控制与决策
Control and Decision

基于多目标特征提取的双层优化决策树分类算法

梁汎琴, 魏静萱, 梁斌豪

引用本文:
梁汎琴, 魏静萱, 梁斌豪. 基于多目标特征提取的双层优化决策树分类算法[J]. 控制与决策, 2026, 41(3): 718-727.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2025.0340

基于多目标特征提取的双层优化决策树分类算法

梁泓琴，魏静萱†，梁斌豪

(西安电子科技大学 计算机科学与技术学院，西安 710000)

摘　要: 高维不平衡数据广泛存在于社会生产的各个领域, 其特点是数据维度高以及数据类别的不平衡, 这种特性对传统分类算法的性能提出了极大的挑战. 不平衡的数据使得分类器偏向于多数类, 冗余特征导致分类性能的进一步下降. 对此, 首先针对冗余的高维特征提出基于多目标优化的特征提取算法, 考虑数据可分性和特征的泛化性能两个目标, 同时在目标内考虑数据的不平衡性; 其次, 提出基于双层优化的决策树分类算法, 将非叶子节点构建为双层优化的分类器, 上层搜索不同的特征组合, 下层求解该组合下的类别分界面; 最后, 在多个公开数据集上将所提出算法与其他算法进行对比实验验, 结果表明所提出算法在 *F*-score 和 *G*-mean 指标上明显优于其他对比算法, 验证了所提出算法的有效性.

关键词: 不平衡学习；特征提取；分类；高维数据；双层模型；决策树；多目标优化

中图分类号: TP391　　**文献标志码:** A

DOI: 10.13195/j.kzyjc.2025.0340

引用格式: 梁泓琴, 魏静萱, 梁斌豪. 基于多目标特征提取的双层优化决策树分类算法 [J]. 控制与决策, 2026, 41(3): 718-727.

Bilevel optimization decision tree based on multi-objective feature extraction

LIANG Feng-qin，*WEI Jing-xuan*†，*LIANG Bin-hao*

(School of Computer Science and Technology, Xidian University, Xi'an 710000, China)

Abstract: High-dimensional imbalanced datasets, characterized by elevated dimensionality and significant class imbalance, which presents substantial challenges to conventional classification algorithms. This imbalance induces classifier bias towards majority classes, while redundant features exacerbate performance degradation. To mitigate these issues, first, we propose a multi-objective feature extraction framework incorporating class separability and feature generalization performance as optimization objectives, explicitly accounting for class imbalance within the objective formulation. Then, we propose a bilevel optimization-based decision tree algorithm in which non-leaf nodes implement bilevel optimized classifiers. The upper level performs feature subspace exploration, while the lower level derives the optimal class-separating hyperplane for the identified subspace. Comparative evaluation on benchmark datasets demonstrates that the proposed methodology achieves statistically significant superiority over state-of-the-art approaches in both *F*-score and *G*-mean metrics, thereby validating its efficacy.

Keywords: unbalanced learning；feature extraction；classification；high dimensional data；bilevel model；decision tree；multi-objective optimization

0　引　言

随着大数据时代的到来, 高维数据分类成为机器学习领域的重要研究方向. 在生物信息学、医学影像分析、金融欺诈检测等实际场景中, 数据除了具备高维度性外还存在着类别间的不平衡性[1]. 高维数据通常包含成千上万的特征, 大量冗余特征使得特征间的关系不清晰, 导致分类性能下降. 此外, 一般的分类算法通常假设不同类别的样本数大致相等, 面对类别不平衡数据的分类任务时, 分类器学习到的特征偏向于多数类, 导致少数类的分类精度较低; 然而, 在这样的数据背景下, 用户通常对少数类样本更感兴趣. 如何在高维空间中识别并学习少数类的判别特征, 是高维不平衡数据分类算法的研究目标.

目前, 高维不平衡数据分类研究通常将数据的

收稿日期: 2025-04-02；**录用日期:** 2025-07-14.
基金项目: 国家自然科学基金项目 (62272367).
责任编委: 刘宝碇.
†通信作者. E-mail: wjx@xidian.edu.cn.

高维性和不平衡性看作两个部分, 分别提出针对性方法进行解决.

针对高维数据, 目前的研究方法是通过特征选择或特征提取对数据进行降维[2]. 特征选择从高维的特征集中选择部分特征表示整个特征集合. 由于高维特征间的复杂关系, 看似冗余的特征与其他特征组合后反而可能会有较好的分类性能, 但目前大多数方法直接从原始特征集中挑选特征子集, 没有考虑特征间的局部组合效果, 很难将这类型特征留在最终的特征子集中. 特征提取侧重挖掘特征间的关系, 将高维特征通过函数关系映射到低维空间, 能够更有效地压缩原始特征, 根据新特征和原特征是否线性相关分为线性和非线性. 为了得到更好的分类效果, 通常要使用非线性方法, 但得到的特征往往不易理解. 大多数的特征提取方法只考虑了单个目标, 难以精确捕捉到高维数据的类别特点.

不平衡性的处理方法分为数据级和算法级[3]. 数据级方法可以分为欠采样和过采样两种方式, 前者根据某种策略从多数类中删除部分样本, 后者则采用一些数据生成算法增加少数类样本的数量, 最终都能使得不同类别的样本量达到平衡. 但删除样本可能使得关键样本丢失, 导致类边界更加模糊, 并且目前的数据生成策略通常基于已有数据, 可能生成重复的数据, 导致分类器的过拟合. 算法级方法是代价敏感学习, 通过增加少数类样本的误分类代价或者减少多数类样本的误分类代价, 迫使分类算法对少数类样本的正确度有更高的关注度, 但是很难为样本误分类构建最优成本.

此外, 集成学习算法也是目前主流的研究方向, 通过结合多个基分类器通常可以得到更好的分类效果[4]. 面对高维不平衡数据的复杂性, 目前的集成学习算法通常将降维、不平衡数据的处理方法与分类器集成相结合. 因此, 对数据高维性和不平衡性的处理方法更为重要.

鉴于此, 本文提出一种基于多目标特征提取的双层优化决策树分类算法 (MOFE-BLODT), 用于解决高维不平衡数据的分类问题. 该算法首先使用多目标的特征提取模型对数据进行降维, 在此基础上使用基于双层优化的决策树模型对数据进行分类. 本文的主要工作如下:

1) 提出一种基于多目标优化的特征提取算法. 其中考虑两个目标: 数据在提取得到的特征中的可分性, 尽可能使得数据在新特征中是可分的; 新特征的泛化能力, 新特征一方面要增强少数类样本点的代表性, 另一方面要能代表整体的数据.

2) 提出一种基于双层优化的决策树算法. 决策树的每个非叶子节点都是一个双层优化的分类器, 上层搜索新特征集的不同组合, 下层求解该组合下的线性分类超平面.

3) 在多个高维不平衡数据集上, 将所提出算法与多个算法进行对比, 验证所提出算法的有效性.

1 相关工作

基于特征选择的方法为解决数据的不平衡性所带来的问题, 通常在特征选择中嵌入不平衡数据的处理方法. 例如, Zhang 等[5] 结合模糊聚类和粒子群优化, 提出了一种针对有缺失数据的高维不平衡数据的特征选择算法, 以填充风险的概念量化类别不平衡背景下缺失值对分类性能的影响, 引入模糊聚类对粒子群初始化, 提高了种群质量. Saadatmand 等[6] 将特征选择建模为多目标优化问题, 引入 Jaccard 相似性控制解的多样性, 设计了一种基于集合的变异操作提高种群的搜索, 采用一种双加权的 KNN 分类器评估个体, 用于解决传统 KNN 在不平衡数据中偏向多数类的问题. 虽然这些方法很大程度上减轻了特征的冗余程度, 能够得到更为精简的特征子集, 但是都没有考虑到特征的局部组合效果, 难以捕捉到数据中的潜在模式.

集成学习的研究重点在于得到多个差异化的分类器, 针对高维不平衡数据通常引入降维和重采样等方法构造分类器. 如文献 [7] 通过随机森林进行特征选择, 引入欠采样方法平衡类别分布. Wu 等[8] 在 AdaBoost 基础上引入折扣因子, 迭代过程中对不同的样本赋予不同的权重, 同时用基于 *k*-hub 聚类的欠采样方法生成平衡数据. 然而, 大多数的集成学习算法都是直接在原始空间中执行, 冗余特征在分类时严重影响了分类性能.

重采样和代价敏感学习分别从数据层面和算法层面缓解类别不平衡带来的问题. 文献 [9] 针对 CGAN 过采样受限于少数类样本规模的问题, 提出了 CGAN 与 SMTOEENN 相结合的过采样方法. 文献 [10] 提出了一种针对大规模图数据的方法, 基于贝叶斯决策规则的最优损失函数以及切割平面算法提出了成本敏感的子图选择方法. 但基于已知信息的过采样方法可能导致数据的过拟合, 欠采样方法可能将关键样本点剔除, 影响到分类效果. 对于代价敏感学习, 由于分类权重与数据高度相关, 很难找到最优的误分类权重, 稳定性和可扩展性较差.

2 基于多目标特征提取的双层优化决策树分类算法

本节将介绍所提出基于多目标特征提取的双层优化决策树分类算法, 算法框架如图 1 所示. 首先, 通过多目标优化的特征提取模型对原始的高维不平衡数据进行降维, 本文考虑了两个目标, 使得提取到的新特征不仅能够有效划分已有数据集, 还能提高对未知数据的泛化能力. 鉴于数据的类别不平衡, 本文在特征提取模型中引入对少数类样本的增强, 使得少数类样本在新特征中易于识别. 其次, 在新特征空间上训练一个基于双层优化的决策树分类器, 在每个非叶子节点上采用双层优化模型, 上层搜索新特征空间下不同的特征组合, 下层负责根据上层提供的特征组合求解线性分类超平面.

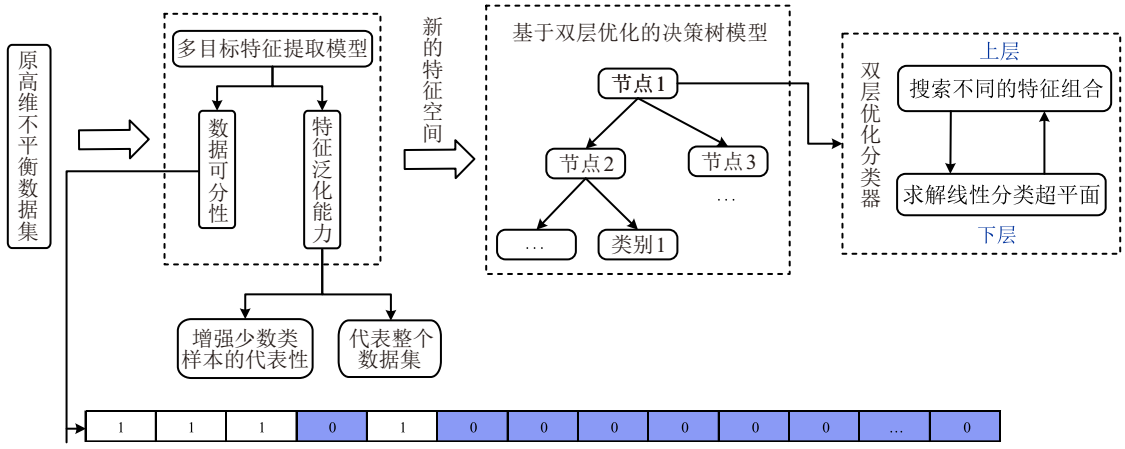


图1 算法架构

2.1 基于多目标优化的特征提取模型

2.1.1 多目标优化问题定义

若一个优化问题中涉及到两个或两个以上的目标, 则将其称为多目标优化问题. 一个具有n个决策变量和m个目标的最小化多目标优化问题可以表示为

$$\begin{aligned}&\min F(\boldsymbol{x})=(f_1(\boldsymbol{x}),f_2(\boldsymbol{x}),\dots,f_m(\boldsymbol{x}))^{\mathrm{T}};\\&\text{s.t. }\boldsymbol{x}\in X.\end{aligned}\tag{1}$$

其中: $\boldsymbol{x}=(x_1,x_2,\dots,x_n)^{\mathrm{T}}\in X$是$n$维决策向量, X是n维决策空间, $F(\boldsymbol{x})$是待求解的m个目标函数. 若满足下式:

$$\begin{cases}\forall i\in\{1,2,\dots,m\},\ f_i(\boldsymbol{x})\leqslant f_i(\boldsymbol{x});\\ \exists j\in\{1,2,\dots,m\},\ f_j(\boldsymbol{x})<f_j(\boldsymbol{x}).\end{cases}\tag{2}$$

则称解$\boldsymbol{x}$Pareto 支配解$\boldsymbol{y}(\boldsymbol{x}\prec\boldsymbol{y})$. 其中$f_i(\boldsymbol{x})$为$\boldsymbol{x}$在第$i$个目标上的真实目标值. 若不存在$\boldsymbol{x}$使得$\boldsymbol{x}^*\prec\boldsymbol{x}$, 则称$\boldsymbol{x}^*$为 Pareto 最优解, 多目标优化问题的目标即为找到一组 Pareto 最优解.

2.1.2 新特征表达式定义

给定一个包含D维特征和N个样例的数据集$X=(x_1,x_2,\dots,x_N)$, 其中$\{x_i\}_{i=1}^N\in R^D$. 特征提取寻找从D维特征集到d维特征集$(d\ll D)$的映射. 对数据集中的任意一个样本$x_i=(x_{i1},x_{i2},\dots,x_{iD})$, 特征提取转换后的样本$y_i=(\varphi_1(x_i),\varphi_2(x_i),\dots,\varphi_d(x_i))$. 因而, 特征提取就是寻找一组从$X$函数到$Y$函数的表达式$\varphi=(\varphi_1,\varphi_2,\dots,\varphi_d)$. 任意样本$x_i$的第$k$个新特征上的值通过函数映射$\varphi_k(x_i)$求得, 为了使得提取到的特征易于理解, 本文将其函数表达式定义为

$$\varphi_k(x_i)=x_{i1}^{b_{k1}}\times x_{i2}^{b_{k2}}\times\dots\times x_{iD}^{b_{kD}},\tag{3}$$

其中$b_{kj}\in E=\{-3,-2,-1,0,1,2,3\}$是特征提取任务的求解目标.

2.1.3 多目标模型建立

特征提取是分类任务的前置步骤, 目的是得到区分性较好的特征空间, 不仅将已有数据区分开, 同时挖掘出数据集类别间的不同, 对未知数据有良好的泛化性能. 对于类别不平衡数据, 还要放大少数类的特征, 使得少数类数据在新的特征空间中容易识别. 因此, 本文针对不平衡数据的特征提取任务设计了两个目标, 分别是现有数据的可分性和未知数据的泛化能力.

目标 1 数据可分性.

记X对应的类别标签为$l=(l_1,l_2,\dots,l_N)$, 其中$l_i\in\{0,1\}$. 本文着重研究二类别的高维不平衡数据分类任务, 少数类样本标签为 1, 多数类样本标签为 0. 考虑特征提取后的数据集$Y=(y_1,y_2,\dots,y_N)$, 其中$\{y_i\}_{i=1}^N\in R^d$, 数据可分性的目标是尽可能使得同类样本点聚集在一起. 具体而言, 对于任意一个新特征φ_k, 将原数据映射后的数据点按照从小到大的顺序排列后, 期望数据点排列形式为多数类与少数类各自聚集在数轴的两端, 尽可能将两个类别分开,

以便于后续分类任务的进行.

基于以上目的设计特征提取的第 1 个优化目标

$$\min f_1 = \sum_{i=1}^{m} p_{l_i} \frac{N_{l_i} - n_i}{N_{l_i}}.$$

$$p_{l_i} = \begin{cases} 1, & l_i = 1; \\ \dfrac{N_0}{N_1}, & l_i = 0. \end{cases} \tag{4}$$

其中: m为此特征上所有数据点按从小到大排列后, 同类别聚集的段数; N_1和N_0分别为数据集中少数类和多数类的样本数; l_i是此段上的数据点对应的类别标签; n_i是此段上的样本数; p_{l_i}为惩罚系数. 考虑到数据集的类别不平衡性, 由于多数类的样本较多, 可以被挖掘到的知识更多, 因此给予多数类较大的惩罚, 避免其混入少数类中. 假设一个数据集中少数类(类别标签为 1)共有 8 例, 多数类(类别标签为 0)共 14 例, 即$N_1 = 8$, $N_0 = 14$. 这些数据点在式(1)的某种变换得到的特征下, 数值按照从小到大排列后每个样本点对应的类别标签如图 2 所示, 可以看到在此数轴中数据按类别聚为 4 段, 各段对应的样本数为 6、4、2、10. 此数据集在这个特征中对应的目标函数计算式为

$$f_1 = \frac{8-6}{8} + \frac{14}{8} \times \frac{14-4}{14} + \frac{8-2}{8} + \frac{14}{8} \times \frac{14-10}{14}.$$

从目标函数可以看出, m越小, 特征中类别聚集的段数越少, 区分度越高; 第i段中n_i越大, 此段上聚集的该类别的样本越多, 区分度越高. 因此目标函数f_1越小, 代表此特征中的样本点越可区分.

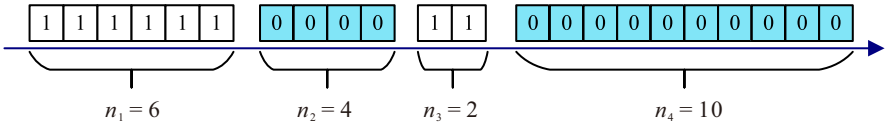


图2　目标 1 计算说明

目标 2　特征泛化能力.

目标 2 有两个作用, 一是使得新特征尽可能代表原始数据, 即最大程度保留原始数据的信息, 二是增强少数类的特征, 使得少数类容易识别.

本文基于不同类别的样本均值定义特征的泛化性能, 由于式 (1) 中新特征为原数据各个特征幂次方的乘积, 若直接采用新特征中的均值计算泛化性能, 新特征与原数据集的数据分布有较大差别, 不能代表类别特性. 为了解决以上问题, 对新特征φ_k上的数据点做如下变换:

$$z_i = \lg y_i = \lg \varphi_k(x_i) = \sum_{j=1}^{D} b_{kj} \lg x_{ij}. \tag{5}$$

图 3 为函数$y = x - 1$和$y = \lg x$在 0.5 ~ 1.5 区间上的函数图像, 可以看出两函数在$x = 1$附近的函数变化幅度基本一致, 为了减少特征分布的改变, 本文将原数据集的各个特征归一化到$[0.8, 1.2]$.

理论上, 收集到的数据是从该数据对应的实际

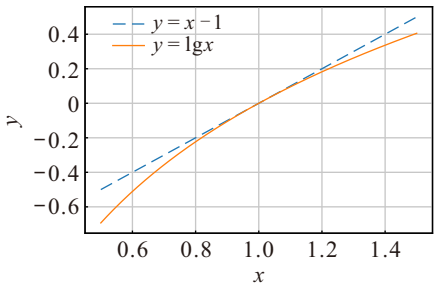


图3　函数对比图

问题中采样得到, 因而每个样本点在实际分布中都对应着一批类似的未知数据. 从这个角度出发, 若训练集中同类样本点尽可能的相似, 不同类样本点尽可能远离, 则在未知数据中应当也会拥有相同的效果. 因此定义目标 2 的第 1 部分为

$$f_{21} = \sum_{i=1, z_i \in Z_1}^{N_1} \frac{(z_i - \mu_1)(z_i - \mu_1)^{\mathrm{T}}}{(\mu_0 - \mu_1)^2} + \sum_{j=1, z_j \in Z_0}^{N_0} \frac{(z_j - \mu_0)(z_j - \mu_0)^{\mathrm{T}}}{(\mu_0 - \mu_1)^2}. \tag{6}$$

其中: Z_1, Z_0为数据集在新特征φ_k中取对数后的少数类和多数类的样本点集合; μ_1, μ_0为少数类和多数类的样本均值.

鉴于少数类样本数较少, 只考虑式 (4) 作为目标函数会导致新特征的过拟合. 由式 (5) 可以看出, z_i为原始特征对数值的线性组合, 为了减轻式 (4) 的过拟合, 应当尽可能减少降维过程中信息量的丢失, 即尽可能做到: 1) 降维前后样本点的总体变化较小, 采用最小投影距离衡量; 2) 降维后的样本点尽可能分散. 以图 4 为例, 原数据集有两个特征, 从分类效果上看, 以线性方式降维后$\boldsymbol{v}_1$与$\boldsymbol{v}_2$两个方向的分类效果相当, 但是以$\boldsymbol{v}_1$降维后的数据更为分散, 保有的信息量更多, 因而选择$\boldsymbol{v}_1$降维会有更好的泛化性. 此外, 还要考虑各类别的分散程度, 令少数类的分散度更大, 保存更多的信息量.

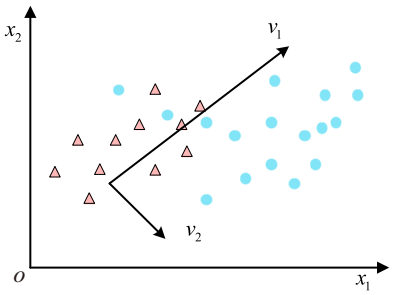


图4 降维方向对比

方差是用于刻画数据集分散程度的常用统计量, 一般而言, 方差越大数据中的信息量越大. 少数类和多数类的分散程度用类内方差表示, 符号表示为var_1和var_2. 本文对方差的公式做出调整用于刻画不平衡数据整体的分散程度, 有

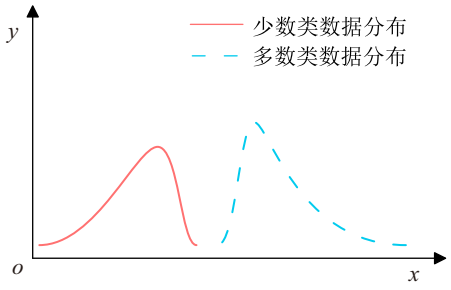
$$\mathrm{var}^* = \frac{1}{N}\sum (z_i - \bar{\mu})^2, \tag{7}$$

其中$\bar{\mu} = \dfrac{N_0}{N}\mu_1 + \dfrac{N_1}{N}\mu_0$.

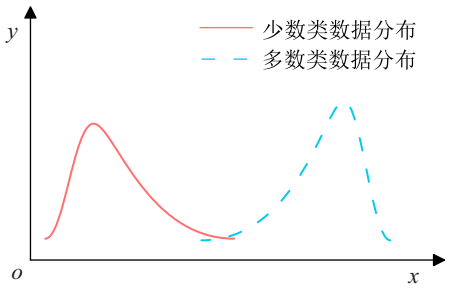
投影距离用投影前数据到投影直线的距离之和衡量, 即

$$\mathrm{dis} = \sum_{i=1}^{N} \frac{|z_i|}{\sqrt{\sum_{j=1}^{D} b_{kj}^2}}. \tag{8}$$

此外, 新特征中多数类和少数类的数据分布应当是偏态分布的. 若某特征中少数类和多数类分别分布在数轴两端, 则其两种不同的分布状态如图 5 所示. 图 5(a) 中少数类和多数类分别为左偏和右偏分布, 少数类样本的极端值分布在左侧, 多数类则分布在右侧, 大多数样本处于两分布的交界附近, 保证了交界线对未知数据的相对可靠性. 图 5(b) 中少数类和多数类分别为右偏和左偏分布, 此情况下交界附近的样本点较少, 未知数据中少数类混入多数类的可能性增大, 对多数类数据而言也有同样的结论. 因此, 定义变量用于刻画少数类和多数类的分布, 有



(a) 少数类左偏分布, 多数类右偏分布



(b) 少数类左偏分布, 多数类右偏分布

图5 数据分布说明

$$a_1 = \begin{cases} \dfrac{\mathrm{var}_{1\text{-be}}}{\mathrm{var}_{1\text{-af}}}, & \text{当少数类分布在数轴右端时;} \\ \dfrac{\mathrm{var}_{1\text{-af}}}{\mathrm{var}_{1\text{-be}}}, & \text{当少数类分布在数轴左端时;} \end{cases}$$

$$a_0 = \begin{cases} \dfrac{\mathrm{var}_{0\text{-be}}}{\mathrm{var}_{0\text{-af}}}, & \text{当多数类分布在数轴右端时;} \\ \dfrac{\mathrm{var}_{0\text{-af}}}{\mathrm{var}_{0\text{-be}}}, & \text{当多数类分布在数轴左端时.} \end{cases} \tag{9}$$

其中: $\mathrm{var}_{1\text{-be}}$, $\mathrm{var}_{1\text{-af}}$分别表示少数类在新特征的值按大小排序后, 前一半数据和后一半数据的方差, $\mathrm{var}_{0\text{-be}}$, $\mathrm{var}_{0\text{-af}}$类似. 由此定义目标 2 的第 2 部分为

$$f_{22} = \left(1 + \frac{N_1}{N}a_1 + \frac{N_0}{N}a_0\right) \times \left(\frac{\mathrm{dis}}{\mathrm{var}^*} + \frac{\mathrm{dis}}{\frac{N_1}{N}\mathrm{var}_1 + \frac{N_0}{N}\mathrm{var}_0}\right). \tag{10}$$

最后得到第 2 个优化目标为

$$\min f_2 = f_{21} + f_{22}. \tag{11}$$

2.1.4 多目标特征提取的优化算法

特征提取的目标是求解一组函数表达式$\varphi = (\varphi_1, \varphi_2, \ldots, \varphi_d)$, 由式 (1) 可知, 最终目标是求解$b_{kj}(k = 1, 2, \ldots, d, j = 1, 2, \ldots, D)$. 由于$b_{kj}$为整数, 每一维新特征的求解都是一个组合优化问题. 本文采用多目标进化算法进行求解, φ_k为个体, 其基因编码为$(b_{k1}, b_{k2}, \ldots, b_{kD})$, 目标函数为

$$\min F = \min(f_1, f_2). \tag{12}$$

算法 1 特征提取算法.

输入: 数据集X;
输出: 新特征集φ.
初始化种群P_1, 种群大小为s_1,
按照式 (1) 和 (9) 计算P_1的目标值.
for i =1: generation do
  对P_i执行快速非支配排序
  使用二元锦标赛方法选择s_1个个体作为父代
  执行交叉变异操作得到子种群C_i
  按照式 (1) 和 (9) 计算C_i的目标值
  $P_{i+1} = P_i + C_i$
  对P_{i+1}执行快速非支配排序
  $P_{i+1} = P_{i+1}[:s_1]$
end

算法 1 描述了该算法的执行过程.

1) 种群初始化.

种群的大小为s_1, 为了使得初始化时每个特征在群体中都会出现, 将D维特征随机均匀划分为s_1组, 每组代表一个个体, 假设第k组特征子集为$B_k=\{d_{k_1},d_{k_2},\ldots,d_{k_{D/s_1}}\}$, 则在对应个体的编码中令$b_{kd_l}=1(d_l\in B_k)$, $b_{kd_l}=0(d_l\notin B_k)$.

2) 种群排序规则.

选择采用二元锦标赛方法, 排序采用快速非支配排序算法[11], 同时针对排序算法中的多样性设计了一种新的度量方法. 任意两个体φ_i与φ_j的距离定义为使用到相同原特征的数目, 即

$$g(\phi_i,\phi_j)=\sum_{k=1}^{D}(\Gamma(b_{ik})\Gamma(b_{jk})). \tag{13}$$

其中

$$\Gamma(x)=\begin{cases}1, & x\neq 0;\\ 0, & x=0.\end{cases}$$

个体φ_i在种群中的多样性为$\max\limits_{j=1,2,\ldots,s_1} g(\varphi_i,\varphi_j)$, 排序后的同层个体中多样性的值越小排序越靠前.

3) 交叉变异操作.

任意两个父代个体φ_{P_1}与φ_{P_2}交叉产生后代φ_{C_1}和φ_{C_2}, 将两个父代个体逐元素进行交叉, 算法 2 描述了此执行过程. 个体φ_k中的元素逐个进行变异操作, 每个元素值都是从E中选取的. 个体中的元素b_{ki}以 5% 的概率发生变异, 为了提取到的特征尽可能的简单, 当发生变异时 75% 概率变为 0; 否则其值随机增加 1 或减少 1 得到变异后的值.

算法 2　特征提取的交叉操作.

输入: 父代个体φ_{P_1}、φ_{P_2};

输出: 子代个体φ_{C_1}、φ_{C_2}.

for i =1: D do:

　$r=\mathrm{rand}()$

　若$r\leqslant 0.5$, 则

　　$b_{C_1i}=b_{P_1i}$

　　$b_{C_2i}=b_{P_2i}$

　若$r>0.5$, 则

　　$b_{C_1i}=b_{P_2i}$

　　$b_{C_2i}=b_{P_1i}$

end

2.2　基于双层优化的决策树模型

2.2.1　双层优化问题

双层优化是指一种具有二层递阶结构的系统优化问题, 有上层和下层两个层次的优化目标. 一般而言双层优化问题具有以下形式:

$$\begin{aligned}&\min H_U(\boldsymbol{v}_u,\boldsymbol{v}_l);\\ &\text{s.t. } v_l\in\{\arg\min H_L(\boldsymbol{v}_u,\boldsymbol{v}_l)\}.\end{aligned} \tag{14}$$

其中: H_U和H_L分别为上层和下层的目标函数, $\boldsymbol{v}_l$和$\boldsymbol{v}_u$分别表示上层和下层的变量. 在一次优化过程中, 上层负责将变量$\boldsymbol{v}_u$传递给下层, 下层则负责求解此时的$\boldsymbol{v}_l$, 之后将求得的变量传递给上层, 由上层决定下一步的优化方向. 在此结构中, 上层的决策影响下层的行为, 下层的优化结果反过来影响上层的优化方向.

2.2.2　模型的建立与求解

大多数的分类器求解分界面时通常将所有的特征都考虑在内, 但求得的分界面不一定是最优的. 因此, 为了得到最优特征组合下的分界面, 本文提出基于双层优化的决策树模型. 从效果上看, 决策树中的每个非叶子节点都是一个分类器, 为了使得分类规则容易理解, 本文将叶子结点的分类规则定义为线性的, 若任意数据y_i满足下式:

$$\begin{aligned}&h(y_i,\boldsymbol{c},\boldsymbol{w},\theta)=\\ &\theta+c_1w_1y_{i1}+c_2w_2y_{i2}+\ldots+c_dw_dy_{id}<0,\end{aligned} \tag{15}$$

则被划分到左子节点, 否则被划分到右子节点. 其中$c_i\in\{0,1\}$标记对应特征φ_i在此次分类中是否起到作用. 为保证非叶子节点得到最优的分类超平面, 将优化目标设为双层, 上层用于搜索特征组合, 下层则求解该组合下的分界面. 此双层优化问题定义为

$$\begin{aligned}&\min H_U(\boldsymbol{c},\boldsymbol{w}^*,\theta^*);\\ &\text{s.t. } (\boldsymbol{w}^*,\theta^*)\in\arg\min H_L(\boldsymbol{w},\theta)|_c,\ c_i\in\{0,1\}.\end{aligned} \tag{16}$$

其中H_U、H_L分别是上层和下层的目标函数.

1) 下层目标函数H_L.

在双层模型中, 上层将某个特征组合c传递给下层, 下层的任务即为求解该组合下的分界面. 考虑到数据的不平衡性, 本文采用最小化误分类实例的比率[12] 作为目标函数, 其数学形式为

$$H_L(\boldsymbol{w},\theta)|_c=\sum_{i=1}^{N}h'(y_i,l_i,\boldsymbol{w},\theta).$$

$$h'(y_i,l_i,\boldsymbol{w},\theta)=\begin{cases}\dfrac{1}{N_0}, & h(y_i,\boldsymbol{c},\boldsymbol{w},\theta)<0\text{且}l_i=0;\\ \dfrac{1}{N_1}, & h(y_i,\boldsymbol{c},\boldsymbol{w},\theta)\geqslant 0\text{且}l_i=1;\\ 0, & \text{otherwise}.\end{cases} \tag{17}$$

函数h'根据错误分类的样例是少数类还是多数类赋予不同的代价. 在优化上采用与文献 [12] 相同的方式.

2)上层目标函数H_U.

如前所述, 上层的目标是找出最优分类效果的特征组合. 对于不同的特征组合, 首先考察该组合下的分类平面是否能将已有数据分开, 其次考虑其对未知数据是否拥有更好的分类效果. 对于两个不同的特征组$\boldsymbol{c}_1$、$\boldsymbol{c}_2$, 首先比较下式:

$$H_{U1}(\boldsymbol{c},\boldsymbol{w}^*,\theta^*)=\frac{N_1-N_{L1}}{N_1}+\frac{N_0-N_{R0}}{N_0}. \tag{18}$$

其中: N_{L1}表示被此分界面分到左子节点的少数类样本数, N_{R0}表示被分到右子节点的多数类样本数, H_{U1}越小则表示由$\boldsymbol{c}$特征组求得的分界面越能够将已有数据分开. 当$\boldsymbol{c}_1$和$\boldsymbol{c}_2$的H_{U1}相同时, 类似式 (4), 定义衡量特征组分类的泛化能力为

$$\begin{aligned}H_{U2}(\boldsymbol{c},\boldsymbol{w}^*,\theta^*)=&\\ \sum_{i=1,y_i\in Y_1}^{N_1}&\frac{(h(y_i,\boldsymbol{c},\boldsymbol{w},\theta)-\mu_1')^2}{(\mu_0'-\mu_1')^2}+\\ \sum_{i=1,y_i\in Y_0}^{N_0}&\frac{(h(y_i,\boldsymbol{c},\boldsymbol{w},\theta)-\mu_0')^2}{(\mu_0'-\mu_1')^2}.\end{aligned} \tag{19}$$

其中

$$\mu_1'=\frac{1}{N_1}\sum_{y_i\in Y_1}h(y_i,\boldsymbol{c},\boldsymbol{w},\theta),$$

$$\mu_0'=\frac{1}{N_0}\sum_{y_i\in Y_0}h(y_i,\boldsymbol{c},\boldsymbol{w},\theta).$$

非叶子节点的求解目标为此节点的分类规则, 即式 (15) 中的$(\boldsymbol{c},\boldsymbol{w},\theta)$, 求解过程采用进化算法, 个体基因编码为$\boldsymbol{c}=(c_1,c_2,\dots,c_d)$, 即个体的特征选取情况. 算法 3 描述了非叶子节点的求解过程.

算法 3 非叶子节点求解过程.

输入: 当前叶子节点的数据集;
输出: $\boldsymbol{c}$、$\boldsymbol{w}$、θ.
best = null
初始化种群P_1, 种群大小为s_2
评估P_1
for i =1:generation do:
  执行选择、交叉变异操作生成个体C_i
  评估C_i
  $P_{i+1}=P_i+C_i$
  对P_{i+1}执行排序操作
  $P_{i+1}=P_{i+1}[:s_2]$
  best = $P_{i+1}[0]$
  if 满足终止条件
    break
end
$[\boldsymbol{c},\boldsymbol{w},\theta]=[\text{best}.\boldsymbol{c},\text{best}.\boldsymbol{w},\text{best}.\theta]$

1) 种群初始化.

初始化时为了让每一个特征都能起作用, 种群中的个体只选取一个特征, 因此种群大小$s_2=d$, 个体$P[i]$的编码中令$c_i=1$, 其余置为零.

2) 种群排序.

选择操作采用二元锦标赛方法, 种群个体的评估过程如算法 4 所示, 评估的过程即为双层问题的求解过程, 个体的适应度由H_U决定, 但在计算之前, 首先求解下层目标函数, 计算出个体对应的$\boldsymbol{w}$和θ. 个体$P[i]$在满足下列情况时优于$P[j]$:

① $P[i].H_{U1}<P[j].H_{U1}$;

② $P[i].H_{U1}=P[j].H_{U1}$且$P[i].H_{U2}<P[j].H_{U2}$.

算法 4 种群评估.

输入: 种群P.
for i =1:Popsize do:
  根据$P[i].\boldsymbol{c}$执行下层求解得到$[P[i].\boldsymbol{w},P[i].\theta]$
  根据式 (15)、(16) 计算$P[i].H_{U1}$和$P[i].H_{U2}$
end

3) 交叉变异操作.

类似于算法 2, 两个父代个体编码中的元素逐对进行交叉, 随机赋给子代. 对个体逐元素进行变异操作, 个体编码中c_i以 5% 的概率变异为$1-c_i$.

3 实验分析

3.1 数 据 集

本文所提出算法针对高维不平衡数据的二分类任务, 实验所用的 20 个数据集的详细信息如表 1 所示, 包括数据集样本数 (N)、特征数 (D) 和不平衡比率 (IR). IR 是多数类和少数类样本数的比值, 用于衡量数据的不平衡程度

$$\text{IR}=\frac{N_0}{N_1}. \tag{20}$$

表 1 中“数据集名-i”表示将该数据集中类别标签为i的看作少数类, 其余看作多数类.

3.2 评价指标和参数设置

本文采用 F-score 和 G-mean 作为评价指标, 这也是不平衡数据分类的常用指标. 表 2 是二分类的混淆矩阵, 其中少数类标记为正类, 多数类标记为负类. 通过结合准确率和召回率, F-score 能更准确地评价不平衡数据的分类性能, 其定义如下:

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{21}$$

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{22}$$

表1　实验数据集

数据集	重命名	来源	N	D	IR
CLL_sub-0	D1	文献[13]	111	11340	9.09
Colon	D2	文献[13]	62	2000	1.18
DLBCL	D3	文献[14]	77	5469	3.05
Leukemia1-1	D4	文献[15]	72	5327	7.00
LSVT	D5	文献[14]	126	309	2.00
Lung-2	D6	文献[13]	203	3312	10.94
Lung-3	D7	文献[13]	203	3312	8.66
Lymphoma-2	D8	文献[16]	96	4026	8.60
Lymphoma-3	D9	文献[16]	96	4026	9.66
Lymphoma-4	D10	文献[16]	96	4026	7.72
Su_2001-3	D11	文献[17]	174	1517	5.69
Tomlins_v1-1	D12	文献[17]	104	2315	2.85
Tomlins_v1-5	D13	文献[17]	104	2315	7.66
Tomlins_v2-1	D14	文献[17]	92	1288	2.40
Tomlins_v2-3	D15	文献[17]	92	1288	1.87
Tox-1	D16	文献[13]	171	5748	2.80
Tox-2	D17	文献[13]	171	5748	2.80
Tox-3	D18	文献[13]	171	5748	3.38
Tox-4	D29	文献[13]	171	5748	3.07
warpAR10P-1	D20	文献[13]	130	2400	9

表2　二分类的混淆矩阵

	预测正类	预测负类
实际正类	TP	FN
实际负类	FP	TN

$$F\text{-score}=\frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}}. \tag{23}$$

G-mean 表示正类样本和负类样本的几何平均精度, 它是评估整体分类性能的有效指标, 其定义如下:

$$G\text{-mean}=\sqrt{\text{Recall}\times\text{Recall}^{-}}, \tag{24}$$

$$\text{Recall}^{-}=\frac{\text{TN}}{\text{TN}+\text{FP}}. \tag{25}$$

实验时特征提取求解算法的种群大小 s_1 设为 50, 进化代数为 300 代; 决策树非叶子节点求解算法的进化代数为 100 代, 当最优个体连续 10 代没有变化时提前退出进化. 交叉概率均设为 0.9.

3.3　对比算法

为了验证所提出算法的有效性, 在高维不平衡数据集上将 MOFE-BLODT 与多个主流算法进行比较, 包括集成算法 BalancedRF[18]、AsBagging[19]、RUSBoost[20]、CEMVO[7] 以及基于特征选择的算法 SVM-BEF[21]. 其中 BalancedRF 是随机森林算法的变体, AsBagging 在 Bagging 中加入不对称的重采样方法, RUSBoost 则是 RUS 与 Boost 算法的结合, SVM-BEF 根据特征对分类的贡献度进行特征选择, 损失函数上采用类别平衡的损失函数.

3.4　实验结果

为了减少随机性对实验结果的影响, 实验采用 5 折交叉验证并且重复 5 次评估算法的分类性能, 实

表3　不同算法的 *F*-score 实验结果

数据集	BalancedRF	AsBagging	RUSBoost	SVM-BEF	CEMVO	MOFE-BLODT
D1	0.8878 ± 0.0621	0.9462 ± 0.0373	0.9169 ± 0.0569	0.9533 ± 0.0581	0.9920 ± 0.0160	**1.0000 ± 0.0000**
D2	0.6681 ± 0.0780	0.6233 ± 0.0769	0.5564 ± 0.1137	0.7696 ± 0.0470	0.7467 ± 0.0227	**0.8084 ± 0.0313**
D3	0.6820 ± 0.0478	0.6665 ± 0.0497	0.7950 ± 0.0728	0.7579 ± 0.1135	0.8800 ± 0.0613	**0.9226 ± 0.0281**
D4	0.5597 ± 0.1280	0.6408 ± 0.0642	0.05525 ± 0.0608	0.7200 ± 0.1758	0.8266 ± 0.0326	**0.9253 ± 0.0159**
D5	0.7423 ± 0.0243	0.7211 ± 0.0266	0.6706 ± 0.0863	0.7972 ± 0.0214	0.8138 ± 0.0175	**0.8189 ± 0.0243**
D6	0.8464 ± 0.0447	0.8544 ± 0.0417	0.6430 ± 0.0410	0.8314 ± 0.0857	0.8587 ± 0.0241	**0.9182 ± 0.0230**
D7	0.6796 ± 0.0395	0.6892 ± 0.0364	0.7316 ± 0.0814	0.7837 ± 0.0508	0.8456 ± 0.0377	**0.8778 ± 0.0105**
D8	0.5676 ± 0.0624	0.6129 ± 0.0810	0.3903 ± 0.1057	0.7573 ± 0.1302	0.7466 ± 0.1066	**0.9600 ± 0.0326**
D9	0.4833 ± 0.0237	0.5395 ± 0.0400	0.3288 ± 0.0459	0.6933 ± 0.0679	0.6906 ± 0.1094	**0.8186 ± 0.0563**
D10	0.7734 ± 0.0439	0.7626 ± 0.0456	0.4685 ± 0.0504	0.8333 ± 0.0824	0.8653 ± 0.0190	**0.9706 ± 0.0258**
D11	0.7450 ± 0.0626	07655 ± 0.0495	0.8293 ± 0.0294	0.7220 ± 0.0795	0.8378 ± 0.0565	**0.9064 ± 0.0265**
D12	0.7142 ± 0.0604	0.7542 ± 0.0426	0.7254 ± 0.0688	0.8608 ± 0.0709	0.8949 ± 0.0293	**0.9240 ± 0.0336**
D13	0.8195 ± 0.0757	0.8672 ± 0.0926	0.5683 ± 0.0914	0.9440 ± 0.0441	0.9626 ± 0.0213	**1.0000 ± 0.0000**
D14	0.7304 ± 0.0475	0.8035 ± 0.0330	0.7090 ± 0.0546	0.8676 ± 0.0315	0.9090 ± 0.0219	**0.9112 ± 0.0212**
D15	0.6377 ± 0.0174	0.5998 ± 0.0354	0.6129 ± 0.0664	0.6066 ± 0.0676	0.7759 ± 0.0447	**0.7814 ± 0.0133**
D16	0.6040 ± 0.0388	0.5878 ± 0.0248	0.4582 ± 0.0774	0.5968 ± 0.0580	0.7118 ± 0.0745	**0.7759 ± 0.0281**
D17	0.6641 ± 0.0279	0.6702 ± 0.0225	0.4980 ± 0.0810	0.6773 ± 0.1027	0.8134 ± 0.0214	**0.8318 ± 0.0353**
D18	0.6620 ± 0.0203	0.7355 ± 0.0706	0.6279 ± 0.0548	0.7838 ± 0.0337	0.8772 ± 0.0402	**0.9057 ± 0.0369**
D19	0.7861 ± 0.0130	0.7855 ± 0.0217	0.7487 ± 0.0313	0.8363 ± 0.0291	0.8962 ± 0.0259	**0.9027 ± 0.0272**
D20	0.7983 ± 0.0094	0.8063 ± 0.0438	0.7571 ± 0.0174	0.9432 ± 0.0259	0.8648 ± 0.0568	**1.0000 ± 0.0000**

表4 不同算法的 *G*-mean 实验结果

数据集	BalancedRF	AsBagging	RUSBoost	SVM-BEF	CEMVO	MOFE-BLODT
D1	0.9663 ± 0.0325	0.9865 ± 0.0126	0.9717 ± 0.0389	0.9596 ± 0.0504	0.9926 ± 0.0146	**1.0000 ± 0.0000**
D2	0.7199 ± 0.0699	0.6890 ± 0.0699	0.6085 ± 0.1231	0.8125 ± 0.0345	0.7951 ± 0.0187	**0.8500 ± 0.0272**
D3	0.7876 ± 0.0492	0.7773 ± 0.0418	0.8844 ± 0.0475	0.7893 ± 0.0988	0.9008 ± 0.0597	**0.9471 ± 0.0182**
D4	0.7378 ± 0.1448	0.8142 ± 0.0540	0.7779 ± 0.0719	0.7345 ± 0.1825	0.8379 ± 0.0376	**0.9398 ± 0.0031**
D5	0.8046 ± 0.0188	0.7843 ± 0.0199	0.7396 ± 0.0710	0.8331 ± 0.0177	0.8575 ± 0.0151	**0.8618 ± 0.0243**
D6	0.9459 ± 0.0357	**0.9638 ± 0.0295**	0.9162 ± 0.0286	0.8641 ± 0.0830	0.8926 ± 0.0205	0.9452 ± 0.0238
D7	0.8908 ± 0.0334	0.9019 ± 0.0323	0.8518 ± 0.0615	0.8405 ± 0.0362	0.8770 ± 0.0276	**0.9338 ± 0.0182**
D8	0.7915 ± 0.0869	0.8251 ± 0.0738	0.6981 ± 0.1275	0.7888 ± 0.1205	0.7628 ± 0.1047	**0.9648 ± 0.0286**
D9	0.8062 ± 0.0460	**0.8881 ± 0.0288**	0.6981 ± 0.0693	0.7062 ± 0.0597	0.7156 ± 0.1149	0.8368 ± 0.0671
D10	0.9454 ± 0.0216	0.9444 ± 0.0114	0.8026 ± 0.0459	0.8644 ± 0.0714	0.8820 ± 0.0151	**0.9797 ± 0.0235**
D11	0.8846 ± 0.0393	0.9093 ± 0.0347	**0.9322 ± 0.0181**	0.7571 ± 0.0639	0.8572 ± 0.0452	0.9265 ± 0.0188
D12	0.8296 ± 0.0455	0.8510 ± 0.0283	0.7881 ± 0.0450	0.8841 ± 0.0634	0.9195 ± 0.0270	**0.9538 ± 0.0258**
D13	0.9017 ± 0.0682	0.9527 ± 0.0249	0.8070 ± 0.0654.	0.9501 ± 0.0385	0.9662 ± 0.0188	**1.0000 ± 0.0000**
D14	0.8122 ± 0.0324	0.8737 ± 0.0272	0.7753 ± 0.0403	0.8932 ± 0.0212	0.9268 ± 0.0194	**0.9364 ± 0.0167**
D15	0.7008 ± 0.0098	0.6706 ± 0.0252	0.6880 ± 0.0554	0.6730 ± 0.0540	0.8135 ± 0.0349	**0.8295 ± 0.0111**
D16	0.7379 ± 0.0351	0.7221 ± 0.0252	0.5789 ± 0.0639	0.6630 ± 0.0427	0.7829 ± 0.0588	**0.8336 ± 0.0148**
D17	0.7913 ± 0.0208	0.7976 ± 0.0169	0.6191 ± 0.0676	0.7402 ± 0.0822	0.8661 ± 0.0155	**0.8765 ± 0.0253**
D18	0.8171 ± 0.0219	0.8551 ± 0.0535	0.7072 ± 0.0534	0.8186 ± 0.0340	0.9028 ± 0.0313	**0.9341 ± 0.0268**
D19	0.8818 ± 0.0059	0.8787 ± 0.0173	0.8178 ± 0.0315	0.8700 ± 0.0231	0.9261 ± 0.0180	**0.9403 ± 0.0151**
D20	0.9676 ± 0.0022	0.9651 ± 0.0127	0.9378 ± 0.0086	0.9636 ± 0.0187	0.8946 ± 0.0709	**1.0000 ± 0.0000**

验结果以平均值 ± 标准差的形式展示, 每一行加粗的数据表示对应数据集的最优实验结果. 各算法在不同数据集上的 *F*-score 和 *G*-mean 的实验结果分别展示在表 3、表 4 中.

从表 3 可以看出, 所提出算法在所有数据集上的 *F*-score 指标均达到了最优, 如在数据集 D5、D9、D10 上分别达到了 0.9253、0.9600、0.8186 的 *F*-score, 分别高于第 2 好的结果 9.87%、20.27%、12.53%. 表明 MOFE-DLDT 算法提高了少数类样本的分类性能. 从表 4 可以看出, 所提出算法在 17 个数据集上的 *G*-mean 指标达到了最优, 如在数据集 D3、D8、D16 上的 *G*-mean 分别达到了 0.8500、0.9338、0.8295, 分别高于第 2 好的结果 3.75%、3.19%、1.6%. 实验结果验证了所提出算法的有效性.

4 结 论

本文针对高维不平衡数据分类问题, 提出了一种基于多目标特征提取的双层优化决策树分类算法. 为了减少冗余特征, 增强少数类样本的特点, 本文首先构建了一个基于多目标优化的特征提取模型, 为了使得已有样本点和未知数据在新特征空间中均能得到较好的区分性, 在此模型中考虑了数据的可分性和特征的泛化能力两个目标, 同时给出了目标函数和优化算法; 然后, 考虑到不同的特征组合对分类效果的影响, 提出了基于双层优化的决策树分类算法, 在非叶子节点上构造双层优化模型, 上层用于搜索不同的特征组合, 下层求解特征组合下的最优分界面; 最后, 在多个不同的高维不平衡数据集上进行对比实验, 实验结果表明本文所提出算法显著提高了分类性能, 验证了算法的有效性.

参考文献 (References)

[1] 李艳霞, 柴毅, 胡友强, 等. 不平衡数据分类方法综述[J]. 控制与决策, 2019, 34(4): 673-688.
(Li Y X, Chai Y, Hu Y Q, et al. Review of imbalanced data classification methods[J]. Control and Decision, 2019, 34(4): 673-688.)

[2] 张腾飞, 张宇迪, 马福民. 基于改进邻域空间的高维混合数据特征选择算法[J]. 控制与决策, 2024, 39(3): 929-938.
(Zhang T F, Zhang Y D, Ma F M. Improved neighborhood space based feature selection algorithm for highdimensional mixed data[J]. Control and Decision, 2024, 39(3): 929-938.)

[3] Han M, Li A, Gao Z H, et al. A survey of multi-class imbalanced data classification methods[J]. Journal of Intelligent & Fuzzy Systems, 2023, 44(2): 2471-2501.

[4] Ganaie M A, Hu M H, Malik A K, et al. Ensemble deep learning: A review[J]. Engineering Applications of Artificial Intelligence, 2022, 115: 105151.

[5] Zhang Y, Wang Y H, Gong D W, et al. Clustering-guided particle swarm feature selection algorithm for high-dimensional imbalanced data with missing values[J]. IEEE Transactions on Evolutionary Computation, 2022, 26(4): 616-630.

[6] Saadatmand H, Akbarzadeh-T M R. Many-objective jaccard-based evolutionary feature selection for high-dimensional imbalanced data classification[J]. IEEE Trans Pattern Anal Mach Intell, 2024, 46(12): 8820-8835.

[7] Xu Y H, Yu Z W, Philip Chen C L. Classifier ensemble based on multiview optimization for high-dimensional imbalanced data classification[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(1): 870-883.

[8] Wu Q, Lin Y P, Zhu T F, et al. HUSBoost: A hubness-aware boosting for high-dimensional imbalanced data classification[C]. 2019 International Conference on Machine Learning and Data Engineering (iCMLDE). Taipei, 2019: 36-41.

[9] 刘宁, 朱波, 阴艳超, 等. 一种混合 CGAN 与 SMOTEENN 的不平衡数据处理方法[J]. 控制与决策, 2023, 38(9): 2614-2621.
(Liu N, Zhu B, Yin Y C, et al. An imbalanced data processing method based on hybrid CGAN and SMOTEENN[J]. Control and Decision, 2023, 38(9): 2614-2621.)

[10] Pan S R, Wu J, Zhu X Q. CogBoost: Boosting for fast cost-sensitive graph classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(11): 2933-2946.

[11] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[12] Bonyadi M R, Reutens D C. Optimal-margin evolutionary classifier[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(5): 885-898.

[13] Li J D, Cheng K W, Wang S H, et al. Feature selection[J]. ACM Computing Surveys, 2018, 50(6): 1-45.

[14] Lichman M. UCI machine learning repository[DB/OL]. (2025-03-02). http://archive.ics.uci.edu/ml.

[15] Chen K, Xue B, Zhang M, et al. An evolutionary multitasking-based feature selection method for high-dimensional classification[J]. IEEE Trans Cybern, 2022, 52(7): 7172-7186.

[16] Zhu Z X, Ong Y S, Dash M. Markov blanket-embedded genetic algorithm for gene selection[J]. Pattern Recognition, 2007, 40(11): 3236-3248.

[17] de Souto M C, Costa I G, de Araujo D S, et al. Clustering cancer gene expression data: A comparative study[J]. BMC Bioinformatics, 2008, 9: 497.

[18] Chen C, Liaw A, Breiman L, et al. Using random forest to learn imbalanced data[R]. Berkeley: University of California, 2004.

[19] Tao D, Tang X, Li X, et al. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval[J]. IEEE Trans Pattern Anal Mach Intell, 2006, 28(7): 1088-1099.

[20] Seiffert C, Khoshgoftaar T M, van Hulse J, et al. RUSBoost: A hybrid approach to alleviating class imbalance[J]. IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans, 2010, 40(1): 185-197.

[21] Maldonado S, Weber R, Famili F. Feature selection for high-dimensional class-imbalanced data sets using support vector machines[J]. Information Sciences, 2014, 286: 228-246.

作者简介

梁泯琴 (2001−), 女, 硕士生, 主要研究方向为高维数据分类, E-mail: liangfq241@163.com;

魏静萱 (1981−), 女, 副教授, 博士, 主要研究方向为智能计算、多目标优化, E-mail: wjx@xidian.edu.cn;

梁斌豪 (1999−), 男, 硕士生, 主要研究方向为大规模多目标优化, E-mail: lbh260763@163.com.