计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2024/18(02)-0496-10
doi: 10.3778/j.issn.1673-9418.2211073

采用快速迁移模型的集成特征选择方法

宁保斌,王士同+
江南大学 人工智能与计算机学院,江苏 无锡 214122
+ 通信作者 E-mail: wxwangst@aliyun.com

摘　要:相较于传统集成特征选择方法,目前的基于块正则化 $m\times2$ 交叉验证的集成特征选择方法(EFSBCV)不仅具有估计量的方差小于随机 $m\times2$ 交叉验证的方差之特点,而且提高了重要特征的入选概率,降低了噪声特征的入选概率。但EFSBCV所采用的线性回归模型因只有误差项而不包含偏置项,故拟合出来的超平面总是过原点的,因而很容易导致欠拟合,而且EFSBCV没有考虑每个特征子集的重要程度。针对EFSBCV方法存在的这两点问题,提出了基于快速迁移模型的集成特征选择方法(EFSFT)。基本思想是EFSBCV中的基特征选择器采用提出的快速迁移模型,从而引入了偏置项,EFSFT将 $2m$ 个特征子集作为源知识进行迁移,然后重新量化每个特征子集的权重,加入偏置项的线性模型拟合能力更好。真实数据实验表明,EFSFT相对于EFSBCV,FP平均值降低了58%,证明EFSFT在去除噪声特征方面更具优势。EFSFT相对于最小二乘支持向量机(LSSVM),TP平均值提高了5%,证明EFSFT在筛选重要特征方面更具优势。

关键词:集成特征选择;交叉验证;迁移学习;回归
文献标志码:A　　**中图分类号:**TP181

Ensemble Feature Selection Method with Fast Transfer Model

NING Baobin, WANG Shitong+
School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, Jiangsu 214122, China

Abstract: Compared with the traditional ensemble feature selection methods, the recently-developed ensemble feature selection with block-regularized $m\times2$ cross-validation (EFSBCV) not only has a variance of the estimator smaller than that of random $m\times2$ cross-validation, but also enhances the selection probability of important features and reduces the selection probability of noise features. However, the adopted linear regression model without the use of the bias term in EFSBCV may easily lead to underfitting. Moreover, EFSBCV does not consider the importance of each feature subset. Aiming at these two problems, an ensemble feature selection method called EFSFT (ensemble feature selection method using fast transfer model) is proposed in this paper. The basic idea is that the base feature selector in EFSBCV adopts the fast transfer model in this paper, so as to introduce the bias term. EFSFT transfers $2m$ subsets of features as the source knowledge, and then recalculates the weight of each feature subset, and the linear model fitting ability with the addition of bias terms is better. The results on real datasets show that compared with EFSBCV, the average FP value by EFSFT reduces up to 58%, proving that EFSFT has more advantages in removing noise features. In contrast to least-squares support vector machine (LSSVM), the average TP value by EFSFT increases up to 5%, which clearly indicates the superiority of EFSFT over LSSVM in choosing important features.

Key words: ensemble feature selection; cross-validation; transfer learning; regression

基金项目:江苏省自然科学基金(BK20191331)。
This work was supported by the Natural Science Foundation of Jiangsu Province (BK20191331).
收稿日期:2022-11-18　　**修回日期:**2023-02-23

特征选择（feature selection）是模式识别中数据预处理阶段重要的步骤，研究人员希望通过特征选择来了解数据集的结构，过去二十年受到了广泛的关注。特征选择是指从给定的特征集合中筛选出相关特征子集，去掉冗余的特征，从而达到降维、提升模型性能、提高通用性及降低过拟合风险的目的。

特征选择有三种方法，分别是过滤法（filter）[1]、包装法（wrapper）[2]和嵌入法（embedded）[3]，其中过滤式方法的思想是给所有特征按照相关性赋代表重要性的权重，通过设置阈值或者待选择特征的个数得出重要特征。过滤法是先进行特征选择，然后用筛选后的特征子集来训练分类器，因此特征选择的过程和分类器无关，主要方法包括卡方检验（Chi-squared test）[4]、信息增益（information gain）和相关系数（correlation coefficient scores）。包装法的思想是将特征子集生成不同的组合，然后对组合进行比较和评价，这个过程可以看作一个优化问题，主要方法包括递归特征消除算法[5]、完全搜索、启发式搜索、随机搜索。嵌入法主要思想是在模型既定的情况下筛选出提升模型准确性最好的特征，将特征选择的过程与模型构建过程相结合，既具有包装法与算法相结合的优点，又具有过滤法计算效率高的优点。

单个特征选择方法的结果易受数据的影响，具有很大的不确定性，对于以上单一方法建模的输出结果研究人员不是很认可。为了克服单个特征选择方法的缺点产生了集成特征选择方法（ensemble feature selection，EFS），该方法建立在上述特征选择方法的基础上，将单个特征选择方法产生的特征子集组合起来产生最终的特征子集。根据基本特征选择器是否相同，集成特征选择方法分为同质的和异质的，本研究的重点是同质集成特征选择。由 Meinshausen 和 Buhlmann 在 2010 年提出的 StabSel（stability selection）[6]吸引了广泛的关注，目的是为了提高特征选择算法的性能，该方法用一个基特征选择器在随机无重复的抽出的子样本（大小为 $\lfloor n/2 \rfloor$）上做特征选择，重复多次，将最频繁被选择的特征选入最终的特征集合，因此 StabSel 与 bagging[7-8]和 subagging[9]关系密切。Shah 等人在 StabSel 的基础上进行改进，提出了 CPSS（complementary pairs' StabSel）[10]的集成特征选择方法。该方法在互补的子集对上进行特征选择，同样重复多次，然后将入选概率最高的特征选入最终的特征集合，预期的预测误差降低了，在相同水平误差控制和新的误差界限下 CPSS 比 StabSel 可以选择更多的变量。Yang 等人提出了基于块正则化 $m\times2$ 交叉验证（$m\times2$ BCV）的集成特征选择方法 EFSBCV（ensemble feature selection with block-regularized $m\times2$ cross-validation）[11]，该方法将 CPSS 的数据抽样方式改为 $m\times2$ BCV，特征入选的阈值设为 0.5，不仅改正了 StabSel 对于无关特征的可交换性和基特征选择器的性能有要求的缺陷，而且克服了 CPSS 的缺点，能够做到同时兼顾低入选概率特征和高入选概率特征，既提高了高入选特征的入选概率，又降低了低入选特征的入选概率。Bach 提出的 Bolasso[12]是在自助抽样样本上获取多个候选子集，然后把这些候选子集的交集作为最终的特征子集。以上提到的特征选择方法适用于监督学习，而无监督学习中的特征选择方法是一个更具挑战性的问题。无监督学习中的集成特征选择方法可以参考文献[13-14]，半监督学习的集成特征选择方法可以参考文献[15]。

对于上面提到的 EFSBCV，它的线性模型只有误差项不包含偏置项，拟合出来的超平面总是过原点的，但真实数据集大部分不符合以 0 为中心的分布，就容易出现收敛速度过慢、精度不高的问题。而且在做回归分析时，总是假设随机误差项是符合正态分布的，这样的条件也很难满足。文献[16-17]通过实验证明，在线性模型中加入偏置项，不仅提高了模型的收敛速度而且改善了泛化性能。受此启发，将 EFSBCV 算法得到的 $2m$ 个候选特征子集作为源知识进行迁移，重新量化每个特征子集的权重而不是直接求均值，提出了本文的基于 LSSVM（least-squares support vector machine）[18]快速迁移模型，借助该算法将模型的误差项改为偏置项，来克服精度过低的问题，改善泛化性能。修改后的模型不仅保留了原 EFSBCV 模型同时兼顾低入选概率特征和高入选概率特征的优点，更提高了模型的泛化能力，从而使算法对不同数据集的样本适应能力更强，在去除噪声特征方面更具优势。

1 相关理论

当数据集比较大时，可以将所有数据分为训练集、验证集和测试集三部分。训练集用来拟合模型，验证集用来评估不同模型的泛化误差，测试集则用来评价所选模型的性能；而当数据集比较小的时候，则可以借助交叉验证[19]的方法来避免发生过拟合。交叉验证的用途有两个：模型评估和模型选择，当把

交叉验证运用于划分训练集和测试集，就可以实现模型评估，把交叉验证运用于划分训练集和验证集，则可以实现模型选择。

$m\times2$ 交叉验证[20]是 m 次的2折交叉验证，块正则化 $m\times2$ 交叉验证[11]是带有某些正则化条件的 $m\times2$ 交叉验证。2折交叉验证会产生一对互补的数据子集，分别作为训练集和验证集，两个数据子集的并集是原数据集，等价于产生了两个数据集；而 m 次的2折交叉验证即 $m\times2$ 交叉验证会产生 m 对互补的训练集 $I_t(s_l)$ 和验证集 $I_v(s_l)$，其中 $l=1,2,\cdots,m$，相当于产生了 $2m$ 个数据集，对于其中任意的两对互补训练集和验证集，$I_t(s_i)$ 与 $I_t(s_j)$ 之间重复样本的数量用 ϕ_{ij} 表示，ϕ_{ij} 为整数变量，随机分布在 $[0,n/2]$ 上，期望为 $n/4$，则 $I_t(s_i)$ 与 $I_v(s_j)$，$I_v(s_i)$ 与 $I_t(s_j)$，$I_v(s_i)$ 与 $I_v(s_j)$ 之间的重复样本数量分别为 $n/2-\phi_{ij}$、$n/2-\phi_{ij}$ 和 ϕ_{ij}。相对于 $m\times2$ 交叉验证，块正则化 $m\times2$ 交叉验证所添加的正则化条件是 $|\phi_{ij}-n/4|\leqslant c$，其中 c 为正则化参数。EFSBCV[8]的创新在于将块正则化 $m\times2$ 交叉验证引入到集成特征选择算法，样本数量设置为4的倍数，ϕ_{ij} 设置为 $n/4$，c 设置为0。

EFSBCV中在数据集大小为 n 的数据集 D_n 上，特征 X_f 通过基特征选择器 $\mathbb{F}$ 选出的特征选择概率定义为：

$$P(X_f|\mathbb{F})=p_{f,n/2,\mathbb{F}} \tag{1}$$

通常重要特征的选择概率应该大于0.5，噪声特征的选择概率应该小于0.5，在数据集大小为 n 的情况下，高选择概率的特征的索引集合用 $H_{0.5,n/2}=\{f:p_{f,n/2,\mathbb{F}}>0.5\}$ 表示，低入选概率的特征的索引集合用 $L_{0.5,n/2}=\{f:p_{f,n/2,\mathbb{F}}<0.5\}$ 表示。

EFSBCV采用的线性模型为 $y=X_1\beta_1+X_2\beta_2+\cdots+X_d\beta_d+\varepsilon$，其中 ε 为误差项，每个特征 X_f 对应的权重 β_f 为0，则将 X_f 归为噪声特征，不为0，则归为重要特征，在 $m\times2$ BCV划分的 $2m$ 个大小为 $n/2$ 的数据集上使用特征选择方法lasso进行训练，训练后将得到 $2m$ 组不同特征权重，将特征权重不为0的记为1，为0的还是记为0，可以计算出每个特征在 $2m$ 次拟合中被选择的概率。

将块正则化 $m\times2$ 交叉验证引入到集成特征选择算法的EFSBCV，相比较于其他集成特征选择算法，只用两个相关系数来简化特征选择结果之间的相关结构，使用更精准的beta分布来近似频率分布，不仅提高了重要特征的入选概率，降低了噪声特征的入选概率，而且估计量的方差小于随机 $m\times2$ 交叉验证的方差。但是EFSBCV直接将求得的 $2m$ 个特征子集求平均值并不合理，而且EFSBCV的线性模型只有误差项不包含偏置项，拟合出来的超平面总是过原点的，但真实数据集大部分不符合以0为中心的分布，很容易导致欠拟合，由此引出了本文提出的集成特征选择方法。

2 采用快速迁移模型的集成特征选择方法

2.1 算法原理

本节提出的集成特征选择方法，是基于 $m\times2$ BCV集成特征选择方法做出的改进。由第1章可知，EFSBCV的线性模型只包含误差项 ε，本节将介绍通过快速迁移模型将偏置项引入到线性模型中的集成特征选择方法EFSFT（ensemble feature selection method using fast transfer model），EFSFT包括参数选择和特征选择两个模块。

对于参数选择模块，本模块和文献[11]一致，选用lasso拟合数据，通过比较预测误差，选出最优 λ。首先设定 λ 的取值范围，在一个大小为 n 的数据集 D_n 上使用 $m\times2$ BCV，设置 m 初始值为3，逐渐增大 m，每增大1，将预测误差最大的一半 λ 舍去，λ 的取值范围则缩小一半，直至 λ 剩下最后一个最优值 $\hat{\lambda}$，采用最优 $\hat{\lambda}$ 来进行后面的集成特征选择。

对于特征选择模块，这是本文重点做出改动的部分，选用快速迁移模型做基特征选择器。首先由第1章可知，在一个大小为 n 的数据集 D_n 上使用 $m\times2$ BCV[11]，可以得到 $2m$ 个大小为 n 的数据集，每个数据集的一半为训练集，另一半为验证集。在这 $2m$ 个数据集上通过特征选择器lasso可以拟合出 $2m$ 个不同的线性模型，其中lasso的参数 λ 则选用参数选择模块选取的最优 $\hat{\lambda}$，这 $2m$ 个不同的线性模型对应着权重值 $\boldsymbol{\beta}_l, l=1,2,\cdots,2m$，$\boldsymbol{\beta}_l$ 为 d 维向量。

文献[11]将 $2m$ 特征子集转化为二值矩阵后直接求均值得出最终模型，每个模型的权重分配并不合理，受迁移学习[21-22]的启发，本文将lasso训练后的 $2m$ 个模型作为源知识，经过线性组合后引入到LSSVM[18]模型中，生成了本文快速迁移模型，该模型可以对源知识进行适当加权，在特征选择模块的第 l 次迭代中优化目标为：

$$\min J(\boldsymbol{\omega}_l,b_l,\hat{\boldsymbol{\mu}},\boldsymbol{e})=\frac{1}{2}\boldsymbol{\omega}_l^{\mathrm{T}}\boldsymbol{\omega}_l+\frac{\gamma}{2N}\sum_{i=1}^{N}e_i^2+\frac{\eta}{2N}\sum_{i=1}^{N}\left[y_i-e_i-\sum_{j=1}^{J}\hat{\mu}_j(\hat{y}_j-\hat{e}_j)\right]^2 \tag{2}$$

$$\text{s.t. } e_i=y_i-(\boldsymbol{\omega}_l^{\mathrm{T}}\boldsymbol{x}_i+b),i=1,2,\cdots,N \tag{3}$$

式(3)为约束条件，与LSSVM不同的是引入了一项 $\frac{\eta}{2N}\sum_{i=1}^{N}\left[y_i-e_i-\sum_{j=1}^{J}\hat{\mu}_j(\hat{y}_j-\hat{e}_j)\right]^2$，其中 $\sum_{j=1}^{J}\hat{\mu}_j(\hat{y}_j-\hat{e}_j)$ 为源知识的线性组合，y_i-e_i 为新的线性模型，通过优化，拟合出新的线性模型，本文 J 取值为 $2m$。$\sum_{j=1}^{J}\hat{\mu}_j(\hat{y}_j-\hat{e}_j)$ 可以写为向量的形式 $\hat{\boldsymbol{\mu}}_i^{\mathrm{T}}\hat{\boldsymbol{Y}}_i$，$\hat{\boldsymbol{\mu}}_i$ 表示源知识权重的列向量，$\hat{\boldsymbol{Y}}_i$ 表示源模型的预测值列向量，其中：

$$\hat{\boldsymbol{Y}}_i=\boldsymbol{\beta}\boldsymbol{x}_i \tag{4}$$

那么式(2)可以写为：

$$\min J(\boldsymbol{\omega}_l,b_l,\hat{\boldsymbol{\mu}},\boldsymbol{e})=\frac{1}{2}\boldsymbol{\omega}_l^{\mathrm{T}}\boldsymbol{\omega}_l+\frac{\gamma}{2N}\sum_{i=1}^{N}e_i^2+\frac{\eta}{2N}\sum_{i=1}^{N}\left[y_i-e_i-\hat{\boldsymbol{\mu}}_i^{\mathrm{T}}\hat{\boldsymbol{Y}}_i\right]^2 \tag{5}$$

对式(5)、式(3)使用拉格朗日乘子法可以得到其对偶问题(dual problem)，该问题的拉格朗日函数可以写为：

$$L(\boldsymbol{\omega}_l,b_l,\hat{\boldsymbol{\mu}},\boldsymbol{e};\boldsymbol{\alpha})=J(\boldsymbol{\omega}_l,b_l,\hat{\boldsymbol{\mu}},\boldsymbol{e})+\sum_{i=1}^{N}\alpha_i(y_i-(\boldsymbol{\omega}_l^{\mathrm{T}}\boldsymbol{x}_i+b)-e_i) \tag{6}$$

其中，$\boldsymbol{\alpha}=(\alpha_1;\alpha_2;\cdots;\alpha_N)$ 为拉格朗日乘子，式(6)对 $\boldsymbol{\omega}_l$、b_l、$\boldsymbol{e}$、$\boldsymbol{\alpha}$、$\hat{\boldsymbol{\mu}}$ 分别求偏导为0可得：

$$\frac{\partial L}{\partial\boldsymbol{\omega}_l}=0\rightarrow\boldsymbol{\omega}_l=\sum_{i=1}^{N}\alpha_i\boldsymbol{x}_i \tag{7}$$

$$\frac{\partial L}{\partial b_l}=0\rightarrow\sum_{i=1}^{N}\alpha_i=0 \tag{8}$$

$$\frac{\partial L}{\partial e_i}=0\rightarrow e_i=\frac{\eta(y_i-\hat{\boldsymbol{\mu}}_i^{\mathrm{T}}\hat{\boldsymbol{Y}}_i)}{\gamma+\eta}+\frac{N\alpha_i}{\gamma+\eta} \tag{9}$$

$$\frac{\partial L}{\partial\alpha_i}=0\rightarrow\frac{\gamma}{\gamma+\eta}y_i=\sum_{j=1}^{N}\alpha_j\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{x}_i+\frac{N\alpha_i}{\gamma+\eta}+b_l-\frac{\eta}{\gamma+\eta}\hat{\boldsymbol{\mu}}_i^{\mathrm{T}}\hat{\boldsymbol{Y}}_i \tag{10}$$

$$\frac{\partial L}{\partial\hat{\boldsymbol{\mu}}_i}=0\rightarrow\frac{N}{\gamma}\alpha_i\hat{\boldsymbol{Y}}_i+\hat{\boldsymbol{Y}}_i\hat{\boldsymbol{Y}}_i^{\mathrm{T}}\boldsymbol{\mu}_i=\hat{\boldsymbol{Y}}_i y_i \tag{11}$$

式(7)~(11)用矩阵形式可以表示为：

$$\begin{bmatrix}\mathbf{1}_N^{\mathrm{T}} & 0 & 0\\ \boldsymbol{K}+\frac{N}{\gamma+\eta}\boldsymbol{I}_N & \mathbf{1}_N & \boldsymbol{M}_{23}\\ \boldsymbol{M}_{31} & 0 & \boldsymbol{M}_{33}\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_l\\ b_l\\ \hat{\boldsymbol{\mu}}\end{bmatrix}=\begin{bmatrix}0\\ \frac{\gamma}{\gamma+\eta}\boldsymbol{Y}\\ \boldsymbol{M}\end{bmatrix} \tag{12}$$

经过求逆可以求出 b_l 和 $\boldsymbol{\alpha}_l$ 的值，其中 $l=1,2,\cdots,2m$：

$$\begin{bmatrix}\boldsymbol{\alpha}_l\\ b_l\\ \hat{\boldsymbol{\mu}}\end{bmatrix}=\begin{bmatrix}\mathbf{1}_N^{\mathrm{T}} & 0 & 0\\ \boldsymbol{K}+\frac{N}{\gamma+\eta}\boldsymbol{I}_N & \mathbf{1}_N & \boldsymbol{M}_{23}\\ \boldsymbol{M}_{31} & 0 & \boldsymbol{M}_{33}\end{bmatrix}^{-1}\begin{bmatrix}0\\ \frac{\gamma}{\gamma+\eta}\boldsymbol{Y}\\ \boldsymbol{M}\end{bmatrix} \tag{13}$$

其中，$\boldsymbol{Y}=[y_1,y_2,\cdots,y_N]^{\mathrm{T}}$，$\mathbf{1}_N=[1,1,\cdots,1]^{\mathrm{T}}$，$\boldsymbol{\alpha}_l^k=[\alpha_1,\alpha_2,\cdots,\alpha_N]^{\mathrm{T}}$，$\boldsymbol{I}_n$ 是一个 $N\times N$ 的单位矩阵，$\boldsymbol{K}\in\mathbb{R}^{N\times N}$ 是一个线性核矩阵。请注意：由于EFSBCV[11]采用线性模型且 $m\times 2$ BCV 数据分块规模不大，本文遵循该方案，而且本文追求的是快速求解，因此选用的是线性核函数，线性核函数虽然不能实现非线性转化，但在实际运用时往往简单有效且安全可行。式(13)中的 $\boldsymbol{M}$、$\boldsymbol{M}_{23}$、$\boldsymbol{M}_{31}$、$\boldsymbol{M}_{33}$ 分别如下，可由式(14)~(17)求出：

$$\boldsymbol{M}=\begin{bmatrix}\hat{\boldsymbol{Y}}_1 y_1\\ \hat{\boldsymbol{Y}}_2 y_2\\ \vdots\\ \hat{\boldsymbol{Y}}_N y_N\end{bmatrix} \tag{14}$$

$$\boldsymbol{M}_{23}=-\frac{\eta}{\gamma+\eta}\begin{bmatrix}\hat{\boldsymbol{Y}}_1^{\mathrm{T}} & 0 & \cdots & 0\\ 0 & \hat{\boldsymbol{Y}}_2^{\mathrm{T}} & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & \hat{\boldsymbol{Y}}_N^{\mathrm{T}}\end{bmatrix} \tag{15}$$

$$\boldsymbol{M}_{31}=\frac{N}{\gamma}\begin{bmatrix}\hat{\boldsymbol{Y}}_1 & 0 & \cdots & 0\\ 0 & \hat{\boldsymbol{Y}}_2 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & \hat{\boldsymbol{Y}}_N\end{bmatrix} \tag{16}$$

$$\boldsymbol{M}_{33}=\begin{bmatrix}\hat{\boldsymbol{Y}}_1\hat{\boldsymbol{Y}}_1^{\mathrm{T}} & 0 & \cdots & 0\\ 0 & \hat{\boldsymbol{Y}}_2\hat{\boldsymbol{Y}}_2^{\mathrm{T}} & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & \hat{\boldsymbol{Y}}_N\hat{\boldsymbol{Y}}_N^{\mathrm{T}}\end{bmatrix} \tag{17}$$

经过求解后的新线性模型为 $y=\boldsymbol{\omega}_l^{\mathrm{T}}\boldsymbol{x}+b_l=\boldsymbol{\omega}_1\boldsymbol{x}_1+\boldsymbol{\omega}_2\boldsymbol{x}_2+\cdots+\boldsymbol{\omega}_d\boldsymbol{x}_d+b_l$，其中 b_l 由式(13)求得，$\boldsymbol{\omega}_l$ 由式(13)求出的 $\boldsymbol{\alpha}_l$ 代入式(7)求得，$\boldsymbol{\omega}_l$ 为新线性模型的权重，这就完成了从误差项到偏置项的转变。本文按照模型的权重的绝对值大小排序，参照过滤法(Filter)的思想，预设选取个数进行选取，预设值一般取为数据集原本特征的个数。到这里已经完成了重要特征的选择，但为了方便进行评估，与EFSBCV算法进行

比较，本文被选取的新的权重值也可以用一个大小为 $2m\times d$ 的二进制矩阵 $\boldsymbol{Z}$ 来表示。

$$\boldsymbol{Z}=\begin{pmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,d} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,d} \\ \vdots & \vdots & & \vdots \\ z_{2m,1} & z_{2m,2} & \cdots & z_{2m,d} \end{pmatrix} \tag{18}$$

矩阵的每一行代表一个候选特征子集，在第 f 个位置处，数值为1则表示特征 X_f 已被选择，数值为0则表示特征 X_f 已被抛弃。$\hat{S}_{l,n/2}=\{f\in\{1,2,\cdots,d\}:z_{l,f}=1\},l\in\{1,2,\cdots,2m\}$ 为在矩阵的每一行值为1的元素集，矩阵 $\boldsymbol{Z}$ 的第 f 列 $Z_{f,n}=(z_{1,f},z_{2,f},\cdots,z_{2m,f})^{\mathrm{T}}$ 代表特征 X_f 经过 $2m$ 次选择的结果。接下来的操作和EFSBCV[11]相同，按照公式：

$$\hat{p}_{f,n/2,m}=\frac{1}{2m}\sum_{l=1}^{2m}\mathbb{I}(f\in\hat{S}_{l,n/2})=\frac{1}{2m}\sum_{l=1}^{2m}z_{l,f} \tag{19}$$

式中，$\mathbb{I}(\cdot)$ 为指示函数，在 $\cdot$ 为真和假时分别取值为1、0，将二进制矩阵 $\boldsymbol{Z}$ 按列求和再求平均值，就会得出每个特征的入选概率，大于0.5的特征入选重要特征，小于0.5的特征则是噪声特征。经过集成特征选择后特征子集可以表示为：

$$\hat{S}_n=\{f:\hat{p}_{f,n/2,m}>0.5\} \tag{20}$$

其中，阈值0.5是基于多数投票而做出的选择。

2.2 算法描述

本节介绍算法具体过程，其中算法步骤1为采用 $m\times 2$ BCV 方法进行数据集划分模块，算法步骤2~8是为lasso参数选择模块，算法步骤9~11是用lasso训练模型，算法步骤12~19是特征选择模块，其中算法步骤12~16是本文所引入的快速迁移模型的特征选择模块，其余的算法步骤和文献[11]一致。

算法1 特征选择算法

输入：维度为 d 大小为 n 数据集 D_n，调节参数 λ 的候选集合 $\Lambda=\{\lambda_r,r=1,2,\cdots,R\}$。

输出：特征子集 $\hat{S}_n$。

1. $\mathbb{S}\leftarrow m\times 2$ BCV 得到的切分数据集合（一共 $2m$ 个，互补子集互为训练集和验证集）

$\{<S_l,S_l^{\mathrm{T}}>:S_l=(I_t(s_l^1),I_v(s_l^1)),S_l^{\mathrm{T}}=(I_t(s_l^2),I_v(s_l^2)),l=1,2,\cdots,m\}$

2. for all $l=3;l\leqslant m;l++$ do

3. 用lasso在训练集 $I_t(s_l^k),k=1,2$ 上计算回归系数估计的集合 $\{\hat{\beta}_{l,\lambda}^k,\lambda\in\Lambda,k=1,2\}$；

4. 用lasso在验证集 $I_v(s_l^k),k=1,2$ 计算预测误差估计，结果记为 $\{L_{l,\lambda}^k,\lambda\in\Lambda,k=1,2\}$；

5. 计算平均预测误差 $L_\lambda=\frac{1}{2m}\sum_{l=1}^{m}\sum_{k=1}^{2}L_{l,\lambda}^k$；

6. λ 的范围缩小一半；

7. end for

8. $\hat{\lambda}\leftarrow\arg\min_{\lambda\in\Lambda}L_\lambda$，取得 λ 最优值；

9. for all $l=1;l\leqslant 2m;l++$ do

10. $\beta_l\leftarrow\lambda$ 取值为 $\hat{\lambda}$ 的lasso在数据集 x_l 上对应的回归系数；

11. end for

12. for all $l=1;l\leqslant 2m;l++$ do

13. 根据式(4)将回归系数 β 和数据集 $\boldsymbol{x}_l$ 代入到快速迁移模型，由式(13)求出 $\boldsymbol{\alpha}_l$ 和 b_l；

14. 根据式(7)计算出快速迁移模型的新的回归系数 $\boldsymbol{\omega}_l$；

15. $\boldsymbol{Z}$ 的第 l 行 $(z_{l,1},z_{l,2},\cdots,z_{l,d})^{\mathrm{T}}\leftarrow$ 快速迁移模型在数据集 $\boldsymbol{x}_l$ 上得到的特征选择结果，根据式(15)可以表示为长度 d 的0-1二值向量；

16. end for

17. $\hat{p}_{f,n/2,m}\leftarrow$ 矩阵按列求均值得到每个特征的入选概率 $\frac{1}{2m}\sum_{l=1}^{2m}z_{l,f}$；

18. $\hat{S}_n\leftarrow\hat{p}_{f,n/2,m}$ 中大于0.5的指标集 $\hat{S}_n=\{f:\hat{p}_{f,n/2,m}>0.5\}$；

19. return $\hat{S}_n$

2.3 算法复杂度分析

由2.2节可知算法整体可分为4个模块，算法步骤1为使用 $m\times 2$ BCV进行数据划分，涉及到矩阵运算（时间复杂度一般为 $O(n^3)$），循环了 m 次，因此时间复杂度为 $O(mn^3)$，其中 n 为训练集样本数，算法步骤2~8是为lasso参数选择模块，共循环了 $m-3$ 次，而其中的lasso训练的时间复杂度为 $O(d^2n+d^3)$，其中 d 为特征数，预测的时间复杂度为 $O(d)$，因此此模块的时间复杂度为 $O((m-3)(d^2n+d^3+d))$，步骤9~11使用lasso求权重的时间复杂度为 $O(d^2n+d^3)$，总共运行了 $2m$ 次，因此时间复杂度为 $O(2m(d^2n+d^3))$，算法步骤12~16是特征选择模块，本文使用的基特征选择器为快速迁移模型，涉及到矩阵求逆，训练时间复杂度为 $O((2mn+n+1)^3)$，共循环运行了 $2m$ 次，因此时间复杂度为 $O(2m(2m+n+1)^3)$，总体的时间复杂度可以表示为 $O(mn^3+(m-3)(d^2n+d^3+d)+2m(d^2n+d^3)+2m(2m+n+1)^3)$。化简此时间复杂度，需要比较 n 和 d 的相对大小，本文和EFSBCV[11]中 n 取值为200，d 的值为1 000，m 取值为10，可见总体的时间复杂度可以化简为 $O(n^3+d^3+(2m+n+1)^3)$。同理EFSBCV[11]的时间复杂度也为 $O(n^3+d^3)$。由此可以看出引入了

快速迁移模型后的特征选择算法的复杂度是有提升的，但性能也相应得到了提升。

3 实验结果和分析

本章把本文提出的算法和EFSBCV算法进行比较，分为模拟数据集和真实数据集。所有实验的实验环境为Win11，电脑CPU型号为i7-12700，2.10 GHz，内存为16 GB，python版本为3.9。

3.1 评估标准

为了公平起见，本文采用与文献[11]相同的评估标准，参考二分类标准的评估标准，将特征是否属于 $H_{0.5,n/2}$ 等价于是否属于正样本，将是否属于 $L_{0.5,n/2}$ 等价于是否属于负样本。本文算法最后筛选出的特征子集可以用如下的矩阵来评判：

$$\boldsymbol{E}=\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}$$

其中

$$TP=\sum_{f=1}^{d}\mathbb{I}(X_f\in H_{0.5,n/2},X_f\in\hat{S}_n)$$

$$FN=\sum_{f=1}^{d}\mathbb{I}(X_f\in H_{0.5,n/2},X_f\notin\hat{S}_n)$$

$$FP=\sum_{f=1}^{d}\mathbb{I}(X_f\in L_{0.5,n/2},X_f\in\hat{S}_n)$$

$$TN=\sum_{f=1}^{d}\mathbb{I}(X_f\in L_{0.5,n/2},X_f\notin\hat{S}_n)$$

式中，TP（TN）代表真阳（真阴）样本个数，FP（FN）代表伪阳（伪阴）样本个数，TP 和 TN 的值越大，FP 和 FN 的值越小，说明被评价对象的性能越好。因为 $TP+FN$（$FP+TN$）的值固定，所以只需要比较 TP 和 FP（或者 FN 和 TN）的值。

3.2 模拟实验

本节采用文献[10]的参数设置来生成模拟数据，对EFSFT和EFSBCV进行比较，模拟实验的平台和真实数据集实验一致，均在python3.9下完成。两者区别在于数据集不同，模拟实验的数据集是模拟生成的，而真实数据集实验的数据集是在UCI下载的，模拟实验的模拟数据参数设置可以参考文献[10]。具体地，关于线性回归的部分，实验设置为首先设置独立同分布的向量 $\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_n$，$\boldsymbol{X}_i\sim N_d(0,\boldsymbol{\Sigma})$，其中 $\boldsymbol{\Sigma}$ 是一个Toeplitz协方差矩阵，每一项为 $\boldsymbol{\Sigma}_{ij}=\rho^{||i-j|-d/2|-d/2}$，其中 ρ 的取值范围为 $[0,1)$；误差项服从 $\varepsilon\sim N_n(0,\sigma^2 I)$，方差 σ^2 是为了实现信噪比（signal-to-noise ratio，SNR）的不同值；再设置 p 维的系数矩阵 $\boldsymbol{\beta}$，设定 $\boldsymbol{\beta}$ 有远远小于 d 的 s 个非0元素，其中 $s/2$ 均匀分布在 $[-1,-0.5]$ 上，剩下 $s/2$ 均匀分布在 $[0.5,1.0]$，而这些非0元素的索引 S 服从四舍五入后的几何级数，第一项为1，第 $s+1$ 项为 $d+1$。非零元素的值随机分配给 S 中的每个索引，对于每个特定的模拟设置，此选择也会相应固定。最后就可以确定相应的 $Y=X\boldsymbol{\beta}+\varepsilon$。对于 ρ、n、d、s 和SNR的不同取值可以获得不同的模拟实验设置，本实验只选取了其中一部分来做说明展示。为了使式(2)各项遵循量纲一致的原则，对于实验中的参数，γ 和 η 在 $\{10,15,20,25,30,35,40\}$ 中进行选取。

表1是EFSBCV与EFSFT在四种不同数据配置下1 000次重复实验的特征选择率比较，n 和 d 的取值分别为200和1 000。其中 ρ_f 是式(1)的简写，代表真实特征选择概率，粗体的数据表示选择概率大于 ρ_f。ρ_f 的取值范围从0.3到0.9，意味着有些重要特征并不属于 $H_{0.5,n/2}$。从实验结果可以看出，对于 $H_{0.5,n/2}$ 中的特征，EFSFT与EFSBCV计算出的选择概率相比较虽没有什么规律，但是 $H_{0.5,n/2}$ 中绝大部分特征在EFSFT算法中的选择概率同样满足大于 ρ_f，这和EFSBCV中提出的结论一致；而对于 $L_{0.5,n/2}$ 中绝大部分特征，EFSFT计算出的选择概率不仅小于 ρ_f，而且也小于EFSBCV算法计算出的选择概率，比如对于 $(SNR,s,\rho)=(1.0,4,0.75)$中的特征 X_{263}，真实选择概率为0.021，EFSBCV的选择概率为0.007，而本文算法选择概率为0。当然也有极个别的特征不满足这个条件，比如 $(SNR,s,\rho)=(0.5,8,0.50)$中的特征 X_1，真实选择概率为0.296，属于 $L_{0.5,n/2}$，但是本文算法计算后的选择概率为0.95。造成这种现象的原因可能是以下几种：数据集的随机性、特征之间的强相关性、比较低的SNR。

表2对比了EFSFT与EFSBCV在20种不同数据配置下1 000次重复实验的 TP 与 FP，其中 n 和 d 的取值分别为200和1 000。粗体表示两种算法的最优值，从最优值的数量来看，EFSBCV算法的 TP 最优值数量更多，而EFSFT算法的 FP 最优值数量更多。从 TP 的角度来分析，EFSBCV的 TP 值略优于EFSFT的 TP 值，但是整体相差不大；而从 FP 的角度来分析，EFSFT的 FP 值略优于EFSBCV的 FP 值。本文进一步从20种不同数据配置下的平均值来分析，EFSFT的 TP 平均值为3.102，FP 的平均值为0.856，

表1 EFSBCV与EFSFT特征选择概率对比

Table 1 Comparison of feature selection probabilities between EFSBCV and EFSFT

Algorithm	$(SNR,s,\rho)=(1.0,4,0.75)$ X_f									
	X_1	X_6	X_{32}	X_{178}	X_{165}	X_{169}	X_{263}	X_{405}	X_{587}	X_{731}
ρ_f	0.864	0.863	0.902	0.904	0.021	0.021	0.021	0.020	0.020	0.022
EFSBCV	**0.997**	0.216	**0.975**	**0.986**	0.002	0.002	0.007	0.006	0.001	0
EFSFT	**0.873**	**0.893**	0.534	0.021	0.001	0.008	0	0.002	0	0.001
Algorithm	$(SNR,s,\rho)=(0.5,4,0)$ X_f									
	X_1	X_6	X_{32}	X_{178}	X_{95}	X_{184}	X_{205}	X_{359}	X_{368}	X_{580}
ρ_f	0.692	0.695	0.698	0.698	0.016	0.016	0.016	0.017	0.017	0.016
EFSBCV	0.576	**0.896**	0.689	0.365	0.001	0.046	0.026	0.031	0.017	0.001
EFSFT	**0.961**	0.544	**0.713**	0.009	0	0	0	0	0	0
Algorithm	$(SNR,s,\rho)=(1.0,8,0.75)$ X_f									
	X_1	X_3	X_6	X_{14}	X_{32}	X_{76}	X_{178}	X_{423}	X_{149}	X_{268}
ρ_f	0.444	0.364	0.498	0.603	0.616	0.615	0.618	0.619	0.022	0.022
EFSBCV	0.195	0	**0.770**	**0.974**	0.550	0.564	**0.873**	0.028	0.003	0.002
EFSFT	0.440	0.002	0	**0.916**	**0.659**	0.013	**0.731**	0.342	0.001	0.003
Algorithm	$(SNR,s,\rho)=(0.5,8,0.50)$ X_f									
	X_1	X_3	X_6	X_{14}	X_{32}	X_{76}	X_{178}	X_{423}	X_{60}	X_{62}
ρ_f	0.296	0.284	0.308	0.325	0.327	0.324	0.325	0.325	0.012	0.012
EFSBCV	**0.310**	**0.882**	0.004	**0.906**	**0.424**	**0.471**	0.283	0.317	0.010	0.003
EFSFT	**0.950**	0.267	0.078	**0.387**	0.008	0	0.002	**0.524**	0.003	0.001

表2 EFSBCV与EFSFT的*TP*与*FP*对比

Table 2 Comparison of *TP* and *FP* between EFSBCV and EFSFT

Algorithm	(SNR,s,ρ)=(0.5,4,0)		(SNR,s,ρ)=(0.5,4,0.50)		(SNR,s,ρ)=(0.5,4,0.75)		(SNR,s,ρ)=(1.0,4,0)		(SNR,s,ρ)=(1.0,4,0.50)	
	TP	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*
EFSBCV	**2.964**	0.757	2.681	0.892	**2.009**	0.211	**3.271**	1.596	**3.237**	1.021
EFSFT	2.824	**0.032**	**3.000**	**0.104**	1.996	**0.098**	2.504	**0.004**	2.692	**0.008**
Algorithm	(SNR,s,ρ)=(1.0,4,0.75)		(SNR,s,ρ)=(1.0,4,0.90)		(SNR,s,ρ)=(2.0,4,0.50)		(SNR,s,ρ)=(2.0,4,0.75)		(SNR,s,ρ)=(2.0,4,0.90)	
	TP	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*
EFSBCV	**3.001**	1.411	**2.562**	2.571	3.578	1.295	**3.540**	2.250	**3.020**	2.482
EFSFT	2.758	**0.820**	2.018	**0.014**	**3.942**	**0.318**	3.438	**2.006**	2.322	**1.470**
Algorithm	(SNR,s,ρ)=(0.5,8,0)		(SNR,s,ρ)=(0.5,8,0.50)		(SNR,s,ρ)=(0.5,8,0.75)		(SNR,s,ρ)=(1.0,8,0)		(SNR,s,ρ)=(1.0,8,0.50)	
	TP	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*
EFSBCV	2.023	1.562	1.768	1.761	**2.081**	1.825	**5.075**	0.502	**4.012**	1.582
EFSFT	**2.968**	**0.028**	**1.880**	**0.760**	1.732	**0.960**	4.800	**0.014**	3.080	**0.420**
Algorithm	(SNR,s,ρ)=(1.0,8,0.75)		(SNR,s,ρ)=(1.0,8,0.90)		(SNR,s,ρ)=(2.0,8,0.50)		(SNR,s,ρ)=(2.0,8,0.75)		(SNR,s,ρ)=(2.0,8,0.90)	
	TP	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*	*TP*	*FP*
EFSBCV	**4.220**	2.263	**3.062**	**1.254**	**6.483**	2.054	**5.622**	**3.045**	**4.992**	3.859
EFSFT	3.810	**1.526**	2.786	2.946	6.246	**0.010**	5.442	3.062	4.914	**3.374**

而EFSBCV的*TP*平均值为3.461，*FP*的平均值为1.71，EFSFT的*TP*平均值比EFSBCV降低了约10.4%，而*FP*平均值却低了约49.9%。因此该实验的实验结果进一步印证了上个实验的结论，在保持$H_{0.5,n/2}$中的特征选择概率相差不大的情况下，进一步降低了$L_{0.5,n/2}$中特征的选择概率，该结论反映到*TP*和*FP*值上就是，在*TP*值相差不大的情况下，进一步降低了*FP*值。

3.3 真实数据集实验

本文从UCI数据集下载了5组数据集，数据集的

大小如表3所示，样本数从1 030到43 824，特征数从5到23。其中数据集Airfoil是NASA记录的关于飞机机翼的一些数据，对该数据集本文只进行了标准化操作；Bias数据集[23]记录了韩国首尔夏季气温，它有两个输出，分别是次日的最高温和最低温，对该数据集的操作包括删除“station”和“date”两个特征，进行标准化处理；PRSA数据集[24]记录了北京在2010年1月1日至2014年12月31日的PM2.5值，数据中包含空值的样本直接进行了删除，并去掉了第一个特征“NO”，这个特征对数据预测是无效的，对“cbwd”风向这个特征，进行了编码，将字符串转为了数字，对“PM2.5”这个特征进行了取对数操作，使其更符合正态分布；Wine数据集记录了红葡萄酒和白葡萄酒的样品质量，本文只选用了白葡萄酒，对数据集进行的操作和文献[11]一致；Concrete数据集记录了混凝土的抗压强度，对该数据集只进行了标准化处理；Fish toxicity是包含908种化学物质的6个属性（分子描述符）值的数据集，用于预测对鱼Pimephales promelas（胖头鲦鱼）的定量急性水生毒性，只对该数据集进行了标准化处理；Seoul bike数据集用于预测首尔自行车管理系统中每小时租用的公共自行车计数，将该数据集的“seasons”“Functioning Day”和“Holiday”3个特征进行编码，将特征“Date”设为索引，并对数据集进行了标准化处理。

表3 数据集信息

Table 3 Dataset information

数据集	样本数	特征数
Airfoil	1 503	5
Bias	7 750	23
PRSA	43 824	12
Wine	4 898	11
Concrete	1 030	8
Seoul bike	8 760	14
Fish toxicity	908	7

本文选择的基特征选择器是快速迁移模型。为了使用这些数据集来进行特征选择的研究，本文参照文献[11]对每个数据集进行了扩展，使每个数据集扩展为1 000个特征，其中扩展出来的伪特征符合标准正态分布。每次实验随机从样本中选取200个样本执行特征选择算法，每个数据集都会执行此过程1 000次，最后的值计算平均值。而在使用快速迁移模型进行重新拟合后，本文按照模型的权重的绝对值大小排序，参照过滤法（filter）的思想，预设选取个数进行选取，预设值一般取为数据集原本特征的个数。而真实数据实验的两个超参数 γ、η，与模拟实验参数选择策略一致。

真实数据的实验结果如表4所示，从实验结果可以看出真实数据集实验印证了模拟实验所得出的结论，EFSFT算法在 *TP* 平均值相差不大的情况下，进一步降低了 *FP* 值。2.3节对EFSFT算法的时间复杂度进行了分析，理论上得出的结论是EFSFT算法的时间复杂度比EFSBCV算法高，而表4展示了每个算法在各个数据集上的CPU平均运行时间，单位为s，CPU平均运行时间指的是1 000轮的CPU平均运行时间。由CPU平均运行时间可以看出，EFSFT算法的实际运行时间比EFSBCV算法长，进一步验证了2.3节的时间复杂度分析。综合模拟实验和真实数据集实验来说，本文提出的EFSFT算法与EFSBCV算法相比，同样满足提高了重要特征的入选概率，降低噪声特征的入选概率。虽然在 *TP* 值和时间复杂度

表4 EFSBCV与EFSFT在真实数据集下的实验数据

Table 4 Experimental data of EFSBCV and EFSFT under real datasets

数据集	EFSBCV			EFSFT		
	TP	*FP*	CPU时间/s	*TP*	*FP*	CPU时间/s
Airfoil	4.001	1.214	1.042	2.764	0.006	12.949
Bias（最高温）	7.996	3.150	2.892	7.826	1.928	14.693
Bias（最低温）	4.000	0.809	2.765	4.074	0.252	14.157
PRSA	3.056	1.276	7.162	2.942	0.050	18.680
Wine	5.000	1.428	2.998	5.668	1.374	14.086
Concrete	4.068	1.548	1.122	4.006	0.100	13.022
Seoul bike	4.007	1.555	1.940	4.088	0.962	13.706
Fish toxicity	3.612	0.289	0.986	3.266	0	12.825
平均值	4.468	1.409	2.613	4.329	0.584	14.265

表5 LSSVM与EFSFT在真实数据集下的实验数据

Table 5 Experimental data of LSSVM and EFSFT under real datasets

数据集	LSSVM			EFSFT		
	TP	*FP*	CPU时间/s	*TP*	*FP*	CPU时间/s
Airfoil	2.624	0.025	2.767	2.764	0.006	12.949
Bias(最高温)	7.523	1.531	3.333	7.826	1.928	14.693
Bias(最低温)	3.847	0.234	3.265	4.074	0.252	14.157
PRSA	2.886	0.165	7.557	2.942	0.050	18.680
Wine	5.000	1.745	2.397	5.668	1.374	14.086
Concrete	3.868	0.001	2.378	4.006	0.100	13.022
Seoul bike	3.917	0.679	3.169	4.088	0.962	13.706
Fish toxicity	3.124	0.168	2.188	3.266	0	12.825
平均值	4.099	0.569	3.382	4.329	0.584	14.265

方面稍显劣势,但进一步降低了 *FP* 的值,而 *FP* 值代表属于低入选概率特征集合中的特征也就是噪声特征入选了最终特征子集的数量, *FP* 值的降低表明EFSFT降低了噪声特征的入选概率,表明EFSFT在去除噪声特征方面更具优势。

3.4 与LSSVM算法对比实验

上一组对比实验中,本文已经验证了与EFSBCV算法相比,EFSFT算法进一步降低了 *FP* 值,证明在去除噪声特征方面更具优势。由于EFSFT算法的快速迁移模型是LSSVM模型改进的,为了验证本文所提算法的优势,本节将EFSFT算法与基特征选择器为LSSVM的EFSBCV算法进行比较,所采用的数据集及相关处理均与上节相同,实验结果如表5所示。从实验结果来看,与LSSVM相比,EFSFT算法的 *FP* 值在相差不大的情况下, *TP* 值提高了5%,而从CPU平均运行时间可以看出,EFSFT算法的实际运行时间比LSSVM略长。综合来看,与LSSVM相比,EFSFT算法在筛选重要特征方面性能更好。

3.5 实验总结

为了验证本文提出的集成特征选择方法的正确性以及优势,本文共进行了四组对比实验,前两组为模拟实验,后两组为真实数据集实验。前两组对比实验在不同的数据配置上从特征选择概率和 *TP*(*FP*)值角度,证明了EFSFT算法在保持EFSBCV算法优势情况下,进一步降低了 *FP* 的值。然后又在真实数据集上验证了EFSFT相比较于EFSBCV进一步降低了 *FP* 的值,证明EFSFT在去除噪声特征方面更具优势。同时分析了EFSFT算法的时间复杂度高于EFSBCV,并在实验阶段记录了实验的CPU运行时间,同样EFSFT算法略高于EFSBCV算法。由于EFSFT算法的模型是LSSVM模型的改进,最后一组对比实验是和LSSVM做对比,证明了EFSFT在筛选重要特征方面具有优势。

4 结束语

本文针对EFSBCV的线性模型只有误差项而不包含偏置项的问题,而且EFSBCV直接将 $2m$ 个特征子集直接求均值得出最终特征子集,并没有考虑每个特征子集的重要程度,提出了一种集成特征选择的方法EFSFT。将lasso训练后的模型经过线性组合作为源知识引入到LSSVM模型,生成了快速迁移模型,将快速迁移模型作为基特征选择器,经过重新拟合后,产生新的特征权重,实现将EFSBCV的线性模型的误差项改为偏置项,提高了特征选择方法的性能。本文首先简单介绍了EFSBCV,又介绍将快速迁移模型作为基特征选择器的EFSFT,然后通过实验证明EFSFT在去除噪声特征方面更具优势。

鉴于本文只是介绍了回归部分,对于今后的工作,可以推广到分类。甚至更深一步,对特征选择过程改动,把前一步的特征选择结果作为参考,来进行下一步的特征选择。

参考文献:

[1] CEKIK R, UYSAL A K. A novel filter feature selection method using rough set for short text data[J]. Expert Systems with Applications, 2020, 160: 113691.

[2] CHEN C W, TSAI Y H, CHANG F R, et al. Ensemble feature selection in medical datasets: combining filter, wrapper, and embedded feature selection results[J]. Expert Systems, 2020, 37(5): e12553.

[3] DROTÁR P, GAZDA M, VOKOROKOS L. Ensemble feature selection using election methods and ranker clustering

[J]. Information Sciences, 2019, 480: 365-380.
[4] 杨春, 郭健, 张磊, 等. 采用卡方检验的模糊自适应无迹卡尔曼滤波组合导航算法[J]. 控制与决策, 2018, 33(1): 81-87.
YANG C, GUO J, ZHANG L, et al. Fuzzy adaptive unscented Kalman filter integrated navigation algorithm using Chi-square test[J]. Control and Decision, 2018, 33(1): 81-87.
[5] 叶明全, 高凌云, 万春圆, 等. 基于对称不确定性和SVM递归特征消除的信息基因选择方法[J]. 模式识别与人工智能, 2017, 30(5): 429-438.
YE M Q, GAO L Y, WAN C Y, et al. Informative gene selection method based on symmetric uncertainty and SVM recursive feature elimination[J]. Pattern Recognition and Artificial Intelligence, 2017, 30(5): 429-438.
[6] MEINSHAUSEN N, BÜHLMANN P. Stability selection[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2010, 72(4): 417-473.
[7] GONZÁLEZ S, GARCÍA S, DEL SER J, et al. A practical tutorial on bagging and boosting based ensembles for machine learning: algorithms, software tools, performance study, practical perspectives and opportunities[J]. Information Fusion, 2020, 64: 205-237.
[8] 梁令羽, 孙铭堃, 何为, 等. Bagging-SVM集成分类器估计头部姿态方法[J]. 计算机科学与探索, 2019, 13(11): 1935-1944.
LIANG L Y, SUN M K, HE W, et al. Head pose estimation method of bagging-SVM integrated classifier[J]. Journal of Frontiers of Computer Science and Technology, 2019, 13 (11): 1935-1944.
[9] PALEOLOGO G, ELISSEEFF A, ANTONINI G. Subagging for credit scoring models[J]. European Journal of Operational Research, 2010, 201(2): 490-499.
[10] SHAH R D, SAMWORTH R J. Variable selection with error control: another look at stability selection[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2013, 75(1): 55-80.
[11] YANG X, WANG Y, WANG R, et al. Ensemble feature selection with block-regularized $m\times2$ cross-validation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 34(9): 6628-6641.
[12] BACH F R. Bolasso: model consistent lasso estimation through the bootstrap[C]//Proceedings of the 25th International Conference, Helsinki, Jun 5-9, 2008: 33-40.
[13] ELGHAZEL H, AUSSEM A. Unsupervised feature selection with ensemble learning[J]. Machine Learning, 2015, 98 (1/2): 157-180.
[14] HALLAJIAN B, MOTAMENI H, AKBARI E. Ensemble feature selection using distance-based supervised and unsupervised methods in binary classification[J]. Expert Systems with Applications, 2022, 200: 116794.
[15] LIU K, YANG X, YU H, et al. Rough set based semi-supervised feature selection via ensemble selector[J]. Knowledge-Based Systems, 2019, 165: 282-296.
[16] KWOK T Y, YEUNG D Y. Use of bias term in projection pursuit learning improves approximation and convergence properties[J]. IEEE Transactions on Neural Networks, 1996, 7(5): 1168-1183.
[17] RAWAT M, GHANNOUCHI F M, RAWAT K. Three-layered biased memory polynomial for dynamic modeling and predistortion of transmitters with memory[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2012, 60 (3): 768-777.
[18] SUYKENS J A K, VANDEWALLE J. Least squares support vector machine classifiers[J]. Neural Processing Letters, 1999, 9(3): 293-300.
[19] BERRAR D. Cross-validation[M]//Encyclopedia of Bioinformatics and Computational Biology. Amsterdam: Elsevier, 2019: 542-545.
[20] WANG R, WANG Y, LI J, et al. Block-regularized $m\times2$ cross-validated estimator of the generalization error[J]. Neural Computation, 2017, 29(2): 519-554.
[21] 李赟波, 王士同. 多源域分布下优化权重的迁移学习Boosting方法[J]. 计算机科学与探索, 2023, 17(6): 1441-1452.
LI Y B, WANG S T. Transfer learning Boosting for weight optimization under multi-source domain distribution[J]. Journal of Frontiers of Computer Science and Technology, 2023, 17(6): 1441-1452.
[22] 徐光生, 王士同. 基于潜在的低秩约束的不完整模态迁移学习[J]. 计算机科学与探索, 2022, 16(12): 2775-2787.
XU G S, WANG S T. Incomplete modality transfer learning via latent low-rank constraint[J]. Journal of Frontiers of Computer Science and Technology, 2022, 16(12): 2775-2787.
[23] CHO D, YOO C, IM J, et al. Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas[J]. Earth and Space Science, 2020, 7(4): e2019EA000740.
[24] LIANG X, ZOU T, GUO B, et al. Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating [J]. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2015, 471(2182): 20150257.

宁保斌(1996—),男,山东德州人,硕士研究生,主要研究方向为模式识别、人工智能。
NING Baobin, born in 1996, M.S. candidate. His research interests include pattern recognition and artificial intelligence.

王士同(1964—),男,江苏扬州人,教授,博士生导师,CCF会员,主要研究方向为模式识别、人工智能。
WANG Shitong, born in 1964, professor, Ph.D. supervisor, CCF member. His research interests include pattern recognition and artificial intelligence.