方式给出基尼指数均值和方差的计算公式.

2 基于不纯度函数的属性评价准则及其多值偏向问题

在决策树分类算法中, 节点的属性重要度选择函数决定了树的泛化能力、深度、平衡程度等重要性能. 通常, 决策树采用的特征选择准则大多数是基于不纯度函数定义的. 而大多数不纯度函数是基于某个凹函数定义的, 这使其判断结果受到属性取值个数的影响, 即不纯度函数倾向于赋予取值多的属性较高的重要程度. 本节, 给出可用于定义不纯度函数的凹函数并从理论上证明其多值偏向问题. 文献 [41]分别给出二类情形下Gini 指数、信息增益、卡方统计量的多值偏向问题. 本节将其结果扩展到多类分类任务及一类基于凹函数定义的不纯度函数.

设训练集合为$\mathcal{S}_N = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$, 样本空间$X \in \mathcal{X} \subseteq \mathcal{R}^d$, 属性个数为$d$, 即$\boldsymbol{x}_i$由$d$ 维向量表示, 每个维度代表一个特征, 特征集合为$\mathcal{A} = \{A_1, A_2, ..., A_d\}$. 标签集合为$Y \in \mathcal{Y} = \{c_1, c_2, ..., c_k\}$, 标签个数为$k$. 给定一个特征$A$, 可能的取值个数为$r \leqslant N$, 特征$A$ 与标签Y形成的交叉列联表如表1 所示. 其中, n_i^A表示在A属性上取值为a_i的样本个数, n_j^Y 表示类别属性c_j的样本个数, n_{ij}表示A 属性上取值为a_i, 类别属性为c_j的样本个数. 样本属于第j类的先验概率为$p(c_j)$, 类别与特征的联合概率为$p(c_j, a_i)$, 条件概率为$p(c_j|a_i)$, 可分别通过$\frac{n_j^Y}{N}$, $\frac{n_{ij}}{N}$ $\frac{n_{ij}}{n_i^A}$进行估计.

首先, 给出不纯度函数的定义:

定义1 设k元组$(\pi_1, \pi_2, ..., \pi_k)$满足$\pi_i \geqslant 0$, $i = 1, 2, ..., k$ 且$\sum_i \pi_i = 1$, ϕ为定义在其上的函数, ϕ称为不纯度函数, 如果ϕ满足如下条件:

(1)ϕ在$(\frac{1}{k}, \frac{1}{k}, ..., \frac{1}{k})$上达到最大值;

表 1　特征A与标签Y的列联表

Table 1　Cross table between the feature A and label Y

$A\backslash Y$	$Y=c_1$	…	$Y=c_j$	…	$Y=c_k$	Total(A)
$A=a_1$	n_{11}	…	n_{1j}	…	n_{1C}	n_1^A
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$A=a_i$	n_{i1}	…	n_{ij}	…	n_{iC}	n_i^A
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$A=a_r$	n_{r1}	…	n_{rj}	…	n_{rC}	n_r^A
Total(Y)	n_1^Y	…	n_j^Y	…	n_C^Y	N

表 2　特征A'与标签Y的列联表

Table 2　Cross table between the feature A' and label Y

$A'\backslash Y$	$Y=c_1$	…	$Y=c_j$	…	$Y=c_k$	Total(A')
$A'=a_1$	n_{11}	…	n_{1j}	…	n_{1C}	$n_1^{A'}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$A'=a_i$	n_{i1}	…	n_{ij}	…	n_{iC}	$n_i^{A'}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$A'=a'_r$	n'_{r1}	…	n'_{rj}	…	n'_{rC}	$n_r^{A'}$
$A'=a'_{r+1}$	$n'_{r+1,1}$	…	$n'_{r+1,j}$	…	$n'_{r+1,C}$	$n_{r+1}^{A'}$
Total(Y)	n_1^Y	…	n_j^Y	…	n_C^Y	N

(2)ϕ在$(1,0,...,0)$,$(0,1,...,0)$,...,$(0,0,...,1)$上达到最小值;

(3)ϕ关于$(\pi_1,\pi_2,...,\pi_k)$对称;

基于如下四种凹函数, 分别称为渐进1近邻误差函数、信息熵、Bayes最小误判率、Matushita误差,

$$f(x)=x(1-x) \tag{1}$$

$$f(x)=-x\log_2 x \tag{2}$$

$$f(x)=\min\{x,1-x\} \tag{3}$$

$$f(x)=\sqrt{x(1-x)} \tag{4}$$

可定义不纯度函数为:

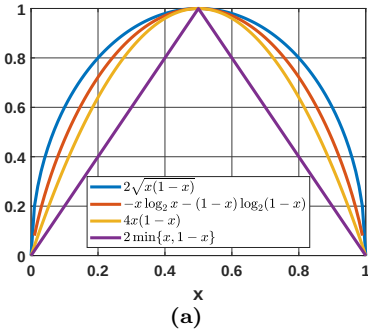
$$\phi(\pi_1,\pi_2,...,\pi_k)=\Sigma_{i=1}^k f(\pi_i). \tag{5}$$

其中$\pi_i\in[0,1]$, 且$\sum_i\pi_i=1$. 图1展示了二类情形下$k=2$时, 基于四种凹函数的不纯度函数. 可见, 这四种凹函数均满足不纯度函数的性质要求. 给定一个不纯度函数, 属性取值为a的样本集合的不纯度为:

$$Im(a)=\phi(p(c_1|a),p(c_2|a),...,p(c_k|a)), \tag{6}$$

其中$p(c_j|a)$属性取值为a 的样本集合中类别为c_j 的样本比例. 如果特征A具有r种取值, 特征A 对类别不纯度的降低程度定义为:

$$\Delta Im(A)=Im(Y)-\Sigma_{i=1}^r p(a_i)Im(a_i). \tag{7}$$



(a)

图 1　凹函数

Figure 1　The concave funtion

例如, 根据公式(1)所示的渐进1近邻误差与公式(2)所示的信息熵, Gini指数与信息增益分别定义为:

$$Gini(A)=1-\Sigma_{j=1}^{k}(p(c_j))^2-\Sigma_{i=1}^{r}p(a_i)\Sigma_{j=1}^{k}p(c_j|a_i)(1-p(c_j|a_i)), \tag{8}$$

$$Gain(A)=-\Sigma_{j=1}^{k}p(c_j)\log_2 p(c_j)+\Sigma_{i=1}^{r}p(a_i)\Sigma_{j=1}^{k}p(c_j|a_i)\log_2 p(c_j|a_i). \tag{9}$$

不纯度函数的降低程度直接反映了特征的重要程度. 原因是, 关于标签向量的信息增益、基尼指数等不纯度函数度量了集合中样本类别的单一程度. 不纯度函数的函数值越高, 集合中所含样本的标签种类越多或分布越均匀. 当按照属性取值将样本集合划分为不同子集时, 若子集的纯度越高, 则这个属性对于决策的重要程度越大. 此时, 在这些子集上计算样本标签的不纯度函数值较小, 相应地, 降低原始集合不纯度的程度较高.

为更加直观地理解不纯度降低程度与特征重要度的关系, 以渐进1近邻误差函数$f(p)=p(1-p)$导出的二类问题的不纯度函数基尼指数$\phi(p)=2p(1-p)$为例进行进一步阐述. 显然, 不纯度函数$f(p)$在两类分布完全相同$p=1/2$时, 取值最大. 此时, 集合内样本的标签分布纯度最低. 表3给出了十个样本的标签及a, b, c三种属性的取值. 如表3所示, 若按照属性a对样本集合划分, 得到的两个样本子集为$\{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3\}$与$\{\boldsymbol{x}_4,\boldsymbol{x}_5,\boldsymbol{x}_6,\boldsymbol{x}_7,\boldsymbol{x}_8,\boldsymbol{x}_9,\boldsymbol{x}_{10}\}$. 相应地, 按照公式(8)计算, 不纯度函数降低值为0.2700. 若按照属性b对样本集合划分, 得到的两个样本子集为$\{\boldsymbol{x}_1,\boldsymbol{x}_2\}$与$\{\boldsymbol{x}_3,\boldsymbol{x}_4,\boldsymbol{x}_5,\boldsymbol{x}_6,\boldsymbol{x}_7,\boldsymbol{x}_8,\boldsymbol{x}_9,\boldsymbol{x}_{10}\}$. 相应地, 不纯度函数降低值为0.1200. 显然, 属性a对样本集合的不纯度降低程度高于属性b的降低程度. 并且, 直观来看, 根据属性a对样本进行分类能取得更好的分类准确度. 那么属性a对于决策更为重要. 因此, 不纯度函数的降低程度可反映特征对于决策的重要程度.

进一步, 通过表3说明不纯度函数选择特征时存在的多值偏向问题. 将属性b的取值进一步细化, 取值为b_2的样本集合进行分裂, 得到属性b'. 若按照属性b'对样本集合划分, 得到的三个样本子集为$\{\boldsymbol{x}_1,\boldsymbol{x}_2\}$, $\{\boldsymbol{x}_3,\boldsymbol{x}_4,\boldsymbol{x}_5,\boldsymbol{x}_6\}$, $\{\boldsymbol{x}_7,\boldsymbol{x}_8,\boldsymbol{x}_9,\boldsymbol{x}_{10}\}$. 相应地, 不纯度函数降低值为0.1733. 从直观来看, 属性b与属性b'的分类准确度相同, 二者的区别仅仅在于取值个数的不同. 然而, 属性b'的不纯度函数降低值高于属性b的不纯度函数值. 这说明不纯度函数具有多值偏向问题. 接下来, 从理论上证明存在基于凹函数的不纯度函数均具有多值偏向问题.

假设属性A的取值为$a_1,a_2,...,a_r$, 另外一个属性A' 的取值为$a_1,a_2,...,a'_r,a'_{r+1}$, A' 的属性取值等价于把原先a_r属性的样本任意划分为两组, 并随机赋予a'_r,a'_{r+1} 的取值. 属性A'与标签Y形成的列联表如表2所示. 由于样本取a'_r或a'_{r+1}是完全随机的, a'_r与a'_{r+1}形成的划分结果不包含对决策有用的信息. 那么, A' 应该不比A 更重要. 如果评价准则ΔIm 赋予A'较高的值$\Delta Im(A)\leqslant\Delta Im(A')$, 称基于不纯度函数$\phi$ 的评价准则具有多值偏向问题.

表 3　不纯度降低程度与特征重要度的关系示例以及多值偏向问题示例

Table 3　An example of the relationship between the degree of impurity reduction and feature importance and the multi-value bias problem

X	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$	$\boldsymbol{x}_6$	$\boldsymbol{x}_7$	$\boldsymbol{x}_8$	$\boldsymbol{x}_9$	$\boldsymbol{x}_{10}$	Gini	SGini
Y	1	1	1	1	2	2	2	2	2	2		
a	a_1	a_1	a_1	a_2	a_2	a_2	a_2	a_2	a_2	a_2	0.2700	3.7690
b	b_1	b_1	b_2	b_2	b_2	b_2	b_2	b_2	b_2	b_2	0.1200	2.1046
b'	b_1	b_1	b'_2	b'_2	b'_2	b'_2	b'_3	b'_3	b'_3	b'_3	0.1733	1.5781

性质1指出基于凹函数定义的不纯度函数存在多值偏向问题(证明见附录).

性质1　基于公式(5), (6), (7)定义的不纯度函数存在多值偏向问题.

3　缓解随机一致性的基尼指数

为了刻画特征A的重要程度, 研究者通常采用A与Y之间的关联函数[42]、相似函数或A对Y的增益函数. 这些函数通常反映了A与Y之间的某种一致程度. 例如, 常用的Kendall tau系数刻画了A与Y的增减一致程度, 以及常用的信息增益刻画了A与Y对数据集划分的一致程度等. 在A与Y一致性刻画过程中, 除了A本身的重要程度外, 还受到其他因素的影响. 这些因素包括属性取值个数、属性本身分布的均衡性、均匀噪音、样本个数等.

缓解随机一致性的方法首先需要引入一个变量集合, 集合中所有变量的取值都是随机的, 但都满足某种约束条件. 这个约束条件与需要缓解的特定因素相关. 然后, 使用集合中变量与真实变量一致性的平均值或中位数等某种代表整体的数字特征, 作为特定因素引起的一致性. 由于集合中变量都是随机的, 因此其一致性的平均值刻画了随机一致性的程度; 最后, 在某种定义式框架下, 通过在一致性刻画中减去随机一致性, 可缓解由于特定因素引起的一致性程度.

本文通过缓解随机一致性的方法消除属性取值个数对一致性评判的影响. 首先假设一个置换集合$\mathcal{A}_{perm}$, 该集合中的变量与特征变量的边缘分布相同, 但取值是完全随机的. 由于仅在边缘分布上与特征相同, $\mathcal{A}_{perm}$中变量与Y形成的一致性可用来刻画由于特征的边缘分布特性与Y形成的一致性程度. 通过减去这部分一致性, 可得到缓解变量取值影响的一致性程度.

在缓解随机一致性的定义式框架方面, Wang等人[38~40]在纯一致性定义式框架下缓解了分类评价准则准确度的随机一致性. 然而, 纯一致性框架中未考虑一致性指标的方差信息, 使其对随机一致性的缓解程度较弱. Romano等人[35]提出标准化框架下的类簇评价准则比纯一致性框架下的效果更好. 因此, 本文采用标准化框架缓解随机一致性.

记A与Y之间的一致程度为$CM(A,Y)$, 那么在标准化的定义式框架下, A与Y之间的标准化一致程度$SCM(A,Y)$:

$$SCM(A,Y)=\frac{CM(A,Y)-\mathbb{E}(CM(A,Y))}{\mathbb{S}(CM(A,Y))}, \tag{10}$$

其中$\mathbb{E}(CM(A,Y))$与$\mathbb{S}(CM(A,Y))$分别为$\mathcal{A}_{perm}$中特征向量一致程度$CM(A,Y)$的期望与标准差.

3.1　标准基尼指数及其决策树算法

经典的决策树算法ID3算法[5]与CART算法[7]分别基于信息增益与基尼指数构建. 关于信息增益的

多值偏向问题以及缓解方法, 已取得较多的研究与进展[2,3,26]. 而现有研究[33,34]在计算基尼指数期望与方差时, 假设列联表元素服从多项分布, 这一假设存在偏差. 基于此, 本文将关注基尼指数的多值偏向问题.

在公式(10)给出的标准化框架下, 可定义标准基尼指数为:

$$SGini(A,Y)=\frac{Gini(A,Y)-\mathbb{E}_{perm}Gini(A)}{\sqrt{\mathbb{V}_{perm}Gini(A)}}, \tag{11}$$

可见, 标准化框架中需要计算基尼指数的期望与方差, 为了论文的可读性, 将在3.2节与3.3节给出基尼指数期望与方差的计算过程与结果.

通过第2节性质1的分析可知, 基尼指数$Gini(A,Y)$ 随着取值的增多而增加. 根据3.2节中公式(16)可知, 基尼指数的期望$\mathbb{E}_{perm}Gini(A)$随着属性取值个数r的增加而增加. 由此可见, 随着属性取值个数的增多, $SGini(A,Y)$分子的增减趋势不确定, 且其分母形式复杂. 因此, $SGini(A,Y)$不存在明显的多值偏向问题.

基于标准基尼指数的决策树算法1所示. 在处理离散属性时, 该算法与ID3算法均为多叉树的树结构, 不同之处在于ID3使用信息增益为节点属性选择准则. 算法1按照属性的取值个数判断属性是离散属性还是连续属性. 当属性取值个数大于d_{num} 时, 判定该属性为连续属性. 对于连续属性, 如果采用Cart算法处理连续属性的方式, 对每一个属性先寻找最佳切分点, 再评估属性. 这样的方式评估每一个属性时需要遍历所有样本的值, 时间成本极高. 如果先将数据预先离散化, 再采用ID3算法处理离散属性的方式, 为每一个节点构建一颗子树, 那么样本点会很快被分散在每个节点中, 导致建树算法过早达到停止条件, 结果得到一颗层平均节点数多(宽)、层数少(浅)的决策树, 形成的划分规则依赖于极少数的特征, 树模型泛化性能较差.

综上, 为了兼顾泛化性能与时间成本, 使得基于标准基尼指数的决策树算法能够更好地处理连续属性较多的数据集. 对于连续属性, 在建树过程将Cart算法的二值离散化方式扩展为多划分的方式. 具体地, 在评估每一列属性之前, 采用层次聚类的方法对属性进行单变量聚类, 离散化的块数设置为最大化评估指标的聚类个数. 离散化区间端点值为每个类簇中样本点的最小值与最大值. 对于测试样本, 采用同样的端点值进行离散化. 离散化之后, 每一列属性当作离散属性进行特征评估. 这样基于聚类的离散化块数取决于聚类的个数, 最终结果可能是将属性二值化或保持原有取值. 因此, 这样的方式同时包含了Cart与ID3处理属性的方式. 而最终离散化的方式在树构建过程中按照评估指标值大小自适应选择, 具体流程如算法1所示.

现有决策树多值偏向问题的解决方法包括AIR[32]方法, PIMP方法[36], $\triangle g_e$[34]方法, AGini方法[33]等. AIR方法在衡量属性重要度时, 将属性在样本上的取值进行一次置换排列, 用这次排列的重要度值作为基准线, 高于基准线的部分作为最后的评估值. 这种方法的随机性较强, 每一次不同的置换排列得到不同的基准线, 从而得到不同的评估值. 与置换属性在样本上的取值排列不同, PIMP方法置换标签在样本上的排列, 并假设评估值服从高斯分布或伽马分布, 分布的参数通过多次排列得到的评估值进行估计, 最终使用分布下评估值对应的p值的负对数作为重要度评价准则. 这种基于统计分布的p值的方法需要对评估值的分布作出假设, 并且评估结果比较依赖于所假设的分布. 此外, PIMP方法需要置换多次标签的排列估计分布的参数, 时间成本较高.

$\triangle g_e$方法假设列联表元素n_{ij}服从多项分布, 在此基础上给出基尼指数的期望与方差如公式(12)所示, 进一步假设基尼指数服从伽马分布, 以基尼指数在该分布下对应的p值作为评估结果. AGini方法根据$\triangle g_e$方法给出的基尼指数的期望与方差, 通过Cantelli不等式得到基尼指数的$1-\alpha$分位数评估特征

算法 1 基于标准基尼指数的决策树算法

输入: 数据集$\boldsymbol{X}=\{(\boldsymbol{x_1},y_1),(\boldsymbol{x_2},y_2),\ldots,(\boldsymbol{x_n},y_n)\}$; 节点最小样本数: $minnum$, 离散属性最大取值个数d_{num}

过程:

如果样本数量小于$minnum$或者所有样本的类别相同, 分支停止生长;
否则,
for 对于每个特征a_i, **do**
  if $|V_{a_i}|\leqslant d_{num}$, ($V_{a_i}$是$a_i$的取值范围), **then**
    计算$SGini_{a_i}=SGini(\{a_i\},\{y\})$, 分割值$\boldsymbol{t}_i=V_{a_i}$.
  else
    根据a_i对样本进行层次聚类树状图
    for 对于$k=1:1:|V_{a_i}|$ **do**
      根据层次聚类树状图得到类簇个数为k的聚类结果:
      $\pi^1=\{a_i(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{\pi^1}}),...,a_i(\boldsymbol{x}_{|\boldsymbol{\pi^1}|}^{\boldsymbol{\pi^1}})\},...,\pi^k=\{a_i(\boldsymbol{x}_{\boldsymbol{1}}^{\boldsymbol{\pi^k}}),...,a_i(\boldsymbol{x}_{|\boldsymbol{\pi^k}|}^{\boldsymbol{\pi^k}})\}$
      根据聚类结果得到分割向量, 由类簇内样本形成区间的端点值组成:
      $\boldsymbol{t}_i^k=(-inf,\max\{\pi^1\},\min\{\pi^2\},\max\{\pi^2\},...,\min\{\pi^k\},inf)$
      $a_i(\boldsymbol{t}_i^k)$表示a_i关于分割向量$\boldsymbol{t}_i^k$的多值特征(取值位于分割向量相邻元素之间样本赋予相同的离散化特征),
      计算$SGini_{a_i(\boldsymbol{t}_i^k)}=SGini(\{a_i(\boldsymbol{t}_i^k)\},\{y\})$
    end for k
  end if
end for i
确定最佳分割特征a^*及其分割向量$\boldsymbol{t}^*$:
$(a^*,\boldsymbol{t}^*)=arg\max_i\max_k SGini(\{a_i(\boldsymbol{t}_i^k)\},\{y\})$.
使用a^*和$\boldsymbol{t}^*$建立节点, 按照$a^*(\boldsymbol{t}^*)$的取值对样本进行划分, 更新数据集
为$a^*(\boldsymbol{t}^*)$的每一个取值构建分支, 递归创建新的分割点直到满足停止条件.

输出: 决策树T

的重要度. $\triangle g_e$方法与AGini方法关于列联表元素的分布假设存在一定的偏差. 如图2与引理1所示, 当属性变量服从均匀分布时, 其与标签变量形成的列联表元素n_{ij}服从超几何分布而非多项分布.

本文提出的标准基尼指数的期望与方差的计算在n_{ij}服从超几何分布的假设下给出, 并通过"减均值, 除以标准差"的方式缓解基尼指数的多值偏向. 综上所述, 与传统缓解多值偏向的决策树算法相比, 本文决策树算法所采用的特征评价准则标准基尼指数不依赖于基尼指数分布的假设, 仅仅依赖于基尼指数的期望与方差. 并且, 该标准基尼指数所采用期望与方差是在更近似的(列联表元素的)分布假设下通过解析计算的方式给出.

3.2 置换集合中基尼指数的期望值

本小节给出置换集合$\mathcal{A}_{perm}$中特征向量的基尼指数期望值的计算方式. 文献 [33, 34]在列联表元素n_{ij}服从多项分布假设下给出基尼指数的期望与方差, 如公式(12)[33, 34]所示.

$$\mathbb{E}(\triangle g_e) = \frac{r-1}{N}\big(1-\sum_{j=1}^{k} p_j^2\big), \tag{12}$$

$$\begin{aligned}\mathbb{V}(\triangle g_e) = \frac{1}{N^2}\Big[&(r-1)\big(2\sum_{j=1}^{k} p_j^2 + 2(\sum_{j=1}^{k} p_j^2)^2 - 4\sum_{j=1}^{k} p_j^3\big)\\ &+ \big(\sum_{i=1}^{r}\frac{1}{n_i^A} - 2\frac{r}{N} + \frac{1}{N}\big)\big(-2\sum_{j=1}^{k} p_j^2 - 6(\sum_{j=1}^{k} p_j^2)^2 + 8\sum_{j=1}^{k} p_j^3\big)\Big],\end{aligned}$$

其中, r, k, N分别为特征A的取值个数, 类别个数, 样本数, p_j为第j类样本的概率, n_i^A为$A=a_i$的样本数. 然而, 特征集合$\mathcal{A}_{perm}$中特征向量的取值分布以及真实标签Y的取值分布是固定的. 当列联表的边缘分布n_i^A与n_j^Y给定时, 列联表元素n_{ij}可证明服从超几何分布.

本节首先证明置换集合中的特征向量与Y形成的列联表元素服从超几何分布, 如引理1所示(3.2节主要结果), 再根据组合数的性质给出置换集合基尼指数的期望与标准差的计算方式, 如公式(16)(3.2节主要结果)与公式(23), 公式(24), 公式(25), 公式(26)所示(3.3节主要结果). 上述三个主要结果为本文主要贡献与改进之处, 3.2节与3.3节用到的其他公式为定义式或已有研究结果. 为了区分本文贡献与已有工作, 本节在已有引理或公式上添加了引用的参考文献. 此外, 引理1与公式(16), 公式(24), 公式(26)的证明在附录中展示.

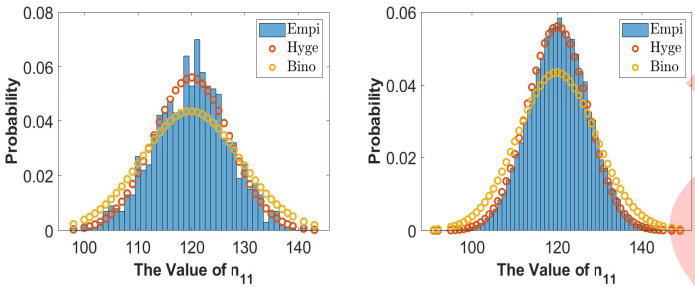
引理1 当$\mathcal{A}_{perm}$中的特征向量A'边缘分布及取值个数固定时, 其与Y形成的列联表元素n_{ij}服从参数为N, n_j^Y, n_i^A的超几何分布. 即从N个样本(其中包含n_j^Y个第j类样本）中不放回地抽出n_i^A个样本, 抽到n个第j类样本的概率. 记作$n_{ij} \sim H(N, n_j^Y, n_i^A)$, 即:

$$\mathbb{P}_{A\in\mathcal{A}_{perm}}\big(n_{ij}=n\big) = \frac{\mathbf{C}_{n_j^Y}^{n}\mathbf{C}_{N-n_j^Y}^{n_i^A-n}}{\mathbf{C}_{N}^{n_i^A}}, \tag{13}$$

其中$n=\max\{0, n_i^A+n_j^Y-N\}, ..., \min\{n_i^A, n_j^Y\}$, $\mathbf{C}_n^r$为从n个样本中取r个样本的组合数.

文献 [33, 34]提出列联表元素n_{ij}服从参数为$N, n_j^Y/N$的多项分布. 实际上, 通过引理1的证明可知, 当集合$\mathcal{A}_{perm}$中的特征向量边缘分布不固定的向量时, n_{ij}服从多项分布. 而当置换集合中特征属性的边缘分布固定时, n_{ij}服从超几何分布.

图2 通过模拟实验验证这一点. 在模拟实验中, 设置属性与标签的类别数为2, 样本个数$N=1000$, 其中标签为1的样本个数为$n_1^Y = 200$, 标签为2的样本个数为$n_2^Y = 800$, 标签向量为固定的向量. 属



(a) S=1000 **(b)** S=30000

图 2　置换集合的列联表元素服从超几何分布

Figure 2　The contingency table elements of a permutation set obey hypergeometric distribution

性取值为1的样本个数为$n_1^A = 400$, 取值为2的样本个数为$n_2^A = 600$. 为生成这样的属性向量, 从样本索引向量$(1, 2, ..., N)$中随机抽取n_1^A个位置放1, 剩余位置放2. 这样的随机抽样进行S次. 图2展示了$S = 1000$与$S = 30000$ 时2×2 列联表中n_{11} 的经验分布、超几何分布与二项分布概率值. 可见, n_{11}的经验分布更接近于对应参数下的超几何分布.

接下来, 基于引理1的超几何分布假设给出$\mathcal{A}_{perm}$中特征向量基尼指数的期望值与标准差. 记$(n)_l = \frac{n!}{(n-l)!}$, 首先根据性质$(n)_l \mathbf{C}_M^n = (M)_l \mathbf{C}_{M-l}^{n-l}$, 给出引理:

引理2 [44] 假设样本总量为N, 其中包含M个具有某种特定性质的样本, 从中无放回抽取K个, 那么有:

$$\Sigma_n (n)_l \frac{\mathbf{C}_M^n \mathbf{C}_{N-M}^{K-n}}{\mathbf{C}_N^K} = \frac{(K)_l (M)_l}{(N)_l} \tag{14}$$

基尼指数关于频数的表达为:

$$Gini(A) = -\Sigma_{j=1}^k \left(\frac{n_j^Y}{N}\right)^2 + \Sigma_{i=1}^r \Sigma_{j=1}^k \frac{(n_{ij})^2}{N n_i^A}, \tag{15}$$

根据引理2以及$n^2 = (n)_2 + (n)_1$, 得到置换集合中基尼指数的期望值为:

$$\mathbb{E}_{perm} Gini(A) = \frac{r-1}{N-1}\left(1 - \Sigma_{j=1}^k \left(\frac{n_j^Y}{N}\right)^2\right) \tag{16}$$

文献 [33,34]在多项分布下给出的基尼指数期望值的计算公式(12)与公式(16)的关系如下:

$$\mathbb{E}_{n_j^Y} \mathbb{E}_{perm} Gini(A) = \mathbb{E}(\triangle g_e), \tag{17}$$

即公式(16)是关于$\mathcal{A}_{perm}$中特征的期望, 是在列联表边缘分布给定的条件(即将n_j^Y, n_i^A当作固定常数)下求得的期望. 当将n_j^Y视作随机变量时, 再将$\mathbb{E}_{perm} Gini(A)$关于$Y$求期望, 会得到一样的结果. 这一现象符合多项分布与超几何分布期望之间的关系形式.

显然, 公式(16)给出的基尼指数期望的计算复杂度为$\mathcal{O}(k)$. 这归功引理2, 这个引理给出平方函数乘以超几何分布概率的求和项的多项式表达结果. 然而对数函数不存在类似的形式, 其期望计算复杂度较高, 文献 [2]给出信息增益期望的计算方式如公式(18)[2]所示:

$$Gain(A) = -\Sigma_{j=1}^k \frac{n_j^Y}{N} \log_2 \frac{n_j^Y}{N} + \Sigma_{i=1}^r \Sigma_{j=1}^k \Sigma_{n \leqslant \min\{n_i^A, n_j^Y\}} \frac{n}{N} \log_2 \frac{n}{n_i^A} \mathbb{P}(n_{ij} = n). \tag{18}$$

这种计算方式的复杂度为$\mathcal{O}(rk\max\{n_1^A,...,n_r^A,n_1^Y,...,n_k^Y\})$.

3.3 置换集合中基尼指数的方差

根据方差的定义,

$$\mathbb{V}_{perm}Gini(A)=\mathbb{E}_{perm}[Gini(A)]^2-[\mathbb{E}_{perm}Gini(A)]^2, \tag{19}$$

其中$\mathbb{E}_{perm}[Gini(A)]^2$的关键在于四次多项式$(\sum_{i=1}^r\sum_{j=1}^k(n_{ij})^2)^2$期望的计算, 该四次多项式中的交叉项$n_{ij}^2n_{i'j'}^2$ 涉及到联合概率分布$\mathbb{P}(n_{ij},n_{i'j'})$的计算. 在信息增益方差的计算过程中, 同样存在对联合概率$\mathbb{P}(n_{ij},n_{i'j'})$的估计. 为了计算信息增益的方差, 文献 [2]将联合概率的计算转换为条件概率的计算, 并证明了概率$\mathbb{P}(n_{i'j}|n_{ij})$, $\mathbb{P}(n_{ij'}|n_{ij})$, $\mathbb{P}(n_{i'j'}|n_{i'j},n_{ij})$服从超几何分布, 如公式(20)[2]所示:

$$\begin{aligned} n_{i'j}|n_{ij}&\sim H(N-n_i^A,n_{i'}^A,n_j^Y-n_{ij})\\ n_{ij'}|n_{ij}&\sim H(N-n_j^Y,n_{j'}^Y,n_i^A-n_{ij})\\ n_{i'j'}|n_{ij'},n_{ij}&\sim H(N-n_i^A,n_{j'}^Y-n_{ij'},n_{i'}^A). \end{aligned} \tag{20}$$

按照同样的方式, 本节给出基尼指数方差的计算方式, 并根据引理2给出基尼指数方差关于边缘概率的表达式, 从而降低计算复杂度.

经过简单的计算, 基尼指数平方的期望为:

$$\begin{aligned} &\mathbb{E}_{perm}[Gini(A)]^2\\ &=-\left(\Sigma_{j=1}^k(\frac{n_j^Y}{N})^2\right)^2-2\Sigma_{j=1}^k(\frac{n_j^Y}{N})^2\mathbb{E}_{perm}Gini(A)+\mathbb{E}_{perm}\left(\sum_{i=1}^r\sum_{j=1}^k\frac{(n_{ij})^2}{Nn_i^A}\right)^2 \end{aligned} \tag{21}$$

根据公式(20), 将列联表元素按照行列是否相同的四种情况进行划分: $(i'=i,j'=j)$, $(i'\neq i,j'=j)$, $(i'=i,j'\neq j)$及$(i'\neq i,j'\neq j)$, 公式(21)中的第三项为:

$$\begin{aligned} &\mathbb{E}_{perm}\left(\sum_{i=1}^r\sum_{j=1}^k\frac{(n_{ij})^2}{Nn_i^A}\right)^2\\ &=\sum_{i=1}^r\sum_{j=1}^k\sum_n\left(\frac{n^2}{Nn_i^A}\right)^2\mathbb{P}(n_{ij}=n)\\ &+\sum_{i=1}^r\sum_{j=1}^k\sum_n\frac{n^2}{Nn_i^A}\mathbb{P}(n_{ij}=n)\sum_{i'\neq i}^r\sum_{n'}\frac{n'^2}{Nn_i^A}\mathbb{P}(n_{i'j}=n'|n_{ij}=n)\\ &+\sum_{i=1}^r\sum_{j=1}^k\sum_n\frac{n^2}{Nn_i^A}\mathbb{P}(n_{ij}=n)\sum_{j'\neq j}^k\sum_{n'}\frac{n'^2}{Nn_i^A}\mathbb{P}(n_{ij'}=n'|n_{ij}=n)\\ &+\sum_{i=1}^r\sum_{j=1}^k\sum_n\frac{n^2}{Nn_i^A}\mathbb{P}(n_{ij}=n)\sum_{j'\neq j}^k\mathbb{P}(n_{ij'}=n'|n_{ij}=n)\sum_{i'\neq i}^r\sum_{n''}\frac{n''^2}{Nn_i^A}\mathbb{P}(n_{i'j'}=n''|n_{ij'}=n',n_{ij}=n). \end{aligned} \tag{22}$$

对于公式(22)的计算, 先将其中n的l次幂函数转换为$(n)_l,(n)_{l-1},...,n$的线性函数. 再根据引理2显示表达$\sum_n f(n^l)\mathbb{P}(n_{ij}=n)$的求和结果, 其中$f$为关于$n^l$的函数, 便可将公式(22)表示关于$n_i^A$, n_j^Y的求和项, 从而大幅度地降低计算复杂度. 具体地,

根据公式(50)及引理2, 公式(22)中第一项可表示为:

$$\begin{aligned}&\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{n}\left(\frac{n^2}{Nn_i^A}\right)^2\mathbb{P}(n_{ij}=n)\\&=\sum_{i=1}^{r}\sum_{j=1}^{k}\frac{1}{(Nn_i^A)^2}\left(\frac{(n_i^A)_4(n_j^Y)_4}{(N)_4}+6\frac{(n_i^A)_3(n_j^Y)_3}{(N)_3}+7\frac{(n_i^A)_2(n_j^Y)_2}{(N)_2}+\frac{n_i^An_j^Y}{N}\right),\end{aligned}\tag{23}$$

该式只包含对n_i^A, n_j^Y的运算.

根据引理2及公式(50), 公式(22)中第二项可表示为:

$$\begin{aligned}&\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{n}\frac{n^2}{Nn_i^A}\mathbb{P}(n_{ij}=n)\sum_{i'\neq i}^{r}\sum_{n'}\frac{n'^2}{Nn_{i'}^A}\mathbb{P}(n_{i'j}=n'|n_{ij}=n)\\&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{i'\neq i}^{r}\frac{1}{N^2n_i^An_{i'}^A}\left(\overline{C}_4\frac{(n_i^A)_4(n_j^Y)_4}{(N)_4}+\overline{C}_3\frac{(n_i^A)_3(n_j^Y)_3}{(N)_3}+\overline{C}_2\frac{(n_i^A)_2(n_j^Y)_2}{(N)_2}+\overline{C}_1\frac{n_i^An_j^Y}{N}\right),\end{aligned}\tag{24}$$

该式只包含对n_i^A, $n_{i'}^A$, n_j^Y的运算. 其中$\overline{C}_1=C_2+C_3+C_4$, $\overline{C}_2=C_2+3C_3+7C_4$, $\overline{C}_3=C_3+6C_4$, $\overline{C}_4=C_4$, $C_1=\frac{n_{i'}^A}{N-n_i^A}$, $C_4=\frac{(n_{i'}^A)_2}{(N-n_i^A)_2}$, $C_2=(C_1-C_4)n_j^Y+C_4(n_j^Y)^2$, $C_3=-2n_j^YC_4+C_4-C_1$.

按照同样的方式, 公式(22)中第三项可表示为:

$$\begin{aligned}&\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{n}\frac{n^2}{Nn_i^A}\mathbb{P}(n_{ij}=n)\sum_{j'\neq j}^{k}\sum_{n'}\frac{n'^2}{Nn_i^A}\mathbb{P}(n_{ij'}=n'|n_{ij}=n)\\&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{j'\neq j}^{k}\frac{1}{(Nn_i^A)^2}\left(\overline{C}_4\frac{(n_i^A)_4(n_j^Y)_4}{(N)_4}+\overline{C}_3\frac{(n_i^A)_3(n_j^Y)_3}{(N)_3}+\overline{C}_2\frac{(n_i^A)_2(n_j^Y)_2}{(N)_2}+\overline{C}_1\frac{n_i^An_j^Y}{N}\right),\end{aligned}\tag{25}$$

其中$\overline{C}_1=C_2+C_3+C_4$, $\overline{C}_2=C_2+3C_3+7C_4$, $\overline{C}_3=C_3+6C_4$, $\overline{C}_4=C_4$, $C_1=\frac{n_{j'}^Y}{N-n_j^Y}$, $C_4=\frac{(n_{j'}^Y)_2}{(N-n_j^Y)_2}$, $C_2=(C_1-C_4)n_i^A+C_4(n_i^A)^2$, $C_3=-2n_i^AC_4+C_4-C_1$.

根据公式(50)及引理2, 公式(22)中第四项可表示为:

$$\begin{aligned}&\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{n}\frac{n^2}{Nn_i^A}\mathbb{P}(n_{ij}=n)\sum_{j'\neq j}^{k}\mathbb{P}(n_{ij'}=n'|n_{ij}=n)\sum_{i'\neq i}^{r}\sum_{n''}\frac{n''^2}{Nn_{i'}^A}\mathbb{P}(n_{i'j'}=n''|n_{ij'}=n',n_{ij}=n)\\&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{j'\neq j}^{k}\sum_{i'\neq i}^{r}\frac{1}{N^2n_i^An_{i'}^A}\left(\overline{\overline{C}}_4\frac{(n_i^A)_4(n_j^Y)_4}{(N)_4}+\overline{\overline{C}}_3\frac{(n_i^A)_3(n_j^Y)_3}{(N)_3}+\overline{\overline{C}}_2\frac{(n_i^A)_2(n_j^Y)_2}{(N)_2}+\overline{\overline{C}}_1\frac{n_i^An_j^Y}{N}\right),\end{aligned}\tag{26}$$

其中$\overline{\overline{C}}_1=C_2+C_3+C_4$, $\overline{\overline{C}}_2=C_2+3C_3+7C_4$, $\overline{\overline{C}}_3=C_3+6C_4$, $\overline{\overline{C}}_4=C_4$, $\overline{C}_4=C_2$, $\overline{C}_3=C_2-C_1-2C_2n_i^A$, $\overline{C}_2=C_0+(C_1-C_2)n_i^A+C_2(n_i^A)^2$, $C_2=b\frac{(n_{j'}^Y)_2}{(N-n_j^Y)_2}$, $C_1=(2b(1-n_{j'}^Y)-a)\frac{n_{j'}^Y}{N-n_j^Y}$, $C_0=n_{j'}^Y(a-b)+b(n_{j'}^Y)^2$, $b=\frac{(n_{i'}^A)_2}{(N-n_i^A)_2}$, $a=\frac{n_{i'}^A}{N-n_i^A}$.

总而言之, 公式(23), 公式(24), 公式(25), 公式(26)给出基尼指数平方的期望的计算方式, 该计算方式只包含对$n_i^A,n_j^Y,n_{i'}^A,n_{j'}^Y$的运算, 对应的时间复杂度为$\mathcal{O}(r^2k^2+4rk)$.

综上, 结合公式(19), 公式(16), 公式(21)与公式(22), 公式(23), 公式(24), 公式(25), 公式(26), 便可得到基尼指数的方差.

4 标准基尼指数缓解多值偏向的有效性验证

为了验证标准基尼指数是否能够有效缓解多值偏向, 本节在人造数据集上观察标准基尼指数的选择倾向. 设置人造数据集的样本量为$N = 100$, 标签类别个数为2, 二类样本的样本量分别为$n_1^Y = 30$, $n_2^Y = 70$. 属性个数为2, 其中A_1 为取值种类较少的变量, 取值个数$r_1 = 3$; A_2为取值种类较多的变量, 取值个数$r_2 \geqslant r_1$. 设置$\mathbb{P}(A_1 \neq Y) = p_1$, $\mathbb{P}(A_2 \neq Y) = p_2$. 即$A_i$与$Y$取值不同的样本比例为$p_i$, 这部分样本的$A_i$从$\{1, .., , r_i\}$ 中均匀随机取值. 显然, p_i值越小, A_i与Y的一致性越强. 当$p_i = 0$时, 属性A_i为与Y取值完全一致的变量. 当$p_i = 1$时, 属性A_i为完全随机变量.

为了验证SGini能够缓解Gini的多值偏向问题及能够选择出具有信息的属性, 在p_i取不同值的情况下, 使用Gini,SGini两种属性重要度评价方法分别评价A_1与A_2 的重要度. 因数据的生成具有随机性, 该对比实验进行T次, 通过观察T次对比中A_1评价指标值不低于A_2值的比例来对比评价方法的选择倾向. 设置$T = 300$, 设置r_2的取值范围为$3 : 1 : 50$, 图3展示了r_2取不同值时的评价指标值. 图3的纵轴为300次实验中A_1的重要度指标值不低于A_2的重要度指标值的比例, 横轴为A_2的取值个数r_2,

第一种情况设置A_1为与Y具有一定一致性的变量, A_2为完全随机变量. 如果评价方法倾向选择A_1, 则说明此方法不具有多值偏向问题. 具体地:

- 当$p_1 = 0.3, p_2 = 1$时, A_1与Y具有较高的一致性, A_2为完全随机变量. 从图3(a)可以看到, SGini选择A_1的概率始终为1, 而Gini在A_2取值个数为25左右时, 随着A_2取值个数的增多, 选择A_1的概率从1开始下降, 即选择A_2的概率开始上升.
- 当$p_1 = 0.7, p_2 = 1$时, A_1与Y具有较低的一致性, A_2为完全随机变量. 从图3(b)可以看到, SGini选择A_1的概率为0.9左右, 而Gini选择A_1的概率随着A_2取值个数的增多呈现下降趋势, 在取值个数为30左右时, Gini选择A_2的概率达到1左右.

综上, 从图3(a)-图3(b)说明, Gini表现出多值偏向问题, 而SGini较好地缓解了Gini的多值偏向问题.

第二种情况设置A_2为与Y具有一定一致性的变量, A_1为完全随机变量. 如果评价方法倾向选择A_2, 则说明此方法能够选择出具有信息的属性. 具体地:

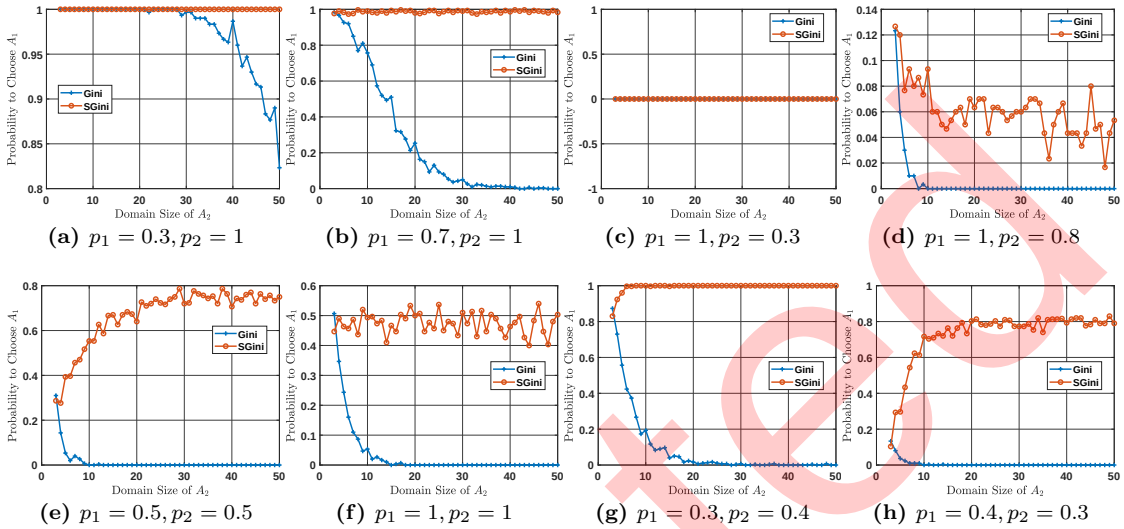
- 当$p_1 = 1, p_2 = 0.3$时, A_1为完全随机变量, A_2与Y具有较高的一致性. 从图3(c)可以看到, SGini与Gini倾向于选择A_2.
- 当$p_1 = 1, p_2 = 0.8$时, A_1为完全随机变量, A_2与Y具有较低的一致性. 从图3(d)可以看到, SGini与Gini倾向于选择A_2.

综上, 从图3(c)-图3(d)说明SGini能够选择出具有信息的属性.

第三种情况设置A_1、A_2为与Y具有相同或相近的一致性, 具体地:

- 当$p_1 = 0.5, p_2 = 0.5$时, A_1, A_2与Y的一致性相同. 从图3(e)可以看到, Gini选择A_2的概率随着A_2取值个数的增多而升高, 在$k = 10$, Gini选择A_1的概率降为0. 而SGini倾向于选择A_1, 但仍存在选择A_2的概率.
- 当$p_1 = 1, p_2 = 1$时, A_1与A_2均为完全随机变量, 从图3(f)可以看到, SGini选择A_1与A_2的概率相同, 而Gini倾向于选择A_2.
- 当$p_1 = 0.3, p_2 = 0.4$时, A_1的一致性略高于A_2, 从图3(g)可以看到, SGini倾向于选择A_1, 而Gini倾向于选择A_2.
- 当$p_1 = 0.4, p_2 = 0.3$时, A_2的一致性略高于A_1, 从图3(f)可以看到, SGini倾向于选择A_1, 但仍存在选择A_2的概率, 而Gini倾向于选择A_2, 在$k = 10$左右, 选择A_2的概率达到1.

综上, 从图3(e)-图3(h)说明相较于Gini, SGini在评价属性的重要度方面更具合理性.



(a) $p_1 = 0.3, p_2 = 1$ (b) $p_1 = 0.7, p_2 = 1$ (c) $p_1 = 1, p_2 = 0.3$ (d) $p_1 = 1, p_2 = 0.8$

(e) $p_1 = 0.5, p_2 = 0.5$ (f) $p_1 = 1, p_2 = 1$ (g) $p_1 = 0.3, p_2 = 0.4$ (h) $p_1 = 0.4, p_2 = 0.3$

图 3 标准基尼指数与基尼指数的选择倾向

Figure 3 The selection tendency of standard Gini index and the Gini index

5 缓解多值偏向的决策树算法的泛化性能比较

为了验证标准基尼指数的泛化性能, 本节在基准数据集上比较基于标准基尼指数的决策树算法与其他缓解多值偏向的决策树算法的分类性能.

5.1 对比方法

为了缓解基尼指数的多值偏向, 研究者使用不同的置换模型与分布统计量描述多值带来的重要程度. 本节列出四种具有代表性的方法, 这些方法将作为性能对比的基准方法. 其中, AIR方法与AGini方法直接计算属性的重要程度值; PIMP方法与$\triangle g_e$ 方法计算重要程度值对应的p值.

5.1.1 AIR方法

AIR方法[32]通过置换样本ID计算属性结构带来的重要度. 令特征索引集合为$\mathcal{O} = \{1, 2, .., d\}$, $\mathcal{P} = \{d+1, d+2, .., 2d\}$, 属性$X_i$, $i \in \mathcal{O}$ 为原始属性, 属性X_i, $i \in \mathcal{P}$为置换原始属性X_{i-d} 在样本上的排列得到的属性. 属性X_i的AIR重要度定义为:

$$AIR(X_i) = Im(X_i; i \in \mathcal{O}) - Im(X_i; i \in \mathcal{P}), \tag{27}$$

其中Im为信息增益、基尼指数等不纯度降低函数.

5.1.2 PIMP方法

PIMP方法[36]置换标签向量在样本上的排列, 假设多次置换之后重要度值形成高斯分布或伽马分布, 用分布下估计准则的$-\log p$值作为重要度评价准则. 令s为置换次数, 具体步骤如下:

Step1. 首先, 计算每个属性的评估准则值$Im(X_i; i \leqslant d)$;

Step2. 然后, 第t $(t \leqslant s)$ 次置换标签向量的位置得到一个伪标签向量, 计算i $(i \leqslant d)$属性对该伪标签向量决策的重要程度$R_{t,i}$;

Step3. 计算每个属性s次置换的均值与标准差, 分别记作$\mu_i = mean(R_{*,i})$与$\sigma_i = std(R_{*,i})$, 输出所有方差的均值$\sigma' = mean(\sigma_i)$;

Step4. 最后, 返回随机变量$Im(X_i)$在参数为$N(\mu_i, \max\{\sigma_i, \sigma'\})$ 的正态分布假设下对应的p 值. PIMP方法按照$-\log p$ 对属性的重要度进行排序.

5.1.3 $\triangle g_e$ 方法

$\triangle g_e$方法[34]假设n_{ij}服从多项分布, 并给出基尼指数的期望与方差如公式(12)所示. 进一步假设基尼指数服从伽马分布, 计算当前基尼指数在该分布下对应的p值:

$$p(\triangle g_e) = Gampdf\left(\frac{(\mathbb{E}(\triangle g_e))^2}{\mathbb{V}(\triangle g_e)}, \frac{Gini\mathbb{V}(\triangle g_e)}{\mathbb{E}(\triangle g_e)}\right), \tag{28}$$

$\triangle g_e$ 方法按照$-\log p$ 对属性的重要度进行排序.

5.1.4 AGini方法

AGini方法[33] 根据$\triangle g_e$方法给出的基尼指数的期望与方差, 进一步通过Cantelli不等式得到基尼指数的$1-\alpha$分位数:

$$AGini = Gini - \left(\mathbb{E}(\triangle g_e) + \sqrt{\frac{1-\alpha}{\alpha}\mathbb{V}(\triangle g_e)}\right). \tag{29}$$

AGini方法按照$AGini$对属性的重要度进行排序, 其中参数α的设置为$\alpha = 0.1$.

5.2 数据集描述及参数选择方法

为验证基于标准基尼指数的决策树算法的泛化性能, 采用来源于UCI或WEKA数据集的12组数据, 12组数据的详细信息如表4 所示. 表4的列分别对应数据集名称、样本个数、属性个数(属性取值种类数的最小值与最大值)及类别数(最小类样本比例与最大类样本比例).

对于存在连续属性的数据集, 本文采用经典的ChiMerge算法[45]对数据进行离散化. ChiMerge算法是一种基于卡方检验的有监督离散化方法, 该方法采用自下而上的策略合并相邻区间. 该算法主要包括两个阶段: 初始化阶段, 对所有样本连续属性的取值排序, 将每个实例单独作为一个区间. 合并阶段, 对每一对相邻的区间计算卡方值; 根据计算的卡方值, 选择其中最小卡方值的一对相邻区间进行合并; 不断重复上述步骤, 直到计算出的卡方值都不低于事先设定的阈值, 或者合并区间达到一定的数量. 本文的属性离散化个数设置为50以内的随机整数.

对于每个数据集, 按照7 : 3的比例划分为训练集和测试集, 再按照7 : 3的比例将训练集划分为训练集和验证集. 这样的划分进行五十次, 所有的算法在同一次划分下运行比较. 关于决策树算法停止条件的参数$minnum$的取值范围设置为$\{1, 2, .., 20\}$. 每一次划分下, 验证集上最大评估值对应的参数值作为模型参数. 为了评估算法的性能, 引入准确度指标Acc与纯准确度指标PAcc[39]:

$$Acc = \mathbb{P}(h(x) = y), \tag{30}$$

$$PAcc = \frac{Acc - RAcc}{1 - RAcc}, \tag{31}$$

其中$RAcc = \Sigma_{i=1}^{k}\mathbb{P}(h(x) = c_i)\mathbb{P}(y = c_i)$, $h(x)$为分类器, y为真实标签, c_i为第i类的标签.

表 4　基准数据集描述

Table 4　The description of benchmark data sets

ID	Name	Objects	Attributes	Class
1	Breast Cancer	699	9(9:11)	2(34.48:65.52)
2	Balance Scale Weight Distance	625	4(5:5)	3(7.84:46.08)
3	Molecular Biology Promoter Gene Sequences	106	57(4:4)	2(50.00:50.00)
4	Liver Disorders	345	6(1:16)	2(42.03:57.97)
5	Libras Movement	360	90(1:19)	15(6.67:6.67)
6	Musk(Version 1)	476	166(1:22)	2(43.49:56.51)
7	Zoo	100	16(2:6)	7(4.00:40.00)
8	Vehicle	846	18(4:29)	4(23.52:25.77)
9	Cardiotocography	2126	40(1:47)	10(2.49:27.23)
10	Segment	2310	19(1:44)	7(14.29:14.29)
11	Spambase	4601	57(3:68)	2(39.40:60.60)
12	Waveform Database Generator(Version 1)	5000	21(3:71)	3(32.94:33.92)

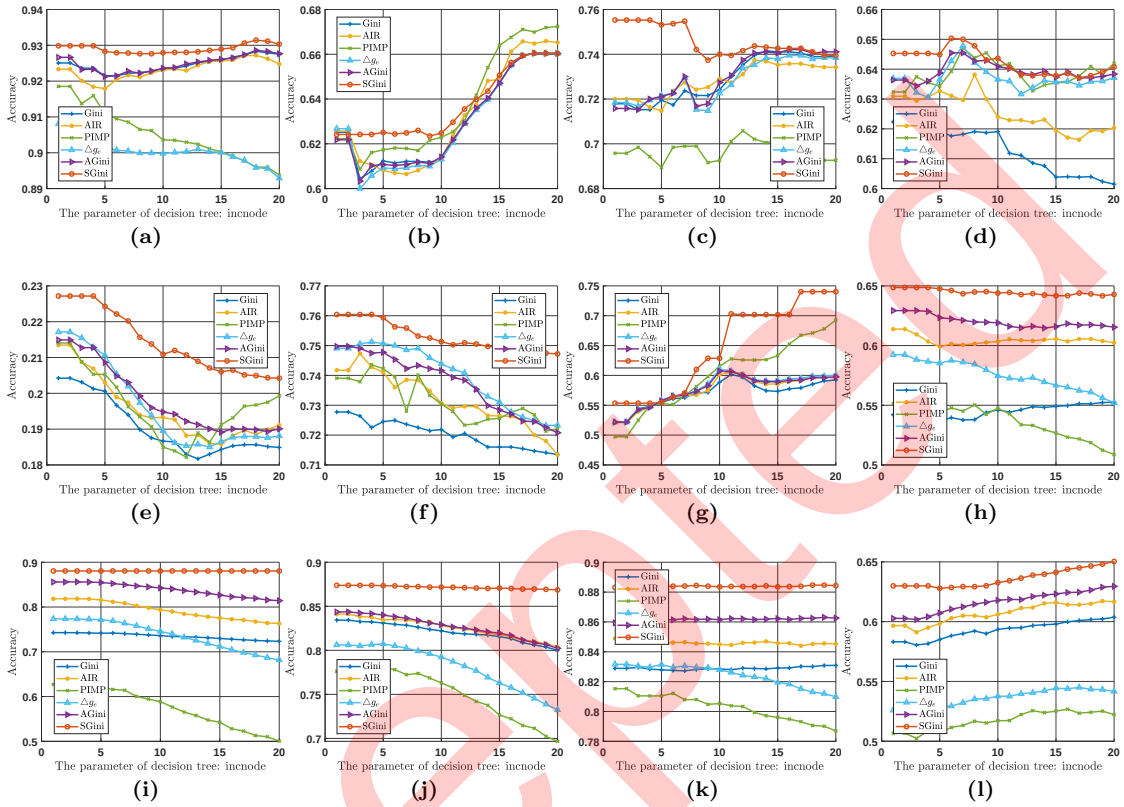


图 4 基准数据集上不同参数下的决策树算法准确度比较

Figure 4 Comparison of decision tree accuracy with different parameters on benchmark data sets

5.3 基准数据集分类性能比较

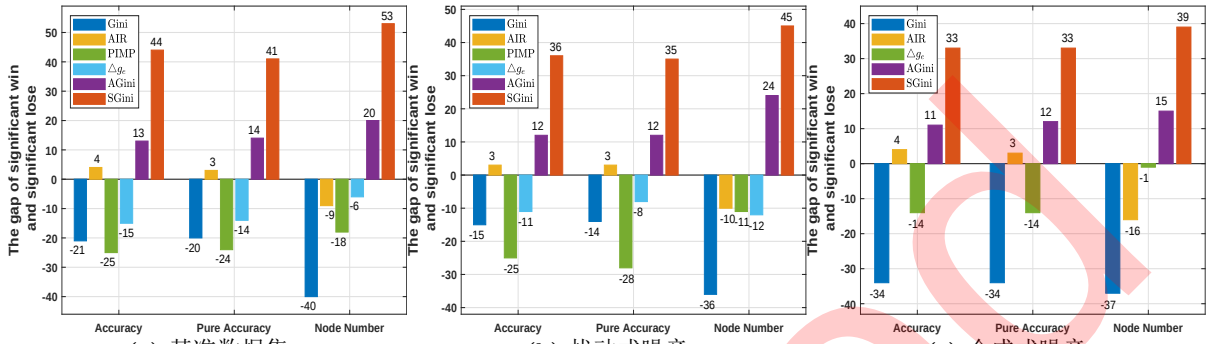
表5展示了基于SGini的决策树算法与基于Gini, AIR, PIMP, $\triangle g_e$, AGini的决策树算法的准确度值、纯准确度值以及每一层的平均节点数(除叶子节点外的节点个数除以树的层数). 每一行的黑体表明该方法取得最大平均值, 前五个方法数值后的黑色圆点表示基于SGini的决策树算法显著地好于该方法. 显著性检验方法为显著性水平为0.1的单侧符号检验. 表格的最后一行为每列方法在所有数据集上序值的平均值, 当方法在数据集上取得最大(准确度值、纯准确度)或最小(层平均节点数)平均值时, 其序值为1, 依次顺推, 当方法取得最小平均值时, 序值为6. 最后一行的平均序值越小, 该方法取得较大平均值的次数越多. 图4可视化了六个方法在每个参数取值下五十次实验的平均准确度.

从表5可以看到, 基于SGini的决策树算法在12个数据集上取得了最高的平均准确度与纯准确度值, 在11个数据集上取得最小的层平均节点数, 以及最小的平均序值. 从图4可以看到, 在不同的判停参数下, 除第二个数据集外, 基于SGini的决策树算法在不同的参数下取得了较高的准确度值. 这说明相较于其他缓解多值偏向的方法, 基于SGini的决策树算法倾向选择取值种类少的属性作为节点属性, 较好地缓解了决策树多值偏向问题, 并且具有较好的泛化性能.

表 5 基准数据集上决策树算法的性能比较

Table 5 Performance comparison of decision tree algorithms on benchmark data sets

Accuracy						
ID	Gini	AIR	PIMP	$\triangle g_e$	AGini	SGini
1	0.93 ±0.02 •	0.92 ±0.01 •	0.92 ±0.02 •	0.91 ±0.02 •	0.93 ±0.02	**0.93** ±0.02
2	0.63 ±0.03 •	0.63 ±0.03 •	0.62 ±0.03 •	0.63 ±0.03 •	0.62 ±0.03 •	**0.64** ±0.04
3	0.72 ±0.06 •	0.73 ±0.07 •	0.70 ±0.07 •	0.73 ±0.07 •	0.73 ±0.07 •	**0.76** ±0.06
4	0.62 ±0.04 •	0.63 ±0.04 •	0.64 ±0.04 •	0.64 ±0.04 •	0.64 ±0.04 •	**0.65** ±0.04
5	0.20 ±0.02 •	0.21 ±0.02 •	0.21 ±0.01 •	0.22 ±0.02 •	0.21 ±0.02 •	**0.23** ±0.02
6	0.73 ±0.04 •	0.74 ±0.04 •	0.75 ±0.04 •	0.75 ±0.03	0.75 ±0.03	**0.76** ±0.04
7	0.61 ±0.05 •	0.62 ±0.05 •	0.69 ±0.04 •	0.61 ±0.05 •	0.60 ±0.05 •	**0.74** ±0.04
8	0.54 ±0.03 •	0.61 ±0.03 •	0.55 ±0.03 •	0.59 ±0.05 •	0.63 ±0.02 •	**0.65** ±0.02
9	0.74 ±0.02 •	0.82 ±0.03 •	0.64 ±0.04 •	0.77 ±0.02 •	0.86 ±0.01 •	**0.88** ±0.00
10	0.83 ±0.01 •	0.84 ±0.01 •	0.78 ±0.01 •	0.80 ±0.01 •	0.84 ±0.01 •	**0.87** ±0.01
11	0.83 ±0.01 •	0.85 ±0.01 •	0.82 ±0.01 •	0.83 ±0.01 •	0.86 ±0.01 •	**0.88** ±0.01
12	0.58 ±0.01 •	0.60 ±0.01 •	0.51 ±0.02 •	0.53 ±0.01 •	0.60 ±0.01 •	**0.63** ±0.01
Ave. Rank	4.8333	3.3333	4.9167	4.0000	2.9167	1.0000
Pure Accuracy						
ID	Gini	PIMP	AIR	$\triangle g_e$	AGini	SGini
1	0.83 ±0.04 •	0.83 ±0.04	0.82 ±0.03 •	0.80 ±0.04 •	0.84 ±0.04	**0.85** ±0.03
2	0.36 ±0.05	0.36 ±0.04	0.35 ±0.04	0.36 ±0.04	0.35 ±0.04 •	**0.36** ±0.04
3	0.44 ±0.13 •	0.45 ±0.15 •	0.38 ±0.13 •	0.44 ±0.13 •	0.43 ±0.12 •	**0.51** ±0.11
4	0.24 ±0.09 •	0.26 ±0.07	0.27 ±0.08	0.27 ±0.08	0.27 ±0.08 •	**0.29** ±0.09
5	0.15 ±0.02 •	0.16 ±0.02 •	0.16 ±0.02 •	0.16 ±0.02 •	0.16 ±0.02 •	**0.17** ±0.02
6	0.45 ±0.07 •	0.48 ±0.08 •	0.48 ±0.08 •	0.49 ±0.06	0.49 ±0.06	**0.52** ±0.08
7	0.43 ±0.03 •	0.43 ±0.03 •	0.47 ±0.03 •	0.43 ±0.03 •	0.43 ±0.04 •	**0.52** ±0.02
8	0.39 ±0.05 •	0.49 ±0.04 •	0.40 ±0.04 •	0.45 ±0.06 •	0.50 ±0.03 •	**0.53** ±0.03
9	0.70 ±0.02 •	0.79 ±0.04 •	0.57 ±0.04 •	0.74 ±0.03 •	0.83 ±0.01 •	**0.86** ±0.00
10	0.81 ±0.01 •	0.81 ±0.01 •	0.74 ±0.02 •	0.77 ±0.01 •	0.82 ±0.01 •	**0.85** ±0.01
11	0.64 ±0.02 •	0.68 ±0.02 •	0.61 ±0.03 •	0.65 ±0.02 •	0.71 ±0.02 •	**0.76** ±0.02
12	0.37 ±0.02 •	0.39 ±0.02 •	0.26 ±0.02 •	0.29 ±0.02 •	0.40 ±0.02 •	**0.44** ±0.02
Ave. Rank	4.8333	3.6667	4.8333	3.6667	3.0000	1.0000
Average Number of Nodes in Each Layer						
ID	Gini	AIR	PIMP	$\triangle g_e$	AGini	SGini
1	14.50 ±2.53	13.87 ±1.85	16.34 ±2.06	16.84 ±1.90	12.85 ±1.78	**10.30** ±1.76
2	47.45 ±5.53	48.23 ±5.83	48.48 ±3.42	47.52 ±5.44	47.57 ±5.47	**22.74** ±5.58
3	4.85 ±0.96	4.69 ±1.00	5.20 ±0.96	4.36 ±0.80	4.42 ±0.83	**3.48** ±0.57
4	20.56 ±2.28	19.17 ±2.89	17.56 ±2.83	18.61 ±1.84	17.28 ±1.97	**14.46** ±1.74
5	38.91 ±2.31	28.90 ±3.56	28.66 ±3.05	28.04 ±3.71	25.91 ±3.52	**18.47** ±2.43
6	28.99 ±2.99	20.93 ±3.18	15.06 ±2.78	16.36 ±2.65	15.61 ±2.65	**11.05** ±1.66
7	4.09 ±0.41	3.75 ±0.57	2.73 ±0.36	3.54 ±0.63	3.54 ±0.62	**1.81** ±0.21
8	65.71 ±4.29	41.10 ±4.59	49.81 ±5.93	41.35 ±6.38	31.70 ±3.76	**26.43** ±2.62
9	42.68 ±5.89	16.83 ±7.12	68.32 ±12.38	30.88 ±3.89	7.82 ±1.42	**1.80** ±0.00
10	57.49 ±7.78	48.49 ±9.56	74.75 ±10.15	62.63 ±7.48	40.45 ±7.46	**29.60** ±3.32
11	89.58 ±21.02	49.72 ±12.36	34.70 ±4.63	30.85 ±9.79	**27.05** ±5.73	44.77 ±12.36
12	417.76 ±12.92	309.88 ±23.64	314.93 ±26.94	290.23 ±23.26	245.27 ±16.36	**218.79** ±10.59
Ave. Rank	5.1667	4.1667	4.4167	3.6667	2.3333	1.2500



(a) 基准数据集 (b) 扰动式噪音 (c) 合成式噪音

图 5 显著性检验

Figure 5 Significance test

图5展示了所提算法与基准算法的显著性差异对比结果. 图5中每个条形描述了相应算法显著性好与显著性差的次数之差. 算法a显著性好于算法b, 如果:

$$\mu_a(d) - 1.96\frac{\sigma_a(d)}{\sqrt{t}} > \mu_b(d) + 1.96\frac{\sigma_b(d)}{\sqrt{t}}, \tag{32}$$

其中$\mu_a(d)$与$\sigma_a(d)$分别是算法a性能评估值的平均值与标准差, t是实验次数, 1.96是标准正态分布0.975的概率值对应的逆累积分布函数值. 对于数据集$\mathcal{DS}$与算法集合$\mathcal{A}$, a显著性好于其他算法的次数定义为:

$$B_a := \sum_{d\in\mathcal{DS},b\in\mathcal{A}} \mathbb{I}\big[\mu_a(d) - 1.96\frac{\sigma_a(d)}{\sqrt{t}} > \mu_b(d) + 1.96\frac{\sigma_b(d)}{\sqrt{t}}\big], \tag{33}$$

a显著性差于其他算法的次数定义为:

$$W_a := \sum_{d\in\mathcal{DS},b\in\mathcal{A}} \mathbb{I}\big[\mu_a(d) - 1.96\frac{\sigma_a(d)}{\sqrt{t}} \leqslant \mu_b(d) + 1.96\frac{\sigma_b(d)}{\sqrt{t}}\big]. \tag{34}$$

图5的条形值分别对应于Accuracy与Pure Accuracy评价下的$B_a - W_a$, 以及层平均节点数负值下计算的$B_a - W_a$. 从图5(a)可以看出, 基于SGini的决策树算法性能显著性地好于其他算法, 层平均节点数显著性地少于其他算法.

5.4 噪音数据集分类性能比较

本节在扰动式噪音与合成式噪音数据集上验证基于SGini的决策树算法对于噪音的鲁棒性.

5.4.1 扰动式噪音数据集

为了验证模型的稳健性, 对属性添加了随机均匀扰动式噪音, 即对每一列属性, 随机选择一部分样本进行值的置换[46]. 这样的干扰式噪音产生的原因可能是数据收集时的随机错误或偏差等. 随机扰动式噪音常出现在离散属性中. 由于本文采用的数据都经过离散化, 相比于连续数据的高斯加性噪音, 随机扰动式噪音更符合算法对比采用的数据类型.

扰动式噪音数据集与基准数据集的实验设置相同, 表6及图6分别与5.3节中表5及图4的含义相同. 从表6可以看到, 基于SGini的决策树算法在10个与12个数据集上分别取得了最高的平均准确度与纯准确度值值, 在9个数据集上取得了最小的层平均节点数, 以及最小的平均序值. 从图6可以看到, 在不同

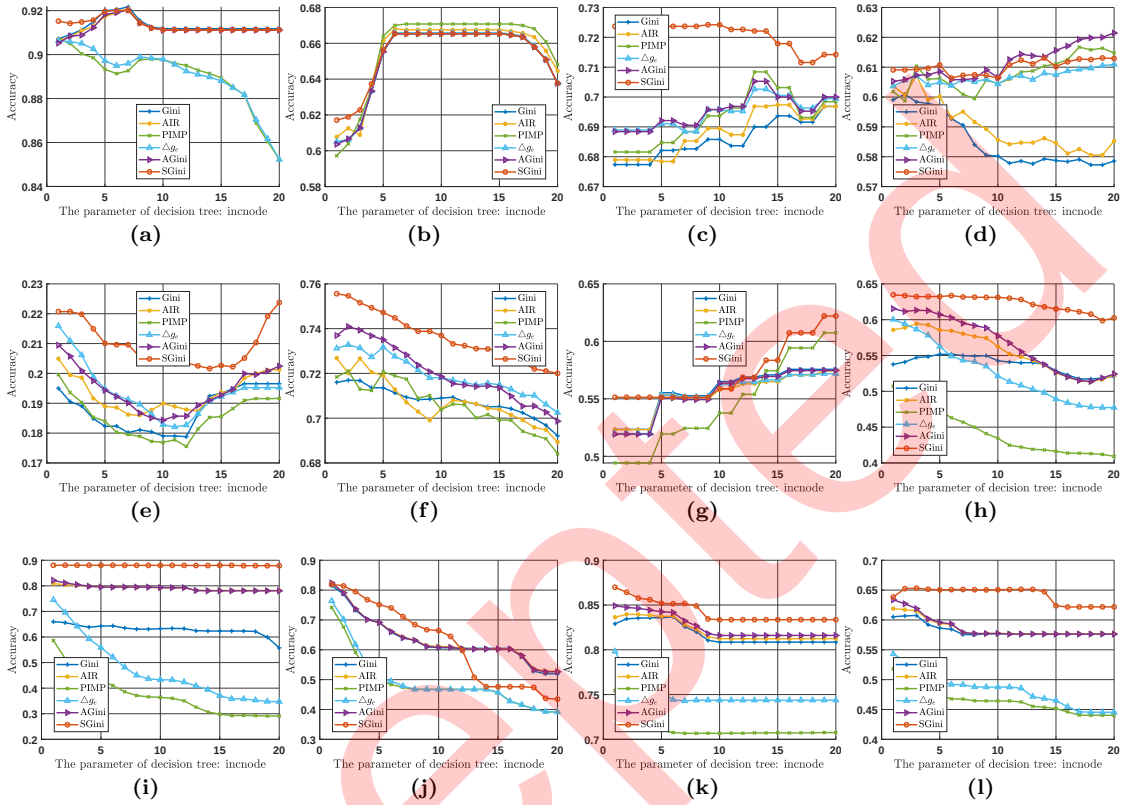


图 6　扰动式噪音水平30%时不同参数下的决策树算法准确度比较

Figure 6　Comparison of decision tree algorithm accuracy under different parameters at disturbed noise level of 30%

的判停参数下, 除第二个与第四个数据集外, 基于SGini的决策树算法在不同的参数下取得了较高的准确度值. 从图5(b)可以看出, 基于SGini的决策树算法性能显著性地好于其他算法, 层平均节点数显著性地少于其他算法. 这些说明基于SGini的决策树算法对于扰动式噪音具有较好的噪音鲁棒性与泛化性能.

5.4.2　合成式噪音数据集

为了进一步验证基于SGini决策树算法的泛化性, 对每个数据集的属性变量加一列噪音属性. 这个噪音属性列通过干扰某些样本上在原始属性取值得到的. 干扰方法为将样本的属性取值设置为在原始属性的最大值与最大值的三倍的范围内随机取值. 干扰比例设置为30%. 这样, 噪音属性相比原始属性具有更多的取值种类. 这样的噪音数据合成原理可能出现在下述场景中, 当对同一份调查问卷, 设置了两套不同的回答方式. 这两套回答方式的答案取值种类不同, 比如回答方式一的答案取值有限, 仅有固定的若干选项, 而回答方式二的答案取值无限, 除了回答方式一设置的固定选项, 还包含了一个开放式回答(例如通过被调查者自填答案). 当我们无法辨别哪种回答设置更合理, 更能反映出我们所关心的变量时, 可将两种回答方式下的答案同时输入分类器算法中进行学习. 此时, 回答方式二的数据类似于这里为每个属性合成的噪音属性列.

合成式噪音数据集的属性个数为原数据集的两倍, 导致PIMP算法时间成本太高, 在后四个数据集上的运行时间时间超过24小时(节点最小样本参数设置为1:1:20, 运行次数为50次). 从基准数据集的

表 6　扰动式噪音水平30%时决策树算法的性能比较

Table 6　Performance comparison of decision tree algorithm at disturbed noise level of 30%

Accuracy						
ID	Gini	AIR	PIMP	$\triangle g_e$	AGini	SGini
1	0.91 ±0.02	0.91 ±0.02	0.91 ±0.02 •	0.91 ±0.02 •	0.91 ±0.02 •	**0.92** ±0.02
2	0.64 ±0.04	**0.65** ±0.04	0.64 ±0.04	0.64 ±0.04 •	0.64 ±0.04 •	0.65 ±0.03
3	0.68 ±0.06 •	0.70 ±0.08	0.69 ±0.07 •	0.70 ±0.07 •	0.70 ±0.07 •	**0.72** ±0.06
4	0.60 ±0.04 •	0.60 ±0.04	0.60 ±0.04	0.61 ±0.04	0.61 ±0.03	**0.62** ±0.04
5	0.20 ±0.02 •	0.21 ±0.02 •	0.20 ±0.03 •	0.22 ±0.02 •	0.21 ±0.02 •	**0.23** ±0.02
6	0.72 ±0.04 •	0.73 ±0.04 •	0.72 ±0.04 •	0.73 ±0.04	0.74 ±0.04 •	**0.75** ±0.04
7	0.57 ±0.05 •	0.56 ±0.05 •	0.61 ±0.06	0.57 ±0.05 •	0.57 ±0.05 •	**0.63** ±0.06
8	0.54 ±0.03 •	0.59 ±0.03 •	0.51 ±0.05 •	0.60 ±0.05 •	0.62 ±0.03 •	**0.64** ±0.03
9	0.66 ±0.02 •	0.81 ±0.02 •	0.59 ±0.04 •	0.75 ±0.04 •	0.82 ±0.02 •	**0.88** ±0.00
10	0.81 ±0.01 •	0.82 ±0.01	0.74 ±0.02 •	0.76 ±0.01 •	**0.82** ±0.01	0.82 ±0.02
11	0.83 ±0.01 •	0.84 ±0.01 •	0.75 ±0.01 •	0.80 ±0.02 •	0.85 ±0.01 •	**0.87** ±0.01
12	0.61 ±0.01 •	0.62 ±0.01 •	0.52 ±0.02 •	0.54 ±0.01 •	0.63 ±0.01 •	**0.64** ±0.02
Ave. Rank	4.67	3.33	5.00	3.92	2.83	1.25
Pure Accuracy						
ID	Gini	AIR	PIMP	$\triangle g_e$	AGini	SGini
1	0.80 ±0.04 •	0.79 ±0.04 •	0.80 ±0.04 •	0.79 ±0.05	0.79 ±0.04 •	**0.82** ±0.04
2	0.34 ±0.05 •	0.34 ±0.05	0.32 ±0.04 •	0.34 ±0.05 •	0.33 ±0.05 •	**0.35** ±0.05
3	0.36 ±0.13 •	0.37 ±0.15 •	0.36 ±0.15 •	0.38 ±0.14 •	0.38 ±0.13 •	**0.45** ±0.13
4	0.20 ±0.09	0.20 ±0.07	0.20 ±0.08 •	0.21 ±0.09	0.21 ±0.08	**0.22** ±0.08
5	0.14 ±0.02 •	0.15 ±0.02 •	0.14 ±0.03 •	0.16 ±0.02	0.15 ±0.02 •	**0.16** ±0.02
6	0.42 ±0.08 •	0.44 ±0.07 •	0.43 ±0.08 •	0.46 ±0.08 •	0.46 ±0.08 •	**0.50** ±0.08
7	0.43 ±0.03 •	0.43 ±0.03 •	0.45 ±0.05	0.43 ±0.04 •	0.43 ±0.04 •	**0.46** ±0.04
8	0.38 ±0.04 •	0.45 ±0.04 •	0.34 ±0.08 •	0.47 ±0.07 •	0.49 ±0.04 •	**0.51** ±0.04
9	0.60 ±0.02 •	0.78 ±0.02 •	0.51 ±0.05 •	0.70 ±0.04 •	0.79 ±0.02 •	**0.86** ±0.00
10	0.78 ±0.02 •	0.79 ±0.01	0.70 ±0.02 •	0.72 ±0.02 •	**0.79** ±0.01	**0.79** ±0.02
11	0.64 ±0.02 •	0.65 ±0.02 •	0.48 ±0.03 •	0.58 ±0.04 •	0.68 ±0.02 •	**0.73** ±0.02
12	0.41 ±0.02 •	0.43 ±0.02 •	0.28 ±0.03 •	0.31 ±0.02 •	0.45 ±0.02	**0.46** ±0.02
Ave. Rank	4.42	3.42	5.17	3.92	2.92	1.17
Average Number of Nodes in Each Layer						
ID	Gini	AIR	PIMP	$\triangle g_e$	AGini	SGini
1	13.57 ±2.86	14.02 ±2.52	16.47 ±2.37	15.66 ±1.81	11.39 ±2.26	**10.02** ±1.76
2	32.37 ±6.68	32.06 ±7.88	31.42 ±6.72	32.25 ±6.79	32.06 ±6.89	**25.10** ±4.66
3	4.97 ±0.90	4.73 ±0.90	5.61 ±1.01	4.46 ±0.77	4.45 ±0.79	**3.65** ±0.53
4	21.00 ±2.48	19.60 ±3.24	19.65 ±2.29	19.16 ±2.29	18.13 ±1.68	**15.30** ±1.71
5	40.34 ±2.13	28.14 ±2.73	29.91 ±3.88	28.98 ±3.59	25.99 ±3.63	**18.59** ±2.35
6	28.69 ±3.14	20.22 ±3.07	14.27 ±3.10	16.25 ±2.79	15.54 ±2.40	**11.73** ±1.67
7	4.04 ±0.40	3.98 ±0.48	3.01 ±0.36	3.94 ±0.44	3.95 ±0.47	**2.16** ±0.26
8	57.50 ±6.73	37.57 ±4.47	45.81 ±7.00	36.22 ±5.63	29.66 ±3.12	**27.75** ±2.95
9	48.70 ±5.81	10.12 ±3.05	33.25 ±6.58	19.50 ±2.07	7.67 ±1.06	**1.81** ±0.05
10	56.80 ±7.17	50.57 ±6.41	65.55 ±8.40	65.39 ±7.99	**43.22** ±5.49	47.96 ±5.17
11	84.90 ±14.44	58.05 ±14.15	**19.41** ±3.84	28.96 ±11.55	20.66 ±4.24	46.97 ±6.20
12	237.80 ±31.04	174.85 ±8.62	122.74 ±14.17	125.84 ±12.12	**114.65** ±9.48	145.67 ±14.83
Ave. Rank	5.50	4.08	3.92	3.75	2.17	1.58

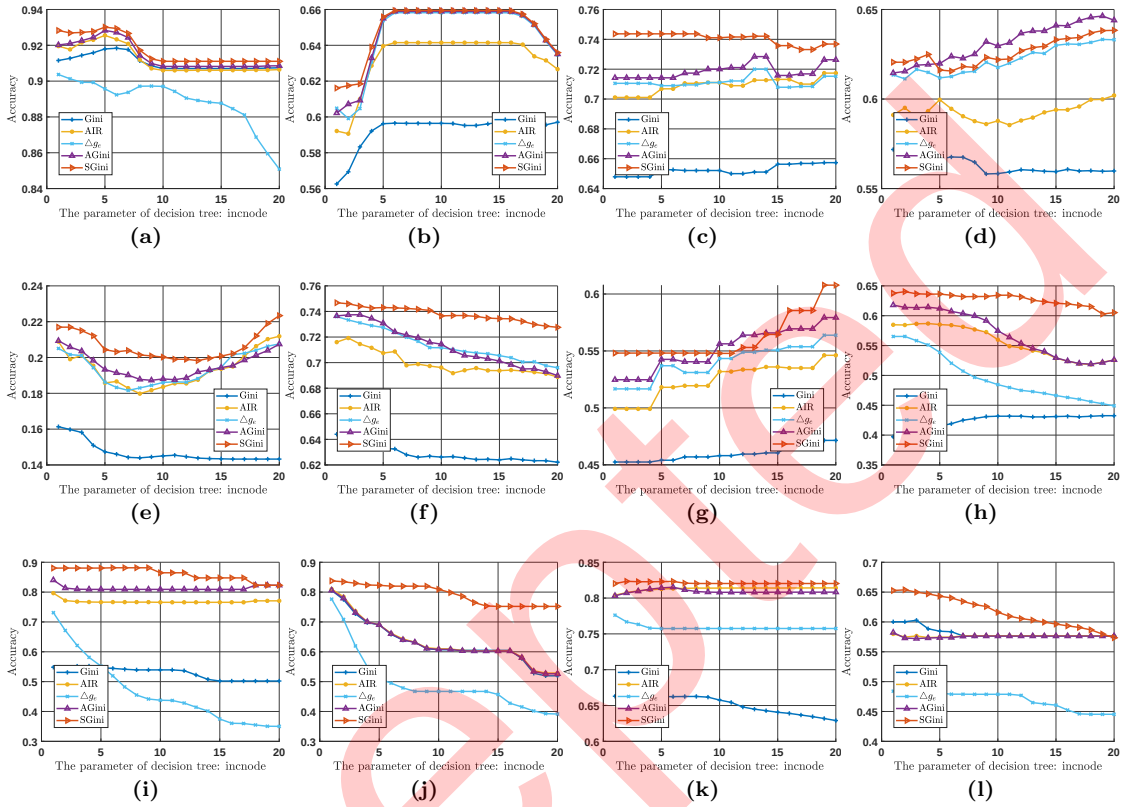


图 7　合成式噪音水平30%时不同参数下的决策树算法准确度比较

Figure 7　Comparison of decision tree algorithm accuracy under different parameters at the synthetic noise level of 30%

对比实验结果来看, PIMP算法的性能较差. 因此, 在合成式数据集上, PIPM算法将不作为基准算法进行对比. 表7及图7分别与5.3节中表5及图4的含义相同. 表8展示了算法选择合成式噪音属性作为节点特征的比例. 从表7可以看到, 基于SGini的决策树算法在11个与12个数据集上分别取得了最高的平均准确度与纯准确度值, 在9个数据集上取得了最小的层平均节点数, 以及最小的平均序值. 从图7可以看到, 在不同的判停参数下, 除第四个数据集外, 基于SGini的决策树算法在不同的参数下取得了较高的准确度值. 从表8可以看到, 在7个数据集上获得了最低噪音属性选择比, 并且获得了最小序值. 从图5(c)可以看出, 基于SGini的决策树算法性能显著性地好于其他算法, 层平均节点数显著性地少于其他算法. 这些说明基于SGini的决策树算法对于合成式噪音具有较好的噪音鲁棒性与泛化性能.

5.5　图像数据集分类性能比较

除了基准数据集, 本文在真实场景的图像数据集上做了进一步实验, 验证所提标准基尼指数与决策树算法的性能. 图像数据集采用机器视觉领域常用的STL-10[47]与ImageNet-10[48]. STL-10包含13000张彩色图像, 这些图像被分为10种不同类别, 每种类别有1300张图像, 其中500张被用作训练集, 800张用作测试集. ImageNet-10包含13500张彩色图像, 这些图像被分成10个类别, 每种类别有1350张图像, 其中1300张被用作训练集, 50张用作测试集.

对于STL-10与ImageNet-10, 本文采用自监督学习模型MOCO V3[49]提取数据的特征表示. MOCO V3为对比学习中较为前沿的方法, 其大致流程如图8所示. MOCO V3网络主要由编码器(Encoder)和

表 7　合成式噪音水平30%时决策树算法的性能比较

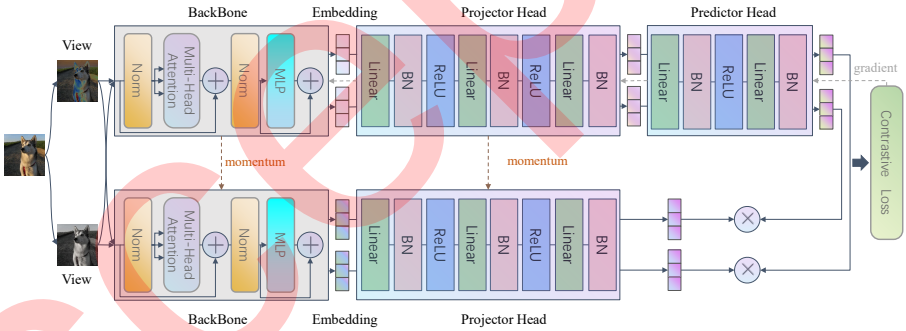
Table 7　Performance comparison of decision tree algorithm at the synthetic noise level of 30%

Accuracy					
ID	Gini	AIR	$\triangle g_e$	AGini	SGini
1	0.92 ±0.02 •	0.92 ±0.02 •	0.90 ±0.02 •	0.93 ±0.02	**0.93** ±0.02
2	0.57 ±0.03 •	0.61 ±0.04 •	0.62 ±0.04	**0.63** ±0.05	0.63 ±0.04
3	0.65 ±0.07 •	0.71 ±0.07 •	0.71 ±0.08 •	0.72 ±0.07	**0.74** ±0.06
4	0.57 ±0.05 •	0.59 ±0.05 •	0.62 ±0.05	0.62 ±0.05	**0.62** ±0.05
5	0.16 ±0.01 •	0.22 ±0.03 •	0.20 ±0.02 •	0.21 ±0.03 •	**0.23** ±0.03
6	0.65 ±0.03 •	0.72 ±0.05 •	0.73 ±0.03 •	0.73 ±0.03	**0.75** ±0.04
7	0.47 ±0.07 •	0.52 ±0.07 •	0.55 ±0.07 •	0.56 ±0.06 •	**0.60** ±0.06
8	0.40 ±0.04 •	0.59 ±0.03 •	0.57 ±0.05 •	0.62 ±0.02 •	**0.64** ±0.03
9	0.55 ±0.06 •	0.80 ±0.04 •	0.73 ±0.03 •	0.84 ±0.01 •	**0.88** ±0.00
10	0.81 ±0.01 •	0.81 ±0.01 •	0.78 ±0.02 •	0.81 ±0.01 •	**0.84** ±0.01
11	0.66 ±0.07 •	0.81 ±0.01 •	0.78 ±0.01 •	0.81 ±0.01 •	**0.82** ±0.01
12	0.60 ±0.01 •	0.58 ±0.01 •	0.48 ±0.01 •	0.58 ±0.01 •	**0.65** ±0.02
Ave. Rank	4.5833	3.2500	3.6667	2.4167	1.0833
Pure Accuracy					
ID	Gini	AIR	$\triangle g_e$	AGini	SGini
1	0.81 ±0.05 •	0.82 ±0.04 •	0.78 ±0.04 •	0.83 ±0.04 •	**0.84** ±0.03
2	0.26 ±0.05 •	0.31 ±0.05 •	0.33 ±0.04 •	0.33 ±0.05 •	**0.35** ±0.05
3	0.30 ±0.14 •	0.42 ±0.15 •	0.42 ±0.14 •	0.43 ±0.12 •	**0.49** ±0.12
4	0.13 ±0.09 •	0.17 ±0.10 •	0.21 ±0.10	0.22 ±0.09 •	**0.23** ±0.10
5	0.10 ±0.01 •	0.15 ±0.02 •	0.15 ±0.02 •	0.15 ±0.02 •	**0.16** ±0.02
6	0.27 ±0.07 •	0.42 ±0.10 •	0.47 ±0.06	0.47 ±0.06	**0.49** ±0.08
7	0.33 ±0.05 •	0.39 ±0.05 •	0.42 ±0.05 •	0.42 ±0.04 •	**0.47** ±0.04
8	0.20 ±0.04 •	0.45 ±0.05 •	0.42 ±0.06 •	0.49 ±0.03 •	**0.52** ±0.04
9	0.47 ±0.07 •	0.76 ±0.05 •	0.69 ±0.04 •	0.81 ±0.02 •	**0.86** ±0.00
10	0.77 ±0.02 •	0.78 ±0.02 •	0.74 ±0.02 •	0.77 ±0.02 •	**0.81** ±0.01
11	0.18 ±0.18 •	0.60 ±0.02 •	0.53 ±0.03 •	0.59 ±0.02 •	**0.63** ±0.02
12	0.40 ±0.02 •	0.37 ±0.02 •	0.23 ±0.02 •	0.37 ±0.02 •	**0.48** ±0.02
Ave. Rank	4.5833	3.3333	3.7500	2.3333	1.0000
Average Number of Nodes in Each Layer					
ID	Gini	AIR	$\triangle g_e$	AGini	SGini
1	17.68 ±4.92	15.98 ±3.88	15.28 ±1.39	12.07 ±2.44	**10.54** ±2.05
2	42.43 ±5.53	32.76 ±5.48	31.72 ±6.05	27.84 ±3.82	**26.50** ±3.73
3	8.24 ±2.04	5.30 ±1.07	4.57 ±0.87	4.49 ±0.78	**3.55** ±0.70
4	28.49 ±2.64	19.16 ±2.86	15.85 ±1.88	13.75 ±1.56	**13.36** ±2.44
5	48.74 ±6.11	28.83 ±3.68	29.91 ±3.21	26.20 ±4.08	**18.93** ±2.53
6	41.88 ±6.57	22.07 ±3.46	17.44 ±3.16	16.10 ±2.70	**11.31** ±1.90
7	7.53 ±1.29	4.36 ±1.18	3.69 ±0.72	3.65 ±0.72	**1.93** ±0.25
8	72.09 ±4.31	39.64 ±4.62	39.15 ±5.69	**27.80** ±4.04	27.96 ±3.05
9	59.76 ±13.86	10.07 ±3.70	24.70 ±3.41	5.40 ±0.61	**1.80** ±0.00
10	64.05 ±8.72	47.59 ±8.02	50.88 ±7.73	41.92 ±9.93	**18.94** ±3.83
11	**6.07** ±3.00	48.66 ±13.07	22.82 ±5.24	89.23 ±18.09	21.04 ±4.55
12	189.96 ±7.56	59.89 ±11.40	42.19 ±5.84	**20.14** ±5.40	20.29 ±3.61
Ave. Rank	4.6667	3.7500	3.2500	2.0833	1.2500

表 8　合成式噪音水平30%时决策树算法的噪音属性选择比例

Table 8　Noise attribute selection ratio of decision tree algorithm at the synthetic noise level of 30%

	Noise Ratio				
ID	Gini	AIR	$\triangle g_e$	AGini	SGini
1	0.33 ±0.17	0.23 ±0.17	**0.03** ±0.06	0.06 ±0.10	0.06 ±0.10
2	0.71 ±0.16	0.24 ±0.17	0.09 ±0.12	0.09 ±0.14	**0.08** ±0.13
3	0.39 ±0.21	0.23 ±0.15	0.06 ±0.11	0.03 ±0.08	**0.02** ±0.08
4	0.81 ±0.22	0.54 ±0.15	0.47 ±0.17	0.49 ±0.17	**0.46** ±0.17
5	0.53 ±0.39	0.07 ±0.18	**0.03** ±0.09 •	0.05 ±0.09	0.06 ±0.11
6	0.50 ±0.28	0.22 ±0.13	0.18 ±0.13	0.13 ±0.10	**0.09** ±0.09
7	0.40 ±0.15	0.14 ±0.15	0.10 ±0.16	**0.07** ±0.13 •	0.11 ±0.13
8	0.86 ±0.26	0.27 ±0.11	0.23 ±0.08	**0.18** ±0.10	**0.18** ±0.12
9	0.40 ±0.16	0.04 ±0.06	0.06 ±0.03	0.00 ±0.01	**0.00** ±0.00
10	0.16 ±0.05 •	0.19 ±0.07 •	**0.14** ±0.05 •	0.22 ±0.08 •	0.31 ±0.13
11	0.75 ±0.12	0.40 ±0.20 •	0.60 ±0.11	**0.37** ±0.18 •	0.55 ±0.21
12	0.92 ±0.08 •	0.67 ±0.20 •	0.66 ±0.22 •	0.88 ±0.14	**0.35** ±0.04
Ave. Rank	4.7500	3.5833	2.4167	2.2500	2.0000



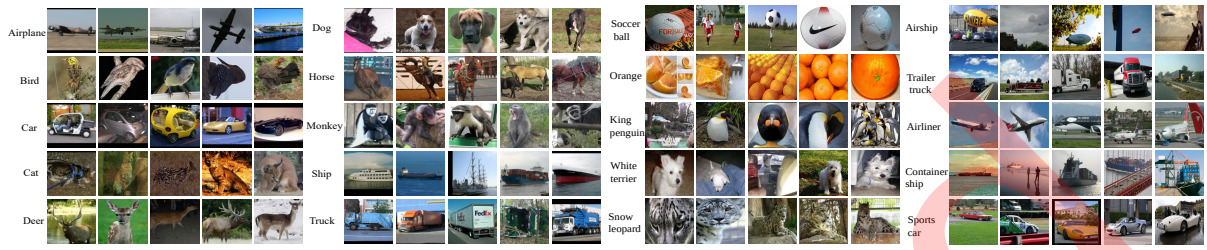
(a) MOCO V3网络

图 8　图像数据集特征提取器示意图

Figure 8　Diagram of image data set feature extractor

动量编码器(Momentum Encoder)两部分组成. 编码器采用视觉转换器(Visual Transformer)作为骨干网络, 并在其后接入了映射头(Projector head)和预测头(Predictor head); 动量编码器由视觉转换器和映射头组成. MOCO V3将同一张图片的两种增强视图经由两个网络的表示作为正样本对, 将增强视图和大批量内所有样本作为负样本对, 采用InfoNCE损失训练模型, 更新编码器的参数, 并通过动量方法更新动量编码器参数. 本文采用He等人[49]等人训练好的MOCO V3模型, 将STL-10与ImageNet-10的每张图片表示为256*1的实数向量. 对于图像数据集提取的每一列实数特征向量, 先采取基于卡方统计量的层次离散化方法Chimerge算法[45]进行分块离散化. 对于STL-10与ImageNet-10两个图像数据集, 每一列特征离散化的块数分别从$[2, 10]$与$[2, 40]$之间随机选择.

为进一步提高决策树算法对图像数据集的适应性, 采用Bagging的集成策略提升树的性能, 集成个数设置为21. 设置单颗树的节点最小样本参数设置为5, 算法运行5次进行对比. 图像数据集经过MOCO V3特征提取之后, 维度较高, 导致$\triangle g_e$与PIMP算法时间成本太高, 在后四个数据集上的运



(a) STL-10 (b) ImageNet-10

图 9　图像数据集示意图

Figure 9　Display of the image data sets

表 9　图像数据集决策树算法的性能比较

Table 9　Performance comparison of decision tree algorithms on image data sets

STL-10				
Measure	Gini	AIR	AGini	SGini
Accuracy	0.8638 ±0.0044	0.8673 ±0.0070	0.8776 ±0.0056	**0.9106** ±0.0017
Pure Accuracy	0.8486 ±0.0049	0.8525 ±0.0078	0.8639 ±0.0062	**0.9006** ±0.0019
Average Node	844.4381 ±3.7992	768.9238 ±6.9872	719.6857 ±3.1441	**509.7619** ±5.8037
Tree Depth	8.9333 ±0.1263	9.1238 ±0.1582	9.4952 ±0.2142	**15.9905** ±0.3594
Tree Time(s)	**223.0507** ±28.3694	241.3341 ±28.4778	223.7008 ±27.8236	345.3840 ±28.4345
ImageNet-10				
Measure	Gini	AIR	AGini	SGini
Accuracy	0.9173 ±0.0102	0.9227 ±0.0062	0.9200 ±0.0082	**0.9427** ±0.0055
Pure Accuracy	0.9081 ±0.0113	0.9142 ±0.0068	0.9111 ±0.0091	**0.9364** ±0.0061
Average Node	886.2762 ±21.5499	784.7905 ±18.5932	739.8667 ±19.0862	**517.5905** ±8.4759
Tree Depth	9.2476 ±0.2029	9.4857 ±0.1457	9.5714 ±0.2793	**15.2095** ±0.1822
Tree Time(s)	1120.6451 ±41.7773	1163.2559 ±67.0833	**1027.6510** ±51.4554	1359.8604 ±88.0581

行时间时间超过72小时(节点最小样本参数设置为5, 运行次数为5次, 集成21棵树). 因此, 在图像数据集上, $\triangle g_e$与PIPM算法将不作为基准算法进行对比. 表9展示了对比的准确度值, 纯准确度值, 层平均节点数, 树的深度及建树时间. 从表9可以看出, 基于SGini的算法得到了最小的平均节点个数, 最深的决策树, 建树时间长, 取得了最高的准确度, 纯准确度值. 这说明, SGini算法不偏向于选择取值属性多的特征作为节点, 不过早将样本地被分配到树的上层节点当中, 支持更多的样本参与下层树的构建. 这样得到的决策规则才有可能考虑更多特征与更细粒度的划分, 从而获得较好的泛化性能. 综上, 基于SGini的的决策树算法能够很好地适用于图像数据集的分类问题.

6　结论

决策树算法是机器学习领域的经典算法, 该算法通常基于不纯度函数定义节点属性评价准则. 本文在多分类任务下, 证明了基于凹函数的不纯度函数具有多值偏向问题, 并指出固定边缘分布的列联表元素服从超几何分布, 在超几何分布假设下, 给出基尼指数期望与方差关于边缘分布的表达式. 这极大程度低降低了基尼指数期望与方差的计算复杂度. 接着, 在标准化框架下, 定义了缓解随机一致性

的标准化基尼指数, 并提出了基于标准化基尼指数的决策树算法. 最后, 在基准数据集, 噪音数据集上与图像数据集上, 验证了所提算法的泛化性能与缓解多值偏向的有效性.

决策树算法每次构建内部节点时, 需对当前所有特征进行重要度评估. 当数据的维度较高时, 提高决策树算法时效性的关键在于加速特征重要度的评估速度. 目前, 深度学习作为非结构化数据的表示方法已取得了成功应用. 而深度学习的输出一般为高维向量. 因此, 如何高效处理高维数据是提升本文所提方法的一个重要方向.

在未来研究中, 探索标准化基尼指数与提升泛化性能间的内在关系是一个值得研究的方向. 评估准则或数据的标准化是机器学习领域常用的策略. 在深度学习领域, 数据的标准化已被证明可通过改善损失函数海森矩阵的条件数来提高算法的收敛速度[43]. 本文在标准化框架下定义了标准基尼指数. 在此基础上, 揭示诸如基尼指数等评价准则的标准化对泛化性能的效用机理对机器学习理论研究的发展具有一定科学意义.

7 附录

性质1 基于公式(5), (6), (7)定义的不纯度函数存在多值偏向问题.

性质1的证明 根据公式(6), (7), 特征A与A'之间重要度的差值为:

$$\Delta Im(A)-\Delta Im(A') \tag{35}$$
$$=-p(a_r)Im(a_r)+p(a'_r)Im(a'_r)+p(a'_{r+1})Im(a'_{r+1})$$
$$=-p(a_r)\phi(p(c_1|a_r),p(c_2|a_r),...,p(c_k|a_r)) \tag{36}$$
$$+p(a'_r)\phi(p(c_1|a'_r),p(c_2|a'_r),...,p(c_k|a'_r))+p(a'_{r+1})\phi(p(c_1|a'_{r+1}),p(c_2|a'_{r+1}),...,p(c_k|a'_{r+1}))$$

显然有, $p(a'_r)+p(a'_{r+1})=p(a_r)$. 令$t=\frac{p(a'_r)}{p(a_r)}$, 有$1-t=\frac{p(a'_{r+1})}{p(a_r)}$. 进一步令$x_j=p(c_j|a_r)$, $p_j=p(c_j|a'_r)$, $q_j=p(c_j|a'_{r+1})$, 有

$$x_j=\frac{p(c_j,a_r)}{p(a_r)} \tag{37}$$
$$=\frac{p(c_j,a'_r)}{p(a_r)}+\frac{p(c_j,a'_{r+1})}{p(a_r)} \tag{38}$$
$$=\frac{p(a'_r)}{p(a_r)}\frac{p(c_j,a'_r)}{p(a'_r)}+\frac{p(a'_{r+1})}{p(a_r)}\frac{p(c_j,a'_{r+1})}{p(a'_{r+1})} \tag{39}$$
$$=tp_j+(1-t)q_j. \tag{40}$$

根据上述关系及公式(5), 有:

$$\frac{\Delta Im(A)-\Delta Im(A')}{p(a_r)} \tag{41}$$
$$=-\Sigma_{j=1}^{k}f(x_j)+t\Sigma_{j=1}^{k}f(p_j)+(1-t)\Sigma_{j=1}^{k}f(q_j) \tag{42}$$
$$=-\Sigma_{j=1}^{k}f(tp_j+(1-t)q_j)+t\Sigma_{j=1}^{k}f(p_j)+(1-t)\Sigma_{j=1}^{k}f(q_j)\leqslant 0. \tag{43}$$

其中最后一个不等式根据f是凹函数, 具有性质$f(rp_j+(1-r)q_j)\geqslant rf(p_j)+(1-r)f(q_j)$, $\forall r\in[0,1]$. 根据上述分析可知, 基于凹函数的和定义的不纯度函数均具有多值偏向问题. □

引理1 当$\mathcal{A}_{perm}$中的特征向量A'边缘分布及取值个数固定时, 其与Y形成的列联表元素n_{ij}服从参数为N, n_j^Y, n_i^A的超几何分布. 即从N个样本(其中包含n_j^Y个第j类样本）中不放回地抽出n_i^A个样本, 抽到n个第j类样本的概率. 记作$n_{ij} \sim H(N, n_j^Y, n_i^A)$, 即:

$$\mathbb{P}_{A\in\mathcal{A}_{perm}}\left(n_{ij}=n\right)=\frac{\mathbf{C}_{n_j^Y}^{n}\mathbf{C}_{N-n_j^Y}^{n_i^A-n}}{\mathbf{C}_N^{n_i^A}}, \tag{44}$$

其中$n=\max\{0, n_i^A+n_j^Y-N\}, ..., \min\{n_i^A, n_j^Y\}$, $\mathbf{C}_n^r$为从n个样本中取r个样本的组合数.

引理1的证明 在置换集合中, 当固定属性边缘分布时, 取特定属性值的样本个数是固定的, 而哪些样本取特定属性值是不固定的, 那么样本的属性取值与其类别无关. 即不同类的样本属性取同一值的概率是相同的, 记为p. 那么,

$$\mathbb{P}_{A\in\mathcal{A}_{perm}}\left(n_{ij}=n\right)=\mathbb{P}\left(n_{ij}=n\Big|\sum_j n_{ij}=n_i^A\right) \tag{45}$$

$$=\frac{\mathbb{P}(n_{ij}=n, \sum_j n_{ij}=n_i^A)}{\mathbb{P}(\sum_j n_{ij}=n_i^A)} \tag{46}$$

$$=\frac{\mathbb{P}(n_{ij}=n)\mathbb{P}(\sum_{j'\neq j} n_{ij'}=n_i^A-n)}{\mathbb{P}(\sum_j n_{ij}=n_i^A)} \tag{47}$$

$$=\frac{\mathbf{C}_{n_j^Y}^{n}p^n(1-p)^{n_j^Y-n}\mathbf{C}_{N-n_j^Y}^{n_i^A-n}p^{n_i^A-n}(1-p)^{N-n_j^Y-n_i^A+n}}{\mathbf{C}_N^{n_i^A}p^{n_i^A}(1-p)^{N-n_i^A}} \tag{48}$$

$$=\frac{\mathbf{C}_{n_j^Y}^{n}\mathbf{C}_{N-n_j^Y}^{n_i^A-n}}{\mathbf{C}_N^{n_i^A}}.\square \tag{49}$$

文中3.1节中引理2给出超几何分布随机变量l次多项式的期望的计算方式, 进一步, 根据n^l与$(n)_l$的关系:

$$\begin{aligned} n^4&=(n)_4+6(n)_3+7(n)_2+n,\\ n^3&=(n)_3+3(n)_2+n,\\ n^2&=(n)_2+n. \end{aligned} \tag{50}$$

便可得到公式(16), 公式(24), 公式(25), 公式(26)的证明, 其中公式(25)与公式(24)的证明方式相同.

公式(16)的证明

根据引理2以及$n^2=(n)_2+(n)_1$, 得到置换集合中基尼指数的期望值为:

$$\mathbb{E}_{perm}Gini(A)=-\Sigma_{j=1}^k\left(\frac{n_j^Y}{N}\right)^2+\Sigma_{i,j}\Sigma_{n\leqslant\min\{n_i^A,n_j^Y\}}\frac{n^2}{n_i^AN}\mathbb{P}(n_{ij}=n) \tag{51}$$

$$=-\Sigma_{j=1}^k\left(\frac{n_j^Y}{N}\right)^2+\Sigma_{i,j}\frac{n_j^Y}{N^2}\left(\frac{(n_j^Y-1)(n_i^A-1)}{N-1}+1\right) \tag{52}$$

$$=\frac{r-1}{N-1}\left(1-\Sigma_{j=1}^k\left(\frac{n_j^Y}{N}\right)^2\right) \tag{53}$$

公式(24)的证明

根据引理2及公式(50), 公式(22)中第二项可表示为:

$$\begin{aligned}
&\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{n}\frac{n^2}{Nn_i^A}\mathbb{P}(n_{ij}=n)\sum_{i'\neq i}^{r}\sum_{n'}\frac{n'^2}{Nn_{i'}^A}\mathbb{P}(n_{i'j}=n'|n_{ij}=n)\\
&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{i'\neq i}^{r}\sum_{n}\frac{n^2}{N^2n_i^An_{i'}^A}\mathbb{P}(n_{ij}=n)\left(\frac{(n_{i'}^A)_2(n_j^Y-n)_2}{(N-n_i^A)_2}+\frac{n_{i'}^A(n_j^Y-n)}{N-n_i^A}\right)\\
&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{i'\neq i}^{r}\frac{1}{N^2n_i^An_{i'}^A}\sum_{n}\mathbb{P}(n_{ij}=n)\left(C_4n^4+C_3n^3+C_2n^2\right)\\
&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{i'\neq i}^{r}\frac{1}{N^2n_i^An_{i'}^A}\sum_{n}\mathbb{P}(n_{ij}=n)\left(\overline{C}_4(n)_4+\overline{C}_3(n)_3+\overline{C}_2(n)_2+\overline{C}_1n\right)\\
&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{i'\neq i}^{r}\frac{1}{N^2n_i^An_{i'}^A}\left(\overline{C}_4\frac{(n_i^A)_4(n_j^Y)_4}{(N)_4}+\overline{C}_3\frac{(n_i^A)_3(n_j^Y)_3}{(N)_3}+\overline{C}_2\frac{(n_i^A)_2(n_j^Y)_2}{(N)_2}+\overline{C}_1\frac{n_i^An_j^Y}{N}\right),
\end{aligned}\tag{54}$$

该式只包含对n_i^A, $n_{i'}^A$, n_j^Y的运算. 其中$\overline{C}_1=C_2+C_3+C_4$, $\overline{C}_2=C_2+3C_3+7C_4$, $\overline{C}_3=C_3+6C_4$, $\overline{C}_4=C_4$, $C_1=\frac{n_{i'}^A}{N-n_i^A}$, $C_4=\frac{(n_{i'}^A)_2}{(N-n_i^A)_2}$, $C_2=(C_1-C_4)n_j^Y+C_4(n_j^Y)^2$, $C_3=-2n_j^YC_4+C_4-C_1$.

公式(26)的证明 根据公式(50)及引理2, 公式(22)中第四项可表示为:

$$\begin{aligned}
&\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{n}\frac{n^2}{Nn_i^A}\mathbb{P}(n_{ij}=n)\sum_{j'\neq j}^{k}\mathbb{P}(n_{ij'}=n'|n_{ij}=n)\sum_{i'\neq i}^{r}\sum_{n''}\frac{n''^2}{Nn_{i'}^A}\mathbb{P}(n_{i'j'}=n''|n_{ij'}=n',n_{ij}=n)\\
&=\sum_{i=1}^{r}\sum_{j=1}^{k}\frac{n^2}{N^2n_i^An_{i'}^A}\mathbb{P}(n_{ij}=n)\sum_{j'\neq j}^{k}\sum_{i'\neq i}^{r}\mathbb{P}(n_{ij'}=n'|n_{ij}=n)\left(\frac{(n_{j'}^Y-n')n_{i'}^A}{N-n_i^A}+\frac{(n_{j'}^Y-n')_2(n_{i'}^A)_2}{(N-n_i^A)_2}\right)\\
&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{j'\neq j}^{k}\sum_{i'\neq i}^{r}\frac{n^2}{N^2n_i^An_{i'}^A}\mathbb{P}(n_{ij}=n)\left(C_2(n_i^A-n)_2+C_1(n_i^A-n)+C_0\right)\\
&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{j'\neq j}^{k}\sum_{i'\neq i}^{r}\frac{1}{N^2n_i^An_{i'}^A}\mathbb{P}(n_{ij}=n)\overline{\overline{C}}_4n^4+\overline{\overline{C}}_3n^3+\overline{\overline{C}}_2n^2\\
&=\sum_{i=1}^{r}\sum_{j=1}^{k}\sum_{j'\neq j}^{k}\sum_{i'\neq i}^{r}\frac{1}{N^2n_i^An_{i'}^A}\left(\overline{\overline{C}}_4\frac{(n_i^A)_4(n_j^Y)_4}{(N)_4}+\overline{\overline{C}}_3\frac{(n_i^A)_3(n_j^Y)_3}{(N)_3}+\overline{\overline{C}}_2\frac{(n_i^A)_2(n_j^Y)_2}{(N)_2}+\overline{\overline{C}}_1\frac{n_i^An_j^Y}{N}\right),
\end{aligned}\tag{55}$$

其中$\overline{\overline{C}}_1=C_2+C_3+C_4$, $\overline{\overline{C}}_2=C_2+3C_3+7C_4$, $\overline{\overline{C}}_3=C_3+6C_4$, $\overline{\overline{C}}_4=C_4$, $\overline{C}_4=C_2$, $\overline{C}_3=C_2-C_1-2C_2n_i^A$, $\overline{C}_2=C_0+(C_1-C_2)n_i^A+C_2(n_i^A)^2$, $C_2=b\frac{(n_{j'}^Y)_2}{(N-n_j^Y)_2}$, $C_1=(2b(1-n_{j'}^Y)-a)\frac{n_{j'}^Y}{N-n_j^Y}$, $C_0=n_{j'}^Y(a-b)+b(n_{j'}^Y)^2$, $b=\frac{(n_{i'}^A)_2}{(N-n_i^A)_2}$, $a=\frac{n_{i'}^A}{N-n_i^A}$.

参考文献

1 Jordan M, Mitchell T. Machine learning: trends, perspectives, and prospects. Science, 2015, 349(6245):255-260.

2 Gates A J, Ahn Y Y. The impact of random models on clustering similarity. The Journal of Machine Learning Research, 2017, 18: 1-28.

3 Vinh N X, Epps J, Bailey J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance[J]. The Journal of Machine Learning Research, 2010, 11: 2837-2854.

4 Vinicius G C, Carlos E. Recent advances in decision trees: an updated survey. Artificial Intelligence Review, 2022, https://doi.org/10.1007/s10462-022-10275-5.

5 Quinlan J R. Discovering rules by induction from large collections of examples. Expert Systems in the Micro Electronic Age. Edinburgh University Press, 1979,168-201.

6 Quinlan J R. C4.5 Programs for machine learning, Morgan Kaufmann, San Mateo, CA, 1993.

7 Breiman L, Friedman J H, et al. Classification and regression trees. Chapman and Hall/CRC, Boca Raton, FL, 1986, 33: 128-128.

8 Hu Q, Che X, Zhang L, et al. Rank entropy-based decision trees for monotonic classification. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(11):2052-2064.

9 Demirovic E, Stuckey P J. Optimal decision trees for nonlinear metrics. In: Proceedings of the AAAI conference on artificial intelligence, 2021, 35(5), 3733 – 3741.

10 Aghaei S, Azizi M J, Vayanos P. Learning optimal and fair decision trees for non-discriminative decision-making. In: Proceedings of the AAAI conference on artificial intelligence, 2019, 33(01), 1418 – 1426.

11 Sok H K, Ooi M P L, Kuang Y C, et al. Multivariate alternating decision trees. Pattern Recognition, 2015, 50: 195 – 209.

12 Paul J B, Milo M L. Explainable neural networks that simulate reasoning. Nature Computational Science, 2021, 1: 607-618.

13 Zhou Z, Feng J. Deep Forest: Towards an Alternative to deep neural networks. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence Main track. 2017, 3553-3559.

14 Breiman L. Bagging predictors. Machine Learning, 1996, 24, 123-140.

15 Breiman L. Random forests. Machine Learning, 2001(1), 45, 5-32.

16 Rodriguez J J, Kuncheva L I, et al. Rotation forest: a new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(10), 1619 – 1630.

17 Rico B, Piotr F. Random rotation ensembles. Journal of Machine Learning Research, 2016(17), 4, 1-26.

18 Schapire R E, Freund Y, Bartlett P L, and Lee W S, Boosting the margin: A new explanation for the effectiveness of voting methods. Annals of Statistics, 1998, 26(5): 1651 – 1686.

19 Wang J, Qian Y, Li F, et al. Fusing fuzzy monotonic decision trees. IEEE Transactions on Fuzzy Systems. 2020, 28(5): 887-900.

20 Giorgi G M, Gigliarano C. The Gini concentration index: a review of the inference literature. Journal of Economic Surveys, 2017, 31(4): 1130-1148.

21 Reshef D N, Reshef Y A, Finucane H K, et al. Detecting novel associations in large data sets. Science, 2011, 334(6062): 1518-1524.

22 Serrurier M, Prade H. Entropy evaluation based on confidence intervals of frequency estimates: Application to the learning of decision trees. In: Proceedings of the International Conference on Machine Learning, 2015: 1576-1584.

23 李飞江, 钱宇华, 王婕婷, 梁吉业, 王文剑. 基于样本稳定性的聚类方法. 中国科学: 信息科学, 2020, 50(8), 1239-1254.

24 Bhargava T N, Uppuluri V R R. Sampling distribution of Gini's index of diversity. Applied Mathematics and Computation, 1977, 3(1): 1-24.

25 Roulston M S. Estimating the errors on measured entropy and mutual information. Physica D: Nonlinear Phenomena, 1999, 125(3-4): 285-294.

26 Ramos A M T, Casagrande H L, Macau E E N. Investigation on the high-order approximation of the entropy bias. Physica A: Statistical Mechanics and its Applications, 2020, 549: 124301.

27 Raileanu L E, Stoffel K. Theoretical comparison between the gini index and information gain criteria. Annals of Mathematics and Artificial Intelligence, 2004, 41(1): 77-93.

28 Shih Y S. Families of splitting criteria for classification trees. Statistics and Computing, 1999, 9(4): 309-315.

29 Breiman L. Some properties of splitting criteria. Machine learning, 1996, 24(1): 41-47.

30 Sandri M, Zuccolotto P. A bias correction algorithm for the Gini variable importance measure in classification trees. Journal of Computational and Graphical Statistics, 2008, 17(3): 611-628.

31 Wright M N, Ziegler A. ranger: A fast implementation of random forests for high dimensional data in C++ and R. Journal of Statistical Software, 2017, 77: 1-17.

32 Nembrini S, Nig I, Wright M N. The revival of the Gini importance?. Bioinformatics, 2018, 34(21): 3711-3718.

33 Romano S, Vinh N X, Bailey J, et al. A framework to adjust dependency measure estimates for chance. In: Proceedings of the 2016 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2016:

423-431.
34 Alin D, Johannes G. Bias correction in classification tree construction. In: Proceedings of the International conference on machine learning, 2001: 90-97.
35 Romano S, Bailey J, Nguyen V, et al. Standardized mutual information for clustering comparisons: one step further in adjustment for chance. International conference on machine learning, 2014: 1143-1151.
36 Altmann A, Tolosi L, Sander O, et al. Permutation importance: a corrected feature importance measure. Bioinformatics, 2010, 26(10): 1340-1347.
37 Kononenko I. On biases in estimating multi-valued attributes. In: Proceedings of the 14th international joint conference on Artificial intelligence-Volume, 1995: 1034-1040.
38 Wang J, Qian Y, Li F. Learning with mitigating random consistency from the accuracy measure[J]. Machine Learning, 2020, 109(12): 2247-2281.
39 Wang J, Qian Y, Li F, et al. Generalization performance of pure accuracy and its application in selective ensemble learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023,45(2): 1798-1816.
40 王婕婷, 钱宇华, 李飞江, 等. 消除随机一致性的支持向量机分类方法. 计算机研究与发展, 2020, 57(8): 1581-1593.
41 韩松来, 张辉, 周华平. 决策树算法中多值偏向问题的理论分析. 全国自动化新技术学术交流会会议论文集(一), 2005.
42 成红红, 钱宇华, 胡志国, 梁吉业, 基于邻域视角的关联关系挖掘方法. 中国科学: 信息科学, 2020, 50(6), 824-844.
43 Susanna L, Kyle H, Qiang Y. Batch normalization preconditioning for neural network training. Journal of Machine Learning Research, 2022, (72):1-41.
44 冯速(译). 组合数学. 机械工业出版社, 2012.
45 Kerber R. ChiMerge: discretization of numeric attribute. In: Proceedings of the 10th National Conference on Artificial Intelligence, 1992, 123-127.
46 Gao W, Wang L, et al. Risk minimization in the presence of label noise. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, 2016, 30(1).
47 Adam C, Honglak L, Andrew Y Ng. An analysis of single layer networks in unsupervised feature learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, 15: 215-223.
48 Chang J, Wang L, Meng G, et al. Deep adaptive image clustering. In: Proceedings of IEEE International Conference on Computer Vision. 2017, 5880-5888. 5879 – 5887.
49 Chen X, Xie S, He K. An empirical Study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 9640-9649.

Gini Index and decision tree method with mitigating random consistency

Jieting WANG[1], Feijiang LI[1], Jue LI[1], Yuhua QIAN[1,2*] & Jiye LIANG[2]

1. *Institute of Big Data Science and Industry, Shanxi University, Taiyuan* 030006*, China*;
2. *Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan* 030006*, China*
* Corresponding author. E-mail: jinchengqyh@sxu.edu.cn

Abstract Decision tree model has strong interpretability and is the basis of machine learning methods such as random forest and deep forest. How to select the segmentation attribute and segmentation value of nodes is the core problem of decision tree method, which has an impact on the generalization ability, depth, balance degree and other important performance of tree. Most of the traditional node selection attribute criteria are defined based on the sum of concave functions, which makes the decision tree algorithm have the problem of multi-value bias, that is, it tends to select the attribute with many values as the node segmentation attribute. In the classification task, the performance evaluation method from the perspective of random consistency is verified to have low classification bias. The evaluation criterion that alleviates random consistency can reduce classification bias and cluster number bias. In this paper, the random consistency of Gini index is alleviated based on the standard framework to alleviate its multi-value bias. It is verified by artificial data sets that standard Gini index can alleviate the multi-value bias problem of Gini index and select the attributes with decision information. Experimental results on twelve benchmark datasets and two image data sets show that the decision tree based on pure Gini index has higher generalization performance than the existing decision tree algorithms to mitigate multi-value bias.

Keywords Gini Index, Bias to Multi-value, Decision Tree, Random Consistency

Jieting WANG was born in 1991. She received the Ph.D degrees in computers with applications from Shanxi University, Taiyuan, China in 2021. She is currently a teacher at the Institute of Big Data Science and Industry, Shanxi University. Her research interest includes statistical machine learning and ensemble learning. She has published paper in the journal of IEEE Transactions on Pattern Analysis, Machine Intelligence, IEEE Transactions on Fuzzy Systems and Artificial Intelligence.

Feijiang LI was born in 1990. He received the Ph.D degree in computers with applications from Shanxi University, Taiyuan, China, in 2020. He is currently a teacher at the Institute of Big Data Science and Industry, Shanxi University. His research interest includes machine learning and knowledge discovery. He has published paper in the journal of Artificial Intelligence, ACM Transactions on Knowledge Discovery from Data, IEEE Transactions on Pattern Analysis and Machine Intelligence and IEEE Transactions on Neural Networks and Learning Systems.

Jue Li was born in 1995. She received the M.S degree in computer technology from Shanxi University, Taiyuan, China, in 2021. She is currently a phD at the Institute of Big Data Science and Industry, Shanxi University. Her research interest includes machine learning and data mining. She has published paper in the International Journal of Bio-Inspired Computation.

Yuhua QIAN was born in 1976. He received the M.S. and Ph.D. degrees in computers with applications from Shanxi University, Taiyuan, China, in 2005 and 2011, respectively. He is currently a Director at the Institute of Big Data Science and Industry, Shanxi University, where he is also a Professor with the Key Laboratory of Computational Intelligence and Chinese Information Processing, Ministry of Education. He is best known for artificial intelligence, machine learning and machine vision. He has authored over 100 articles on these topics in international journals He has published more than 120 papers in his research fields, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Fuzzy Systems and Artificial Intelligence.