计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(11):281−288 [doi: 10.15888/j.cnki.csa.008124]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 MCP 惩罚的群组变量选择和 AdaBoost 集成剪枝①

万红燕, 张云云

(中国科学技术大学 管理学院, 合肥 230026)
通讯作者: 张云云, E-mail: sa017013@mail.ustc.edu.cn

摘　要: 针对高维群组变量下的分类问题, 本文提出了一种基于 MCP 惩罚的 AdaBoost 集成剪枝逻辑回归模型 (AdaMCPLR), 将 MCP 函数同时应用于特征选择和集成剪枝, 在简化模型的同时有效地提升了预测精度. 由于传统的坐标下降算法效率较低, 本文引用并改进了 PICASSO 算法使其能够应用于群组变量选择, 大大提高了模型的求解效率. 通过模拟实验, 发现 AdaMCPLR 方法的变量选择和分类预测效果均优于其他预测方法. 最后, 本文将提出的 AdaMCPLR 方法应用于我国上市公司财务困境预测中.

关键词: AdaBoost; 集成剪枝; 双层变量选择; 财务困境预测

引用格式: 万红燕,张云云.基于 MCP 惩罚的群组变量选择和 AdaBoost 集成剪枝.计算机系统应用,2021,30(11):281–288. http://www.c-s-a.org.cn/1003-3254/8124.html

MCP-Based Method in Group Variable Selection and AdaBoost Ensemble-Pruning

WAN Hong-Yan, ZHANG Yun-Yun

(School of Management, University of Science and Technology of China, Hefei 230026, China)

Abstract: To tackle the classification problem of high-dimensional group variables, this study proposes an MCP-based AdaBoost ensemble-pruning logistic regression model (AdaMCPLR). The MCP function is applied to feature selection and ensemble pruning simultaneously, which not only simplifies the model, but also effectively improves the prediction accuracy. For the efficiency enhancement, this paper improves the PICASSO algorithm to make it applicable to group variable selection. Simulation experiments show that the AdaMCPLR method is superior to other prediction methods in variable selection and classification prediction. Finally, the AdaMCPLR method proposed in this study is applied to the financial distress prediction of listed companies in China.

Key words: AdaBoost; ensemble-pruning; bi-level selection; financial distress prediction

在统计和数据分析领域研究中, 一个重要主题就是在响应变量和预测变量之间找到具有更好解释性和预测性的模型. 但是, 随着大数据的到来, 我们所处理的数据和研究的问题越来越复杂, 这种复杂性往往体现在高维性. 例如: 在影像遗传学研究中, 预测变量为基因 SNP 数据, 响应变量为扫描影像数据, 两者都高达几十万维, 若将全部变量代入模型分析, 模型的解释性和预测性会大大降低, 且没有什么实际意义. 所以, 简化模型对指导研究尤其重要, 较为常用的想法是进行特征选择.

特征选择通过从众多变量中选出重要变量从而获得稀疏的模型, 这些方法多是采用在损失函数后面添加惩罚项的方式来达到变量选择的作用. 针对于单变量选择的惩罚函数有 LASSO, bridge[1], 平滑削边绝对

① 收稿时间: 2021-01-11; 修改时间: 2021-02-07; 采用时间: 2021-02-23; csa 在线出版时间: 2021-10-22

偏离惩罚 (SCAD)[2], 极小极大凹惩罚 (MCP)[3] 等. 在处理实际问题时, 经常会遇到变量分组的情况, 比如本文实证部分数据来源于 CSMAR 数据库, 其就将企业财务数据根据盈利能力、成长能力、偿债能力、分红能力等将变量分成几组, 每组均包含若干个变量. 在这种情况下使用单变量选择方法, 就会忽略分组信息, 从而可能导致变量选择效果大大降低, 因此有些学者开始研究群组变量选择的方法. Yuan 等[4] 提出了 group LASSO, 其惩罚函数由群组水平上的L_2范数组成, 从而达到组级别上的稀疏性; Meier 等[5]将这一想法扩展到逻辑回归; 随后受 group LASSO 的启发, Wang 等[6] 提出了 Group SCAD, Huang 等[7] 提出了 Group MCP.

为了实现在选择组变量的同时又可以对组内的变量进行选择, 学者们在已有的方法上进行改进, 提出了双层变量选择方法, Huang 等[3] 对组结构使用L_1凹惩罚提出了 Group Bridge 方法[8]; Breheny 等[9]2009 年提出了 CMCP (Composite MCP), 其惩罚项是组间惩罚和组内惩罚的复合函数[8]; Simon 等[10] 提出了 SGL (Sparse Group LASSO), 这里的惩罚项是组内和组间惩罚函数组成的可加惩罚函数; 在此基础上, 王小燕等[11] 提出了 adSGL (adaptive SGL), 其通过附权的方式对不同的组系数和单变量系数进行不同程度惩罚, 从而避免了对大系数的过度惩罚. 双层变量选择方法既考虑了分组信息, 也做到了对组间变量和组内变量的同时选择, 灵活性高, 同时变量选择的效果也更好.

针对群组变量下的二分类问题, 已有研究均集中在 logistic 模型, 即在 logistic 模型的负对数似然损失函数后面添加惩罚项, 在变量选择的同时直接对二分类问题作出预测, 其预测精度有进一步提升空间. AdaBoost 是一种高效的集成学习方法, 其依次生成一系列基学习器, 并结合它们的预测来生成最终结果, 能够显著提升预测效果, 然而, 由于 AdaBoost 在处理高维数据时使用大量的基学习器来产生最终结果, 因此高内存空间消耗成为一个关键的挑战[12]. 已有学者将特征选择方法应用于集成剪枝, 通过集成剪枝, 能产生一个规模更小但精度更高的模型. Margineantu 等[13] 提出的 Kappa 剪枝是一种基于顺序的剪枝方法, 通过某种标准对学习器进行排序, 并选择最优的学习器组合组成最终的学习器; Lazarevic 等[14]提出一种基于族的集成剪枝方法, 即先对各学习器进行 K-means 聚类, 保留每类中表现较优的基学习器组成最终的分类器, 从而提高模型的泛化能力; Azmi 等[15] 提出了一种利用 LASSO 的类关联规则进行集成剪枝的新方法, 该方法对不太重要的基学习器进行剪枝; Jiang 等[16] 针对高维单变量选择提出了两阶段集成剪枝方法, 显著地提升了预测性能.

本课题提出并研究了一种基于 MCP 惩罚的 AdaBoost 集成剪枝逻辑回归模型 (AdaMCPLR), 以产生对高维群组变量下的分类问题的卓越预测精度. 该方法可分为 CMCP 双层变量选择、集成和集成剪枝 3 个步骤. 第 1 步, 使用 CMCP 惩罚 logistic 回归来降低数据的维数; 第 2 步, 经过特征选择后的数据集生成 AdaBoost 集成; 最后, 再次使用 MCP 函数对集成系统进行剪枝, 并得到一个简单但更有效的模型. 在此步骤中, 我们还使用 LASSO 和 SCAD 对集成进行了修剪, 并通过模拟对比了 3 种惩罚的分类和选择性能, 显示了基于 MCP 的剪枝方法的优越性.

1 AdaMCPLR 方法

AdaMCPLR 可分为 CMCP 双层变量选择、集成和集成剪枝 3 个步骤. 首先, 在 logistic 模型的负对数似然损失函数后添加 CMCP 函数来降低数据的维数; 接着, 利用特征选择后的数据集生成 AdaBoost 集成; 最后, 我们再次使用 MCP 函数对集成系统进行了剪枝, 得到了一个简单但更有效的模型. 以下分别介绍各步骤中涉及的方法.

1.1 CMCP 双层变量选择

当处理二分类任务时, logistic 回归往往是最常用的模型. 考虑一个标准化的$n\times d$设计矩阵$X=(x_1, x_2,\cdots,x_n)^{\mathrm{T}}$ (n为观测数, d为特征数), 已知X的d个特征共分为J组, 且第j组变量包含K_j个变量 ($j=1,2,\cdots,J$), p维向量$y=(y_1,y_2,\cdots,y_n)^{\mathrm{T}}$为响应变量. 那么, logistic 回归模型的形式如下:

$$y_i=\frac{1}{1+\exp(-x_i\beta)},i=1,\cdots,n \tag{1}$$

其中, $\beta=(\beta_1,\cdots,\beta_d)$是模型的系数, logistic 模型等价于最小化:

$$L(\beta)=\sum_{i=1}^{n}\{-y_i x_i\beta+\ln[1+\exp(x_i\beta)]\} \tag{2}$$

其中, $L(\beta)$ 表示 logistic 模型的负对数似然损失函数. 通过在损失函数后面添加惩罚项来得到稀疏的模型,

是一种常用的正则化方法. 本文采用 CMCP (Composite MCP) 作为模型的惩罚项, CMCP 是一种复合惩罚项, 其是由组内惩罚和组间惩罚组成的复合函数, 复合惩罚的具体形式为:

$$\sum_{j=1}^{J} f_{\lambda,b}\left(\sum_{k=1}^{K_j} f_{\lambda,a}(|\beta_{jk}|)\right) \tag{3}$$

其中, a、b 分别是内层惩罚函数和外层惩罚函数的调整参数, CMCP 中的内层惩罚和外层惩罚都是极小极大凹惩罚 (MCP) 函数, 其具体形式及其导数形式分别为:

$$f_{\lambda,\mu}(\beta_j)=\begin{cases}\lambda\beta_j-\dfrac{\beta_j^2}{2\mu}, 0\le\beta_j\le\mu\lambda, \mu\ge 0\\ \dfrac{\mu\lambda^2}{2}, \beta_j\ge\mu\lambda, \mu\ge 0\end{cases} \tag{4}$$

$$f'_{\lambda,u}(\beta_j)=\begin{cases}\lambda-\dfrac{\beta_j}{\mu}, 0\le\beta_j\le\mu\lambda, \mu\ge 0\\ 0, \beta_j\ge\mu\lambda, \mu\ge 0\end{cases} \tag{5}$$

由式 (5) 可以看出, MCP 一开始采用了与 LASSO 相同的处罚率, 然而, 它却不断地放松惩罚, 直到$\beta_j\ge\mu\lambda$, 惩罚率降为零. 添加惩罚项之后, logistic 回归的优化问题变为:

$$\min_{\beta}\ \sum_{i=1}^{n}\{-y_i x_i\beta+\ln[1+\exp(x_i\beta)]\}+f_{\lambda,b}\left(\sum_{k=1}^{K_j} f_{\lambda,a}(|\beta_{jk}|)\right) \tag{6}$$

通过最小化目标函数, 求解得到模型系数$\hat{\beta}$, 由于 MCP 函数具有特征选择的作用 (某些变量系数会被压缩至 0), 所以当组内惩罚和组间惩罚均为 MCP 时, 能同时对组间变量和组内变量进行选择, 达到双层变量选择的作用, 经过变量选择之后的设计矩阵记为X_{sub}.

1.2 AdaBoost 算法

Boosting 是一种高效的集成学习方法, 它将众多基学习器结合在一起, 组成最终的强学习器, AdaBoost 就是一个典型的 Boosting 算法. 本文利用经过特征选择后的数据集X_{sub}来构造集成模型, 进一步提升预测精度. AdaBoost 模型主要由两个主要部分组成: 加法模型和前向分步算法. 加法模型可以表示如下:

$$H(X)=\sum_{t=1}^{T}\alpha_t h_t(x) \tag{7}$$

其中, $h_t(x)$代表第 t 步迭代中的弱分类器, 此处采用的基学习器是决策树; α_t是第 t 个弱分类器在最终的强分类器上所占权重, $H(X)$代表弱分类器的线性组合, 也即最终的强分类器. 在前向分步算法中, 每一步迭代生成的分类器都是在前一次迭代的分类器的基础上产生的, 可以表示为:

$$H_m(x)=H_{m-1}(x)+\alpha_t h_t(x) \tag{8}$$

其中, $H_{m-1}(x)$是前 $m-1$ 步迭代中所有弱分类器的线性组合. 算法开始每个样本的权重相同, 在这样的样本分布下训练得到第一个弱分类器以及该分类器在训练集上的误差, 误差决定此分类器的权重. AdaBoost 算法使用的损失函数是指数损失函数, 每一轮的弱分类器的权重计算公式如下:

$$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\epsilon_{\text{t}}}{\epsilon_{\text{t}}}\right) \tag{9}$$

由式 (9) 可知弱分类器错误率越低, 它在最终的分类器所占的权重就越大, 同时每一步迭代的训练样本的分布也会根据α_t调整, t 轮迭代中第 i 个样本的权重计算如下:

$$\begin{aligned}W_{t+1,i}&=\frac{W_{t,i}}{Z_t}\exp(-\alpha_t y_i h_{t,i}(x))\\&=\begin{cases}\dfrac{W_{t,i}}{Z_t}\mathrm{e}^{-\alpha_t}, & h_{t,i}y_i(x)=y_i\\ \dfrac{W_{t,i}}{Z_t}\mathrm{e}^{\alpha_t}, & h_{t,i}y_i(x)\ne y_i\end{cases}\end{aligned} \tag{10}$$

其中, y_i表示第i个样本的分类标签, $h_{t,i}(x)$表示第i个样本在第t个弱分类器的预测结果, 由式 (10) 可知当该样本被误分类时, 样本权重将会增大, 而被正确分类时样本权重将会减少. 假设训练集共有 N 个样本, $Z_t=W_{t,i}\Big/\sum_{i=1}^{N}W_{t,i}$是规范化的因素, 用于确保每轮迭代所有的样本权重总和为 1. 最终的强分类器表示为$f(x)=\text{sign}(H(X))$, 即最终的分类结果取决于最终组成的强分类器分类结果的符号.

1.3 集成剪枝

AdaBoost 通过迭代生成一组基学习器来组成最终的分类器, 可以显著提高分类的准确率. 但是, 它也占用了相当多的内存, 特别是在处理具有许多特征的高维数据时; 此外, 由于错误分类的样本权重较大, 基学习器倾向于适应这些情况, 当数据中存在噪声时, 会导致过拟合. 因此, 必须对集成进行剪枝, 以减少存储空间的需求, 同时提高分类器性能. 受 Jiang 等[16] 的启发, 本文将集成剪枝扩展应用于群组变量下的分类预测问题.

AdaMCPLR 在集成剪枝步骤中的构造方法类似于

stacking 方法, 在 stacking 方法中, 称单个基学习器为初级学习器, 将各基学习器结合起来的学习器叫做次级学习器或元学习器. 本文中的 AdaMCPLR 与 stacking 方法区别在于 stacking 的初级学习器是多种模型组成, 而 AdaMCPLR 的初级学习器均是决策树, 其次级学习器是 logistic 回归模型, 也即 AdaMCPLR 以 AdaBoost 迭代生成的一系列基学习器的预测结果作为 logistic 回归的输入, 并在 logistic 回归的负对数似然函数后面加上 MCP 惩罚, 从而达到集成剪枝的目的.

如上所述, 为了简化模型和提高模型性能, 必须进行集成剪枝. 本文提出的 AdaMCPLR 将惩罚 logistic 回归的概念用于 Boosting 集成剪枝, 考虑以下 $n\times T$ 矩阵:

$$Z=\begin{pmatrix} h_1(x_1) & \cdots & h_T(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_n) & \cdots & h_T(x_n) \end{pmatrix} \tag{11}$$

其中, $i(i=1,2,\cdots,n)$ 行表示 T 个基学习器在给定样本 x_i 的情况下所做的预测,也即每个基学习器都可以被看作 logistic 回归模型中的一个特征, 以 Z 作为 logistic 回归模型的设计矩阵, 从而将集成剪枝问题转化为一个优化问题, 通过最小化式 (12) 来寻求稀疏权向量 $\hat{\theta}$:

$$\sum_{i=1}^{n}\{-y_i z_i\theta+\ln[1+\exp(z_i\theta)]\}+f_{\lambda_1,\gamma_1}(\theta_j) \tag{12}$$

其中, z_i 表示 Z 的第 i 行, 这里的惩罚项 $f_{\lambda_1,\gamma_1}(\alpha_j)$ 是 MCP 函数. 将 MCP 惩罚应用于集成剪枝中, 一些基学习器的相关系数会被压缩至 0, 从而达到集成剪枝的目的.

2 算法

Zhao 等[17] 提出了 PICASSO 算法 (PathwIse CalibrAted Sparse Shooting algOrithm), 其分为 3 个循环: 外循环、中间循环和内循环, 由于在每次迭代中, 中间循环从所有变量中只选定一小部分变量的索引组成活跃集, 并将活跃集传递到内循环中, 利用坐标下降算法求解出对应的解, 从而大大提高了求解效率. 本文对 PICASSO算法进行改进, 使其能够应用于群组变量选择.

考虑一个标准化的 $n\times d$ 设计矩阵 $X=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ (n 为观测数, d 为特征数) 和二元响应向量 $y\in\{-1,1\}^n$, 已知 X 的 d 个特征共分为 J 组, 且第 j 组变量包含 K_j 个变量 ($j=1,2,\cdots,J$). 在生成 Boosting 集合之前, 我们筛选变量并通过最小化 $F_\lambda(\beta)$ 来降低维数:

$$F_\lambda(\beta)=\sum_{i=1}^{n}\{-y_i x_i\beta+\ln[1+\exp(x_i\beta)]\}+\sum_{j=1}^{J}f_{\lambda,b}\left(\sum_{k=1}^{K_j}f_{\lambda,a}(|\beta_{jk}|)\right) \tag{13}$$

其中, 式 (13) 的第一项表示 logistic 回归的负对数似然损失函数 $L(\beta)$, 后一项中的 $f_{\lambda,a}$、$f_{\lambda,b}$ 分别表示组内和组间的惩罚函数, 此处都是 MCP 函数, a 和 b 分别表示组内和组间惩罚的权重, 一般取 $b=K_j a\lambda/2$.

在外循环中, 生成正则化参数 λ 的一个递减序列. λ 的初始值记为 $\lambda^{(0)}$, 初始状态下 $\beta^{(0)}=(0,0,\cdots,0)^{\mathrm{T}}$, 随着 λ 的逐渐减小, 解向量出现更多的非零元素. λ 递减的速率由参数 $\eta\in(0,1)$ 控制.

在中间循环中, 会从 X 中选出活跃变量, 并将活跃变量的索引组成活跃集 A, 初始的活跃集 $A^{(0)}$ 定义为:

$$A^{(0)}=\{j|\beta_j^{(0)}=0 \text{ and } \nabla_j L(\beta^{(0)})\geq(1-\delta)\lambda\}\cup\{j|\beta_j^{(0)}\neq 0\} \tag{14}$$

其中, δ 是一个很小的值, 此处取 0.1, $\nabla_j L(\beta^{(0)})$ 是梯度向量 $\nabla L(\beta^{(0)})$ 的第 j 个元素. 相应的非活跃集定义为:

$$\bar{A}^{(0)}=\{j|\beta_j^{(0)}=0 \text{ and } \nabla_j L(\beta^{(0)})<(1-\delta)\lambda\} \tag{15}$$

只有 A 内的元素会被传递到内循环中, 从而求解出对应的解. 同时中间循环的优化目标变为:

$$\hat{\beta}=\arg\min F_\lambda(\beta),\ \text{s.t.}\ \beta_{\bar{A}}=0 \tag{16}$$

在迭代过程中, 一旦活跃集中某元素对应的系数变为 0, 如 β_j, 那么 j 将从活跃集中移除. 同时如果非活跃集中的元素 k 满足式 (17), 将会被加入到活跃集中.

$$k=\arg\max_{k\in\bar{A}}|\nabla_j L(\beta)| \tag{17}$$

直到 $|\nabla_j L(\beta)|<(1-\delta)\lambda$, 中间循环的迭代停止. 此处的 λ 是由外循环给定的.

在内循环中, 利用坐标下降算法对活跃集中元素对应的变量系数进行求解, 在 $t+1$ 步迭代中 ($t=0,1,2,\cdots,T$):

$$\beta_j^{t+1}=\arg\min_{\beta_j}F_\lambda(\beta_j,\beta_{\backslash j}^{(t)}) \tag{18}$$

其中, $\beta_j^{(t)}$ 表示 β 的第 j 个分量在第 t 步迭代中估计值, $\beta_{\backslash j}$ 是 β 删除了第 j 个分量后的子向量. 在内循环中, β^{t+1} 每次只更新一个元素而保持其他元素不变, 这也是坐标优化的主要思想.

以上是针对群组变量选择而改进的 PICASSO 算法. 本文提出的 AdaMCPLR 分为 3 个步骤, 在特征选择和集成剪枝中均用到了 PICASSO 算法, 每个步骤的具体算法如算法 1.

算法 1. 特征选择

(1) 输入设计矩阵X; 响应变量y; 参数δ; $\eta\in(0,1)$; 算法迭代次数M;

(2) 选定初始值 $\lambda^{(0)}=\max\{\nabla L(0)\}$ (0 是长度为d的零向量), 初始解 $\beta^{(0)}=(0,0,\cdots,0)^T$;

(3) 定义初始活跃集为:

$$A^{(0)}=\{j|\beta_j^{(0)}=0 \text{ and } \nabla_j L(\beta^{(0)})\geq(1-\delta)\lambda^{(0)}\}\cup\{j|\beta_j^{(0)}\neq 0\},\mathrm{t}=0;\lambda^{(\mathrm{m})}=\lambda^{(0)}\eta^{\mathrm{m}}$$

(4) 计算 $\beta^{(t+0.5)}=\arg\min_\beta F_{\lambda(m)}(), \text{s.t.}\beta_{\bar{A}}=0$;

(5) 更新活跃集$A^{(t|0.5)}=\left\{j|\beta_j^{(t+0.5)}\neq 0\right\}$, $\bar{A}^{(t+0.5)}=\left\{j|\beta_j^{(t+0.5)}=0\right\}$;

(6) $k_t=\arg\max_{k\in\bar{A}^{(t+0.5)}}\left|\nabla_k L(\beta^{(t+0.5)})\right|$, 将非活跃集中的$k_t$加入到活跃集中, 从而活跃集变为 $A^{(t+1)}=A^{(t+0.5)}\cup\{k_t\}$, $\bar{A}^{(t+1)}=\bar{A}^{(t+0.5)}\backslash\{k_t\}$, 直到 $\left|\nabla_k L(\beta^{(t+0.5)})\right|\leq(1+\delta)\lambda$;

重复步骤 (4)–步骤 (6) 直至 M 次迭代结束. 经过特征筛选后, 设计矩阵X包含的特征数量大大减少. 然后将响应向量y重新编码为$\{-1,1\}^n$, 并将简化后的设计矩阵表示为X_{sub}. 本文第 1.2 节的算法如算法 2 所示.

算法 2. 集成

(1) 初始化样本权重$\omega_i=\frac{1}{n}$, $i=1,2,\cdots,n$, n为样本容量, 根据权重为ω_i的样本构造决策树$G_t(x)$.($t=1,2,\cdots,T$);

(2) 计算决策树的加权错误率$err_t=\frac{\sum_{i=1}^n \omega_i I(y_i\neq G_t(x_i))}{\sum_{i=1}^n \omega_i}$;

(3) 根据错误率计算决策树的权重$\alpha_t=\ln((1-err_t)/err_t)$;

(4) 更新样本权重$\omega_i\leftarrow\omega_i\cdot\exp[\alpha_t\cdot I(y_i\neq G_t(x_i))]$, $i=1,2,\cdots,n$;

迭代T次 (T为决策树的数量), 输出最终预测$G(x)=\text{sign}\left(\sum_{t=1}^T \alpha_t G_t(x)\right)$, 将$T$个基学习器的预测结果组成$n\times T$的设计矩阵$Z$, 对响应向量$y$重新编码为$\{0,1\}^n$, 将集成剪枝转化为特征选择问题, 对特定的冗余项$\theta_t(t=1,2,\cdots,T)$压缩到零来达到集成剪枝的目的. 我们再次使用 PICASSO 和 MCP 来解决, 集成剪枝算法的目标函数如式 (20), 集成剪枝算法除了目标函数与算法 1 不一致以外, 其余和算法 1 均相同, 所以此处的集成剪枝算法省略不表.

$$\sum_{i=1}^n\{-y_i z_i\theta+\ln[1+\exp(z_i\theta)]\}+f_{\lambda,\mu}(\theta_j) \quad (19)$$

其中,

$$f_{\lambda,\mu}(\beta_j)=\left(\lambda\beta_j-\frac{\beta_j^2}{2\mu}\right)I(|\beta_j|\leq\mu\lambda)+\frac{\mu\lambda^2}{2}(|\beta_j|>\mu\lambda) \quad (20)$$

3 模拟分析

为了验证提出的 AdaMCPLR 方法的优劣, 本节设置了模拟实验, 将不同的双层变量选择方法进行了对比, 同时将不同的惩罚项对集成进行剪枝的性能也作了对比. 采用选择单个变量数 (*nv*)、选择组变量数 (*ngrp*)、错误发现率 (*FDR*)、假阴性率 (*FNR*) 以及交叉验证错误率 (*cv.error*) 作为变量选择效果评价指标, 其中 *nv* 和 *ngrp* 是越接近真实数值越好, 其他指标则是越低模型越优; 同时采用交叉验证错误率 (*cv.error*)、集成规模 (*Average size*) 作为集成剪枝效果评价指标, 其中 *Average size* 表示经过集成剪枝之后保留的基学习器的个数.

模拟实验的数据生成过程如下: ① 设置样本数为 n, 变量数为 p, 变量分为 m 组, 每组变量数设为 5 (n=200, p=50, 则 m=10); ② 产生随机变量$R_1,\cdots,R_p$, i.i.d$N(0,I)$, 以及标准正态分布向量$Z=(Z_1,\cdots,Z_m)^{\mathrm{T}}$, 其中$E(Z)=0$, $COV\left(Z_{j_1},Z_{j_2}\right)=0.6^{|j_1-j_2|}$; ③ 产生自变量 $X_{5(j-1)+k}=Z_j+R_{5(j-1)+k}/\sqrt{2}, j=1,\cdots,m, k=1,\cdots,5$; ④ 自变量和因变量之间的系数设定为$\omega^{\mathrm{T}}=\left(\omega_1^{\mathrm{T}},\omega_2^{\mathrm{T}},\cdots,\omega_m^{\mathrm{T}}\right)$, 其中 $\omega_1^{\mathrm{T}}=(-1,1,0,1.5,2)$, $\omega_2^{\mathrm{T}}=(0,0,-0.5,0,1)$, $\omega_3^{\mathrm{T}}=(0,1,2,0,0)$, $\omega_4^{\mathrm{T}}=(0,0.5,1,0,0)$, $\omega_5^{\mathrm{T}}=(-0.5,1,0,0,0)$, $\omega_6^{\mathrm{T}}=\ldots=\omega_m^{\mathrm{T}}=0$; ⑤ 因变量$y_i\sim Bernoulli(1,p_i)$, $p_i=prob(y_i=1|x_i)=1/1+\exp\left(-\omega^{\mathrm{T}}x_i\right), i=1,2,\cdots,n$ 模拟重复 100 次, 各评价指标值为 100 次模拟结果的均值.

由表 1 可以看出, 群组变量选择方法中, group LASSO 总是倾向于选择过多的变量, 导致很多非重要变量被误选; group SCAD 恰好相反, 总是倾向于选择过少变量, 导致变量漏选; 在双层变量选择方法中, group bridge 也是漏选变量导致 FNR 指标过大. 综合各指标来看, CMCP 在变量选择环节中表现最优, 这也是我们在提出的 AdaMCPLR 方法中使用 CMCP 作为变量选择方法的原因.

表 1 变量选择模拟结果

方法	*nv*	*ngrp*	*FDR*	*FNR*	*cv.error*
真实模型	12	5	0	0	0
Group LASSO	16.98	4.15	0.42	0.00	0.14
Group SCAD	6.50	2.32	0.03	0.43	0.17
Group MCP	10.23	3.70	0.12	0.17	0.14
Group Bridge	7.45	2.57	0.07	0.25	0.14
SGL	10.40	3.60	0.03	0.16	0.16
CMCP	10.98	4.02	0.007	0.10	0.15

表 2 中的第一行表示未集成剪枝之前模型的各项评价指标值, 其余各行分别表示使用 LASSO、 SCAD 和 MCP 作为集成剪枝惩罚项时模型表现. 将表 2 中的 *cv.error* 值与表 1 中的做对比可以发现, 无论是否进行集成剪枝, 模型的预测精度都有了很大的提升, 同时使用 MCP 函数作为惩罚项, 能够很大限度地精简模型, 同时模型的预测精度也有所提升, 所以本文提出的方法中使用 MCP 函数作为集成剪枝的惩罚项.

表 2 集成剪枝模拟结果

模型	*cv.error*	*Average size*
AdaBoost	0.117	100
LASSO	0.121	34.5
SCAD	0.119	20.3
MCP	0.108	26.6

4 实证检验

当一家公司出现财务困境时, 会对许多外部和内部经济主体产生严重影响, 如股东、公司债权人、客户和供应商、员工以及政府, 因此开发财务困境预测方法一直是金融研究的优先目标之一. 有效的财务困境预测机制不仅可以帮助公司管理者更好地根据自身财务信息提前做好规划, 避免企业遭受财务困境的影响; 也能帮助投资者根据上市公司披露的财务指标信息更好地作出投资选择, 合理规避重大风险; 对于政府监管机构, 有效的财务预警机制有利于政府科学调控经济和优化资源配置, 更好地维护市场稳定.

Beaver[18] 首次提出利用单一指标来构建财务困境预测模型, 但是单一指标并不能涵盖整个财务指标体系对模型的影响, 不能达到理想的预测精度; Altman[19] 提出了 Z 计分模型, 提出使用多变量判别进行财务困境预测研究; Martin[20] 用 logistic回归模型构建财务困境预测模型, 并对银行财务困境预测进行了实证研究; Laitinen 等[21] 也使用了 logistic 回归构建了上市公司财务困境预测模型. 国内方面, 郭斌等[22] 首先构建了完善的预测指标体系, 并在此基础上建立了 logistic 回归财务困境预测模型; 韩立岩等[23] 利用 logistic 回归模型对中小公司的财务困境预测进行了实证研究; 在相当长的一段时间内, logistic 回归都是财务困境预测研究的主流方法.

随着机器学习的理论和技术的不断发展, 越来越多的学者开始使用机器学习算法建立财务困境预测模型. 殷尹[24] 利用 BP 神经网络构建了中短期预测模型, Min 等[25] 利用神经网络与 SVM 和 Logistic 回归等方法分别构建了财务困境预测模型, 发现 SVM 取得的预测效果最优; 丁德臣[26] 提出了基于混合全局优化正交遗传算法和支持向量机的财务困境预警模型; 王小燕等[11] 提出了 adSGL-logit 模型, 并将其应用于信用卡信用评分模型中; 方匡南等[27] 提出了稀疏组 LASSO 支持向量机方法 (SGL-SVM), 并将其应用到我国制造业上市公司财务困境预测中.

当上市公司出现财务异常, 并且该异常可能会导致其股票终止上市, 或者投资者的投资权益可能会受到重大损害时, 证券交易所会对该上市公司股票交易实行特别处理 (Special Treatment, ST). 被特别处理的股票除了会被实行股票报价的日涨跌幅限制为 5% 以外, 证监会还要求其在股票名称前加上提示性注释“ST”,以区别于一般股票.

本文数据来源于 CSMAR 经济金融研究数据库, 以全部上市公司为基础样本, 从中选取截止到 2020 年 6 月底因财务困境被特别处理 (ST) 的公司作为正样本, 其余的上市公司作为负样本. 剔除了正样本中缺失值超过 30% 的指标后, 共选取 280 个财务指标进行建模. 根据 CSMAR 数据库对财务指标的分类, 本文选取了偿债能力、披露财务指标、比率结构、经营能力、盈利能力、现金流分析、风险水平、发展能力共 8 组 280 个指标.

在进行数据处理时, 首先剔除指标缺失超过 30% 的上市公司, 其余的缺失值用指标均值代替; 然后对所有指标进行 Z-score 标准化处理, 从而消除指标量纲对模型的影响. 最终得到 3451 个样本, 其中正样本 83 个, 负样本 3368 个, 正负样本的数量差异很大, 是非常典型的非平衡数据, 需要采用重抽样的方法使其正负样本均衡. 因此, 本文采用双层重抽样的方法对样本进行处理, 即通过上采样来增加少数类样本的数量, 同时利用下采样来减少多数类样本, 从而来获取相对平衡的数据集. 本文利用 SMOTE 算法[28] 对少数类进行过抽样, SMOTE 算法的基本原理大致描述为: 在每个少数类样本 a 的最近邻样本中随机选择一个样本 b, 接着在样本 a 和 b 之间的连线上随机选取一点作为新的少数类样本. Python 中的 imblearn 包提供了更方便的接口, 本文直接调用 imblearn 中对应的 SMOTE 方法和下采样中的 RandomUnderSampler 方法来对样本进

行双层重抽样, 从而使正负例比例接近 1:1, 最终重抽样之后的总样本个数为 830 个. 在总样本中随机抽取 70% 作为训练集, 剩余的 30% 的样本作为模型的交叉验证. 接下来, 本文利用 AdaMCPLR 对处理后的所有样本进行建模, 经过特征选择之后, 保留了 39 个重要指标. 用这 39 个重要性指标数据进行集成及集成剪枝. 在此处设置 200 棵决策树来构建 AdaBoost模型, 并用 MCP 函数来进行集成剪枝, 从而得到最终结果. 将最终 AdaMCPLR 方法的变量选择效果和预测效果与 logistic 回归、SVM、SGL-SVM 以及 adSGL 方法的进行了对比, 对比结果如表 3.

表 3 集成剪枝模拟结果

方法	*cv.error*	*TPR*	*TNR*	*AUC*	*nv*	*Average size*
logistic	0.17	0.64	0.87	0.77	280	—
SVM	0.23	0.63	0.86	0.74	280	—
SGL-SVM	0.14	0.73	0.83	0.83	48	—
adSGL	0.11	0.76	0.85	0.85	43	—
AdaMCPLR	0.10	0.71	0.85	0.89	39	47

由表 3 可知, 传统的 logistic 回归和 SVM 方法均不具有变量选择的作用, 保留了全部的变量, 但是预测精度和 AUC 值均不高. AdaMCPLR 的预测精度和 AUC 值较其他预测方法都是最高的, 另外从选择的变量个数来看, AdaMCPLR 仅选择了 39 个变量, 就达到了很高的样本外预测精度, 同时经过集成剪枝, 模型只保留了 47 个基学习器. 由此可以看出, 并不是越多变量参与模型, 模型就越优, 反而过多的冗余变量进入模型, 会降低模型的预测精度.

5 总结

针对高维群组变量下的二分类问题, 本文提出的 AdaMCPLR 分为两阶段, 利用惩罚项既对群组变量进行了特征选择, 又对 AdaBoost 集成模型的基学习器进行了剪枝, 这是在已有的高维群组变量数据分类预测问题上所没有出现过的方法.

将 CMCP 用于双层变量筛选, 降低了数据的冗余度, 提高变量选择的精度, 用特征选择后的数据生成 AdaBoost 集成, 大大提高了模型的预测精度, 由于集成模型由大量基学习器组成, 使用 MCP惩罚方法对这些基学习器进行剪枝, 选择的基学习器组合为最终的预测集合, 在提高预测精度的同时, 也大大降低了集成模型的空间消耗. 本文还对 PICASSO 算法进行了改进, 使其能够应用于群组变量选择问题, 大大提高了模型的求解速度.

本文提出的 AdaMCPLR 方法适用于生物信息、管理科学、经济学、金融学等领域高维群组变量数据分类预测问题, 在降低空间消耗的同时, 较已有的方法提高了预测精度.

参考文献

1 Frank IE, Friedman JH. A statistical view of some chemometrics regression tools. Technometrics, 1993, 35(2): 109–135. [doi: 10.1080/00401706.1993.10485033]

2 Fan JQ, Li RZ. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American Statistical Association, 2001, 96(456): 1348–1360. [doi: 10.1198/016214501753382273]

3 Huang J, Ma SG, Xie HL, *et al*. A group bridge approach for variable selection. Biometrika, 2009, 96(2): 339–355. [doi: 10.1093/biomet/asp020]

4 Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2006, 68(1): 49–67. [doi: 10.1111/j.1467-9868.2005.00532.x]

5 Meier L, Van De Geer S, Bühlmann P. The group lasso for logistic regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2008, 70(1): 53–71. [doi: 10.1111/j.1467-9868.2007.00627.x]

6 Wang LF, Chen G, Li HZ. Group SCAD regression analysis for microarray time course gene expression data. Bioinformatics, 2007, 23(12): 1486–1494. [doi: 10.1093/bioinformatics/btm125]

7 Huang J, Breheny P, Ma SG. A selective review of group selection in high-dimensional models. Statistical Science, 2012, 27(4): 481–499.

8 姜叶飞. 惩罚变量选择方法比较分析及其在信用卡信用风险中的应用 [硕士学位论文]. 厦门: 厦门大学, 2014.

9 Breheny P, Huang J. Penalized methods for bi-level variable selection. Statistics and its Interface, 2009, 2(3): 369–380. [doi: 10.4310/SII.2009.v2.n3.a10]

10 Simon N, Friedman J, Hastie T, *et al*. A sparse-group lasso. Journal of Computational and Graphical Statistics, 2013, 22(2): 231–245. [doi: 10.1080/10618600.2012.681250]

11 王小燕, 方匡南, 谢邦昌. Logistic 回归的双层变量选择研究. 统计研究, 2014, 31(9): 107–112. [doi: 10.3969/j.issn.1002-4565.2014.09.017]

12 郑伟华. 基于 MCP 惩罚的 AdaBoost 集成剪枝技术的研

究 [硕士学位论文]. 南昌: 江西财经大学, 2019.
13 Margineantu DD, Dietterich TG. Pruning adaptive boosting. Proceedings of the Fourteenth International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers Inc., 1997. 211–218.
14 Lazarevic A, Obradovic Z. Effective pruning of neural network classifier ensembles. Proceedings of the IJCNN'01. International Joint Conference on Neural Networks. Washington DC: IEEE, 2001. 796–801.
15 Azmi M, Berrado A. Class-association rules pruning using regularization. Proceedings of 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications. Agadir: IEEE, 2016.
16 Jiang H, Zheng WH, Luo LQ, *et al*. A two-stage minimax concave penalty based method in pruned AdaBoost ensemble. Applied Soft Computing, 2019, 83: 105674. [doi: 10.1016/j.asoc.2019.105674]
17 Zhao T, Liu H, Zhang T. Pathwise coordinate optimization for sparse learning: Algorithm and theory. The Annals of Statistics: An Official Journal of the Institute of Mathematical Statistics, 2018, 46(1): 180–218.
18 Beaver WH. Financial Ratios as Predictors of Failure. Journal of Accounting Research, 1966, 4(1): 71–111.
19 Altman EI. Financial Ratios. Discriminant Analysis and the Prediction of Corporate Bankruptcy. The Journal of Finance, 1968, 23(4): 589–609.
20 Martin D. Early warning of bank failure: A logit regression approach. Journal of Banking & Finance, 1977, 1(3): 249–276.
21 Laitinen EK, Laitinen T. Bankruptcy prediction: Application of the Taylor's expansion in logistic regression. International Review of Financial Analysis, 2001, 9(4): 327–349.
22 郭斌, 戴小敏, 曾勇, 等. 我国企业危机预警模型研究-以财务与非财务因素构建. 金融研究, 2006, (2): 78–87.
23 韩立岩, 李蕾. 中小上市公司财务危机判别模型研究. 数量经济技术经济研究, 2010, 27(8): 102–115.
24 殷尹. 公司财务困境预测及成本估计 [博士学位论文]. 合肥: 中国科学技术大学, 2002.
25 Min JH, Lee YC. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Systems with Applications, 2005, 28(4): 603–614. [doi: 10.1016/j.eswa.2004.12.008]
26 丁德臣. 混合 HOGA-SVM 财务风险预警模型实证研究. 管理工程学报, 2011, 25(2): 37–44, 36. [doi: 10.3969/j.issn.1004-6062.2011.02.006]
27 方匡南, 杨阳. SGL-SVM 方法研究及其在财务困境预测中的应用. 统计研究, 2018, 35(8): 104–115.
28 Chawla NV, Bowyer KW, Hall LO, *et al*. Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research. 2002, 16: 321–357.