

引用格式:郭嘉. 面向非均衡数据的二进制排队搜索特征选择机制[J]. 微电子学与计算机, 2021, 38(8): 45-52. [GUO J. Feature selection mechanism based on binary queue search for unbalanced data[J]. Microelectronics&Computer, 2021, 38(8): 45-52.]

面向非均衡数据的二进制排队搜索特征选择机制

郭 嘉

(郑州西亚斯学院 电子信息工程学院, 河南 郑州 451150)

摘 要: 非均衡数据(分类不均匀分布)和冗余特征的出现极大增加了数据准确分类的难度. 以最优化学习算法的预测准确率为目标, 结合合成少数过采样技术 SMOTE, 设计了一种针对非均衡数据的二进制排队搜索方法的包装式特征选择算法 BQSA. 利用 PROMISE 知识库中十四种软件故障预测数据集进行实验分析. 测试了数据集过采样比例的影响, 证实合成少数过采样对高度非均衡数据的分类预测具有正面影响, 并得到了最佳过采样率; 比较了 BQSA 与同类算法的性能, 证实结合合成少数过采样技术的 BQSA 算法拥有更优的预测准确性, 在分类敏感度、专一性以及曲线下面积 AUC 等指标上表现更佳.

关键词: 特征选择; 非均衡数据; 排队搜索算法; 合成少数过采样; 学习算法

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2021)08-0045-08

Feature selection mechanism based on binary queue search for unbalanced data

GUO Jia

(School of Electronic Information Engineering, Zhengzhou Sisa University, Zhengzhou 451150, Henan, China)

Abstract: The unbalanced data (non-uniform distribution of classes) and the redundant features dramatically increased the difficulty of data accurate classification. Taking the prediction accuracy of the optimal learning algorithm as the goal, combined with the synthetic minority oversampling technology SMOTE, a wrapper feature selection algorithm BQSA was designed for binary queue search method of unbalanced data. Using 14 kinds of software fault prediction in PROMISE knowledge base to conduct experimental analysis of datasets. The influence of the over-sampling ratio of the dataset is tested, and it is proved that the synthesis of a few over-sampling has a positive effect on the classification prediction of highly unbalanced data, and the optimal over-sampling rate is obtained. The performance of BQSA is compared with similar algorithms, and it is proved that the BQSA algorithm combined with synthetic minority oversampling has better prediction accuracy and better performance in classification sensitivity, specificity and AUC of area under the curve.

Key words: feature selection; imbalanced data; queuing search algorithm; synthetic minority oversampling; learning algorithm

1 引言

数据挖掘领域中,分类技术可基于数据集的可用特征将数据点根据预定义标签进行分类^[1].因此,为建构分类模型所使用的特征对构造模型的性能具有重要影响.换言之,若不相关特征或冗余特征在数据集中仍是可用信息,将会误导分类模型,进而降低分类性能.为了以更少的计算时间获取高性能的分类模型,选择最为信息化的特征是必不可少的步骤.特征选择 FS^[2]是数据分类的预处理步骤,其目标是通过提取最信息化特征,丢弃噪声和不相关特征,建构健壮的分类模型.因此,特征选择可考虑为一个优化问题,并重点解决两个子问题:一是最优特征子集生成问题,二是特征子集评估问题.搜索机制可用于生成可用特征子集,而评估技术(学习算法)则用于评估特征子集并指导搜索过程向最优解转移.

特征选择方法主要依据用于建构分类模型的学习算法的依赖性进行区分^[3].若学习算法参与分类过程,则该方法为包装法;若不参与,则为过滤法.过滤法与包装法的区别在于,前者计算效率更高,但所选特征可能不适用于学习算法.包装法中,特征选择基于分类准确性和机器学习算法共同决定,虽然计算时间代价略高,但可以确保分类性能.

搜索最优特征子集是一个困难的优化问题,主要体现在:首先,特征选择是多目标优化问题,需要在满足最高预测质量的同时,最小化特征选择量;其次,拥有大量特征(高维度)的数据集会进一步增加问题的复杂度.通常来说,若一个数据集拥有 N 个特征,则存在的特征子集理论上有 2^N 个,是指数级的.搜索空间呈指数级递增,完成随机化搜索显然不适宜于解决这一问题.因此,通常采用启发式方法解决这一问题,可在相对合理的时间内得到质量较好的求解方案^[4].相关研究中,文献[5]依据词条权重重要性设计基于遗传算法的特征选择算法 GAFS,可以较高效率选择最优特征子集.文献[6]通过改进和声搜索过程设计了自适应特征选择算法,利用音调调整、限制特征域和内存合并率三种动态策略改进传统和声搜索,多维度数据集测试也验证了最优特征子集选择的有效性.粒子群算法是一种计算代价较小且收敛较快的智能算法.文献[7]提出基于粒子群优化的特征选择算法 BPSOFS.利用新的动态惯性权重策略对粒子进化操作进行改进,而数据集测试也证明新算法生成的聚类在准确度和效率上均有所提升.文献[8]结合遗传算法和并行协同进化算

法求解特征选择,先利用遗传算法搜索特征子集,再进一步利用进化协同机制提升搜索效率,得到准确度更高的特征选择解.猫群算法也可用于求解优化问题.文献[9]通过修正猫群算法设计了特征选择算法,结果也证实利用猫群算法后的聚类明显优于仅使用词条频率下的聚类效果.文献[10]设计了基于蚁群算法的无监督概率特征选择算法,算法利用特征相似性降低特征维度,引入矩阵记录特征共存率,通过特征提取概率函数对特征进行排序,并获取相关分值更高的特征子集作为聚类特征基础.文献[11]通过小生境改进了传统粒子群算法,以增强的粒子搜索性能对文本特征选择进行进化求解,获得的文本分类效率得到有效提升.

除了特征子集求解本身,数据集的不均衡性也会降低分类模型的性能.在现实世界中,数据集中各个类别的样本量是极不平衡的,如正类样本数量远大于负类样本数量(比例接近于 100:1),这种不平衡的目标分类的表达同样会降低分类模型的预测性能,此时不均衡数据集的学习算法需要在分布不均匀的数据集中学习到更加有用的信息.

为了解决不均衡数据特征选择问题,提升分类模型准确率,利用合成少数过采样技术 SMOTE,提出一种基于二进制排队搜索的包装式非均衡数据特征选择模型.

2 不均衡数据与过采样模型

数据不均衡问题是数据挖掘与机器学习中的常见问题,当数据集中实例的目标分类出现不均匀分布,使得少数分类拥有比多数分类明显的较低表现时,即所谓数据不均衡问题.此时,分类模型会偏向于占据优势的分类,对少数分类预测不准,即多数实例会被预测为多数分类.然而,在诸如疾病诊断、软件缺陷预测等应用中,恰恰是少数分类对于预测模型拥有更好的重要性.采样机制中,分类均衡表达可以通过减少多数分类样本数量(即欠采样)或增加少数分类样本数量(即过采样)得到.然而,欠采样会带来信息损耗,并降低分类模型性能.为了克服这一问题,本文首先利用一种合成少数过采样方法 SMOTE 作分类均衡表达,基于插值为少数类合成新的样本.

对于每个少数样本 x_i ,SMOTE 利用欧氏距离标识出 k 个最邻近的少数样本邻居 x'_{ij} , $j = 1, 2, \dots, k$.然后,新的合成样本依线性形式加入少数样本及其 j 个所选邻居得到,表示为:

$$x_{\text{new}} = x_i + (x'_i - x_i) \cdot r \quad (1)$$

式中, r 是 $[0, 1]$ 间的随机量; x'_i 是 k 个邻居样本中随机选择的样本 i ; k 值取决于期望的过采样量. 对以上合成操作重复 N 次, 即可合成 N 个新样本. 而对全部少数类样本进行以上过采样操作, 即可得到针对少数类样本的新样本. SMOTE 的本质即为利用插值合成样本, 同时增加了随机性. 一方面是从 k 个近邻中随机选择样本, 另一方面则通过参数 r 增加随机性.

3 基于排队搜索的特征选择

排队搜索算法 QSA 是最新提出的一种元启发式智能算法, 模拟了人类社会排队过程中人的行为及其相互影响. 在人类日常活动中, 如购物等待服务过程中, 客户通常会跟随能力更强的员工获取服务. 基于这一原则, 客户群体代表搜索代理, 员工则指导搜索代理的搜索方向. QSA 算法的全局搜索和局部开发间的均衡由三个阶段构成, 将其分别命名为活动 business1、活动 business2 和活动 business3.

3.1 活动 business1

该阶段中, 所有客户(搜索代理) 分布于三个队列, 命名为 $queue_{1n}, n = 1, 2, 3$. 将适应度最优的三个代理标记为 A_{1n} , 并将其选定为员工(即领导者). 每个队列中的客户数量 q_{1n} 计算方式如式(2), 计算依据是: 相同时间内, 越优秀的员工, 服务客户数量也越多.

$$q_{1n} = N \times \frac{1/T_{1n}}{1/T_{11} + 1/T_{12} + 1/T_{13}}, n = 1, 2, 3 \quad (2)$$

式中, N 表示搜索代理数量; T_{1n} 表示所选员工的适应度值(即服务时间). 则每个队列中客户状态的更新方式利用以下两个原则之一进行:

$$X_i^{\text{new}} = A + F_{11} \quad (3)$$

$$X_i^{\text{new}} = X_i + F_{12} \quad (4)$$

式中, i 表示当前迭代数; X_i^{new} 表示 X_i 的更新状态; A 表示服务客户 i 的员工; F_{11} 和 F_{12} 分别表示服务过程的波动, 分别计算为:

$$F_{11} = \beta \times \alpha \times (E \cdot |A - X_i|) + (e \times A - e \times X_i) \quad (5)$$

$$F_{12} = \beta \times \alpha \times (E \cdot |A - X_i|) \quad (6)$$

式中, α 表示区间 $[-1, 1]$ 内的随机量, 用于控制客户的波动方向; E 表示大小为 D 维的厄兰分布随机矢量; e 表示厄兰分布随机值; $| \cdot |$ 表示绝对值符号; $(\cdot)$ 表示元素乘符号; β 表示控制参数, 用于调整波

动范围, 根据迭代终止条件定义, 计算为:

$$\beta = e^{(\ln(1/t) \times (t/T)^{0.5})} \quad (7)$$

式中, t 表示当前迭代数; T 表示算法的最大迭代数. 由上式可知, β 值将随着迭代增加而减小.

3.2 活动 business2

第二个阶段中, 同样拥有三个队列, 命名为 $queue_{2n}, n = 1, 2, 3$. 部分客户被选择参与本阶段的活动. 初始时, 客户根据其适应度值 f_i 进行降序排列, 每个客户给定一个待处理概率, 计算为:

$$pr_i = \text{rank}(f_i)/N, i = 1, 2, \dots, N \quad (8)$$

式中, $\text{rank}(f_i)$ 表示所有客户降序排列时客户 i 的适应度 f_i 的排序. 由此可知, 较差的搜索代理拥有比适应度更优的代理更大的参与该阶段的活动机会. 为了决定客户 i 是否执行活动 business2, 对于每个代理, 生成一个 $[0, 1]$ 间的随机值, 若该值小于概率 pr_i , 则客户 i 执行活动 business2. 该操作用于增加拥有较大概率值的客户接受活动 business2 的机会. $queue_{2n}$ 中的客户数量以相同方式计算为:

$$q_{2n} = N \times \frac{1/T_{2n}}{1/T_{21} + 1/T_{22} + 1/T_{23}}, n = 1, 2, 3 \quad (9)$$

所选代理根据两种模式进行状态更新, 表示为:

$$X_i^{\text{new}} = \begin{cases} X_i + e \times (X_{r1} - X_{r2}), r < c_v, i = 1, 2, \dots, N_2 \\ X_i + e \times (A - X_{r1}), r \geq c_v, i = 1, 2, \dots, N_2 \end{cases} \quad (10)$$

式中, N_2 表示参与活动 business2 的客户数量; e 表示厄兰分布随机量; X_{r1} 和 X_{r2} 分别表示两个随机选择客户; A 表示每个队列的领导者(员工); c_v 表示混淆度, 用于控制两种更新模式的选择概率, 计算为:

$$c_v = T_{21}/(T_{22} + T_{23}) \quad (11)$$

式中, $T_{2n}(n = 1, 2, 3)$ 表示至目前为止活动 business2 中得到的适应度最优的前三个客户适应度. 根据式(9)可知, 随着 c_v 值的增加, 第一个更新模式将逐步拥有更高的选择概率, 因此, 单个客户的状态将根据其它客户状态更新.

3.3 活动 business3

该阶段的更新模型为:

$$X_i^{\text{new}} = X_{r1,j} + (e \times X_{r2,j} - e \times X_{i,j}), i = 1, 2, \dots, N_3 \quad (12)$$

式中, N_3 表示参与活动 business3 的客户数量; X_{r1} 和 X_{r2} 分别表示两个随机选择的客户. 根据此式可知, 其它客户将对客户 i 的服务过程具有更大的影响.

如图 1 是排队搜索中客户搜索的演变过程.

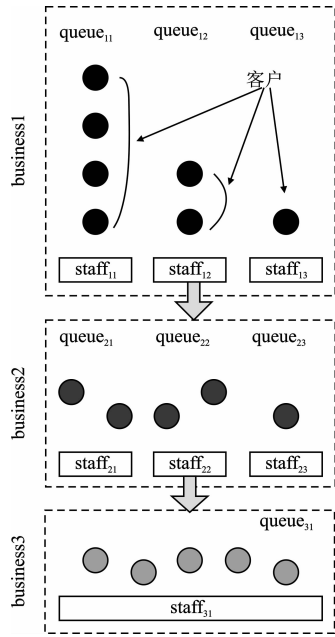


图 1 排队搜索过程

3.4 特征选择解的表示

特征选择可视为一个二进制优化问题,即特征子集需要编码为由 0 和 1 组成的矢量.若特定特征被选择,则相应矢量元素值为 1;否则,则忽略该特征,相应矢量元素为 0.如表 1 所示的特征选择解中,若初始特征量为 10,则此时特征 2、4、5、6、9、10 被选择,特征 1、3、7、8 未被选择.

表 1 特征选择解的表示

X	0	1	0	1	1	1	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---

进一步,由于传统排队搜索算法 QSA 是连续搜索空间内的寻优算法,故需要利用二值化方法将其转换为二进制算法,使其可以求解二进制特征选择问题.本文利用转换函数 TF 将连续搜索空间映射至二进制空间上.算法采用的转换函数为反曲 S 型函数,定义为:

$$T(x_i^j(t)) = 1 / (1 + e^{-x_i^j(t)}) \tag{13}$$

S 型函数曲线如图 2 所示.式中, x_i^j 表示迭代 t 时解 i 上的 j 维值; $T(x_i^j(t))$ 表示 TF 的概率值.相应更新为:

$$x_i^j(t+1) = \begin{cases} 0, & r < T(x_i^j(t+1)) \\ 1, & \text{otherwise} \end{cases} \tag{14}$$

式中, r 表示 $[0,1]$ 内的随机量; $x_i^j(t+1)$ 表示新的二进制输出值.

3.5 适应度函数

数据集分类中的特征选择问题实质是多目标优化问题,算法目标是通过寻找最小数量的特征数量,

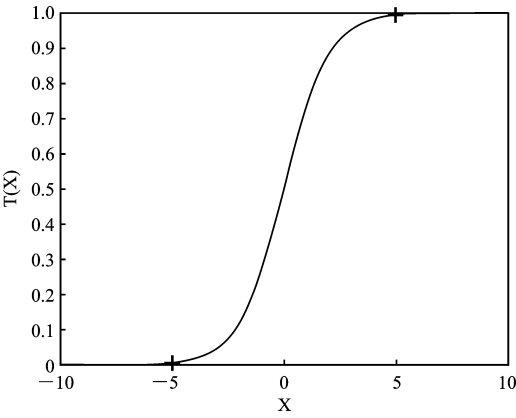


图 2 S 型转换函数

实现特征降维,从而达到分类准确率最大化.因此,为了均衡特征选择量和分类准确率,可将两个相互冲突目标以权重适应度函数形式表示为:

$$fitness = \rho \cdot E(D) + \varphi \cdot |R| / |N| \tag{15}$$

式中, $E(D)$ 为标识特征子集的分类错误率,算法利用 KNN 分类器评估特征子集优劣; $|R|$ 为特征子集所选特征量; $|C|$ 为原始数据集的总特征量; ρ 和 φ 分别表示控制分类准确率和特征降维的常量, $\rho, \varphi \in [0,1]$, 且 $\rho + \varphi = 1$.

4 实验分析

4.1 实验环境搭建

为了测试算法性能,利用表 1 中 PROMISE 知识库的基准软件故障预测数据集进行仿真分析.根据表 1 可知,该数据集是不均衡数据集,正类样本远大于负类样本出现的比例.

表 1 软件故障预测数据集相关属性

数据集	实例数	缺陷实例数	缺陷实例比例(%)
Ant-1.7	745	166	23.3
Camel-1.0	339	13	3.8
Camel-1.2	608	216	35.5
Camel-1.4	872	145	16.6
Camel-1.6	965	188	19.5
Jedit-3.2	272	90	33.1
Jedit-4.0	306	75	24.5
Jedit-4.1	312	79	25.3
Jedit-4.2	367	48	13.1
Jedit-4.3	492	11	2.2
Log4j-1.0	135	34	25.2
Log4j-1.1	109	37	33.9
Log4j-1.2	205	189	92.2
Xalan-2.4	723	110	15.2

仿真实验按以下三个阶段进行. 第一阶段:计算最优的数据过采样比例,如后文表 4. 第二阶段:观察使用合成少数过采样 SMOTE 方法前后,利用二进制排队搜索算法 BQSA 进行特征选择的性能差异,并利用敏感度、专一性和曲线下面积 AUC 三个指标评估 BQSA 算法的性能. 第三阶段:比较利用 BQSA 算法作特征选择后,利用 KNN 分类器得到的分类输出,与其它几种智能群体优化算法结果作对比分析. 实验选择二进制鲸鱼优化算法 BWOA^[12]、二进制引力搜索算法 BGSA^[13]、二进制蝙蝠算法 BBA^[14] 以及二进制蚁狮优化算法 BALO^[15] 作性能对比.

利用 n 折交叉验证方法($n=10$) 在基准数据集上生成训练集和测试集. 该过程中,每个数据集被划分为 10 个部分,9 个用于训练,1 个用于测试. 重复以上过程 k 次,使数据集中的每个实例拥有 $k-1$ 次模型训练机会,1 次验证分类模型机会. 由于启发式算法固有的随机性,每个实验重复 10 次,则单个算法将在每个数据集上评估 $10 \times k$ 次,以此提升算法验算性能的可行性.

实验平台为 Matlab-R2017a, KNN 分类器中 $k=5$. CPU 为 Intel Core i5 2.2 GHz,内存为 4 GB. 为了公平性和一致性,所有智能算法使用相同的常规参数,即最大迭代次数为 100,搜索代理即种群规模为 30. 其它特定参数配置如表 2 所示.

表 2 对比算法实验参数配置

说明	具体配置
适应度函数	$\alpha=0.99, \beta=0.01$
常规参数	最大迭代=100,种群规模=30 个体维度对应特征数量,实验运行次数=10
分类器	KNN 分类器 $k=5$, 10 折交叉验证
BGSA	$G_0=100, \alpha=20$
BBA	$Q_{\min}=0, Q_{\max}=2$, 响度=0.5, r 脉冲率=0.5
BGWO	收敛因子 a 从 2 线性递减至 0

4.2 评估指标

分类器在测试数据集上的性能可以一种特定表即混淆矩阵(误差矩阵)描述. 表 3 是一种针对二进制分类器的混淆矩阵,矩阵中对于一个给定测试数据的实例被分类为正类或负类. 根据混淆矩阵中四个输出(TP、TN、FP、FN)计算分类准确率、误差率、敏感度和专一性. AUC 指标根据前面两个已知指标计算.

表 3 二进制分类的混淆矩阵

	预测正类	预测负类
实际正类	真正正类(TP)	错误负类(FN)
实际负类	错误正类(FP)	真正负类(TN)

敏感度(真正正类率):被预测为正类的正类百分比. 定义为:

$$\text{敏感度} = TP / (TP + FN)$$

专一性(真正负类率):被预测为负类的负类百分比. 定义为:

$$\text{专一性} = TN / (TN + FP)$$

AUC:敏感度与专一性的均衡指标,计算为:

$$AUC = (\text{敏感度} + \text{专一性}) / 2$$

对于非均衡数据集,算法期望在少数分类中拥有较高的敏感度,而在多数分类中拥有较高的专一性. 因此, AUC 在非均衡数据中是一个比准确率更适合的预测质量度量指标.

4.3 实验结果分析

1)过采样率对 AUC 指标的影响

为了得到最佳过采样比例,本节首先配置不同过采样比例进行扩展实验,过采样比例设置为 100%、200%、300%、400%,得到的过采样比例最优值在后续实验中利用. 表 4 记录了不同过采样比例下的 AUC 指标. 由结果可知,比较原始采样数据,所有过采样情况均在 AUC 指标上得到了性能提升. 同时,较低过采样比例(100%、200%)的影响要弱于高过采样比例(300%、400%)的影响,这是由于在高度不均衡数据集内,大量少数分类实例需要被过采样. 同时,过采样比例的进一步增加(高于 300%)对预测质量有着负面影响. 通过表中最后一行 F 测试的结果可以看出,少数分类的过采样比例增加至 300%后达到了最优 F 测试排序,紧随其后是 400%、200%、100%和 0%(原始数据). 因此,后续实验中,算法将利用 300%的过采样比例进行其它实验.

2)利用合成少数过采样 SMOTE 前后性能比较

本节利用 BQSA 算法进行特征选择,然后比较在原始数据集(未作过采样),以及利用 300%过采样比例(同步少数过采样方法)后算法的性能表现. 表 5 比例了 BQSA 算法在利用过采样前后的敏感度、专一性和 AUC 三个指标情况. 对于敏感度值,其代表的是发现兴趣分类(缺陷分类)的性能度量. 可以看到,利用合成少数过采样 SMOTE 可以在所有数据集上显著提升 BQSA 算法的性能. 此外,在

专一性上有明显的降低,该指标代表的是发现正常分类的性能度量.为了均衡两个指标上的冲突行为,进一步检测 AUC 指标值.可知,利用 SMOTE 在所有数据集上,BQSA 算法仍然实现了较好的性能提升,这是由于对于非均衡数据集,原始数据集会出现向着多数分类偏向的不足.

表 4 不同过采样比例下利用 SMOTE 前后 BQSA 的平均 AUC

数据集	原始数据	合成少数过采样 SMOTE			
		100%	200%	300%	400%
Ant-1.7	0.699	0.804	0.815	0.835	0.819
Camel-1.0	0.498	0.666	0.727	0.763	0.814
Camel-1.2	0.590	0.670	0.687	0.665	0.663
Camel-1.4	0.592	0.719	0.777	0.786	0.798
Camel-1.6	0.569	0.702	0.757	0.770	0.764
Jedit-3.2	0.740	0.795	0.801	0.809	0.799
Jedit-4.0	0.703	0.765	0.775	0.795	0.785
Jedit-4.1	0.723	0.791	0.822	0.828	0.818
Jedit-4.2	0.667	0.813	0.826	0.851	0.857
Jedit-4.3	0.508	0.563	0.583	0.787	0.788
Log4j-1.0	0.788	0.827	0.873	0.859	0.862
Log4j-1.1	0.804	0.819	0.845	0.849	0.835
Log4j-1.2	0.683	0.761	0.867	0.871	0.880
Xalan-2.4	0.614	0.751	0.799	0.828	0.822
F-测试排序	5.00	3.86	2.50	1.64	2.00

表 5 利用 SMOTE 前后 BQSA 算法的性能表现

数据集	敏感度指标		专一性指标		AUC 指标	
	原始数据	SMOTE	原始数据	SMOTE	原始数据	SMOTE
Ant-1.7	0.501 8	0.921 8	0.896 5	0.747 6	0.699 1	0.834 7
Camel-1.0	0.000 0	0.576 9	0.996 0	0.948 2	0.498 0	0.762 5
Camel-1.2	0.374 5	0.887 7	0.806 1	0.441 6	0.590 3	0.664 7
Camel-1.4	0.232 4	0.854 8	0.951 2	0.718 0	0.591 8	0.786 4
Camel-1.6	0.204 3	0.856 3	0.933 2	0.684 2	0.568 7	0.770 2
Jedit-3.2	0.653 3	0.922 8	0.826 4	0.696 2	0.739 9	0.809 5
Jedit-4.0	0.510 7	0.879 3	0.896 1	0.710 0	0.703 4	0.794 6
Jedit-4.1	0.549 4	0.901 6	0.895 7	0.754 9	0.722 5	0.828 3
Jedit-4.2	0.389 6	0.878 6	0.943 6	0.823 2	0.666 6	0.850 9
Jedit-4.3	0.018 2	0.620 5	0.997 7	0.953 0	0.507 9	0.786 7
Log4j-1.0	0.626 5	0.912 5	0.949 5	0.805 9	0.788 0	0.859 2
Log4j-1.1	0.686 5	0.925 7	0.922 2	0.772 2	0.804 4	0.848 9
Log4j-1.2	0.985 2	0.886 2	0.381 3	0.856 3	0.683 2	0.871 2
Xalan-2.4	0.271 8	0.867 5	0.955 3	0.787 9	0.613 6	0.827 7

3)BQSA 与其它算法的对比

本节观察 BQSA 与对比算法的性能优势.为了公平性,所有实验环境相同,参数配置相同.表 6 是六种特征选择算法的平均 AUC 指标结果,包括 BQSA、WOA、BGSA、BBA、BALO,KNN * 表示在所有过采样数据集上未作特征选择的 KNN 算法.可以看到,BQSA 在 65%数据集中得到了所有算法中的最优结果,其 F 测试排序第 1.WOA 在 28%数据集中得到最优 AUC,排序第 2.与未作特征选择的 KNN 算法对比可知,特征选择作为数据分类的预处理步骤,对学习算法性能具有重要影响.根据 F 测试结果,BBA 和 KNN * 是所有算法中表现最差的.

表 6 利用 SMOTE 下不同特征选择算法的 AUC 表现

数据集	KNN *	BQSA	BWOA	BGSA	BBA	BALO
Ant-1.7	0.835	0.835	0.836	0.826	0.818	0.834
Camel-1.0	0.684	0.763	0.761	0.740	0.697	0.758
Camel-1.2	0.658	0.665	0.688	0.660	0.636	0.666
Camel-1.4	0.789	0.786	0.786	0.780	0.772	0.783
Camel-1.6	0.739	0.770	0.765	0.747	0.735	0.764
Jedit-3.2	0.751	0.809	0.798	0.794	0.767	0.802
Jedit-4.0	0.760	0.795	0.793	0.781	0.769	0.788
Jedit-4.1	0.793	0.828	0.826	0.806	0.768	0.798
Jedit-4.2	0.786	0.851	0.852	0.835	0.808	0.843
Jedit-4.3	0.673	0.787	0.782	0.748	0.690	0.780
Log4j-1.0	0.752	0.859	0.854	0.823	0.810	0.849
Log4j-1.1	0.727	0.849	0.856	0.825	0.805	0.852
Log4j-1.2	0.681	0.871	0.867	0.825	0.768	0.840
Xalan-2.4	0.821	0.828	0.824	0.824	0.809	0.818
F-测试排序	5.04	1.57	1.86	3.96	5.43	3.14

图 3 是特征选择算法在所选四个数据集上的收敛曲线情况.显然,BQSA 是所有算法表现最好的,在所有数据集均拥有最快的收敛速度和最小的适应度值(适应度值越小,性能越优),说明算法寻优精度高,收敛速度快.其它算法中,仅有 WOA 在 Xalan-2.4 数据集上拥有不错的性能.此外,BBA 和 BGSA 在所有数据集上均表现出早熟收敛的不足.

综合以上结论,BQSA 算法拥有比其它对比算法更佳的性能表现,原因在于其多领导者的优化机制,整个群体划分为三个群组,每个群组可跟随三个最优领导者中的一个进行移动,这使得搜索代理可以同时搜索多个区域,避免陷入局部最优解.此外,

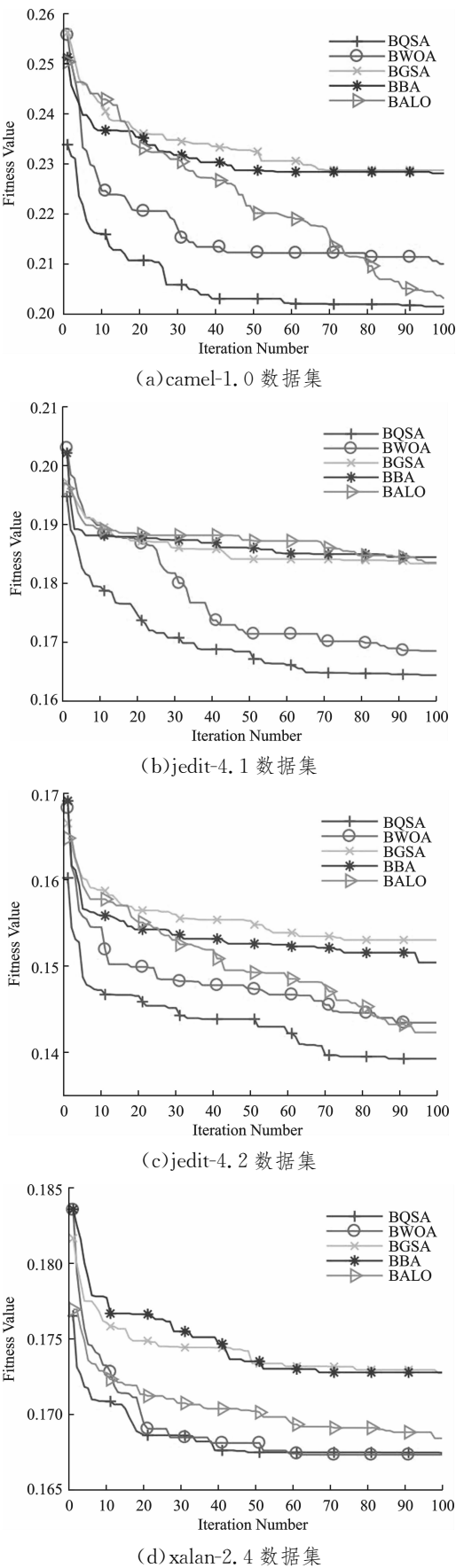


图 3 算法在四个数据集上的收敛曲线

排队搜索算法利用多种行为更新策略,使得算法具有更佳的健壮性,在多类型目标函数寻优上表现出更佳的性能.而所使用的自适应变异机制也使得迭代后期算法能够增强局部开发能力,使算法尽快收敛.

5 结束语

为了实现对非均衡数据集的特征选择,提升分类准确率,结合合成少数过采样方法,设计了一种基于二进制排队搜索理论的特征选择算法 BQSA. 设计了多层次的仿真实验模型.在第一阶段测试了数据集的过采样比例的影响,得到了最佳的过采样率. 比较了 BQSA 与同类算法的性能,结果显示,结合合成少数过采样技术的 BQSA 算法拥有更优的预测准确性,在分类敏感度、专一性及 AUC 指标上表现更佳. 进一步研究可集中于 BQSA 的变种算法上,如利用不同的 S 型或 V 型转换函数,使排队搜索机制具有在搜索最优特征子集上具有更高的适应度.

参考文献:

[1] GOLDBERG M, GOLDBERG R. Data classification: algorithms and applications[J]. Computing Reviews, 2015, 56(12): 724-725.

[2] LI J D, CHENG K W, WANG S H, et al. Feature selection: a data perspective[J]. ACM Computing Surveys, 2017, 50(6): 94. DOI: 10.1145/3136625.

[3] LE THI H A, LE H M, DINH T P. Feature selection in machine learning: an exact penalty approach using a difference of convex function algorithm[J]. Machine Learning, 2015, 101: 163-1836. DOI: 10.1007/s10994-014-5455-y.

[4] ZHANG Y. A modified artificial bee colony algorithm-based feature selection for the classification of high-dimensional data[J]. Journal of Computational and Theoretical Nanoscience, 2016, 13(7): 4088-4095. DOI: 10.1166/jctn.2016.5255.

[5] ABUALIGAH L M, KHADER A T, AL-BETAR M A. Unsupervised feature selection technique based on genetic algorithm for improving the Text Clustering [C]//2016 7th International Conference on Computer Science and Information Technology. Amman: IEEE, 2016: 1-12. DOI: 10.1109/CSIT.2016.7549453.

[6] ZHENG L, DIAO R, SHEN Q. Self-adjusting harmony search-based feature selection[J]. Soft Computing, 2015, 19(6): 1567-1579. DOI: 10.1007/s00500-014-

- 1307-8.
- [7] LU Y H, LIANG M H, YE Z Y, et al. Improved particle swarm optimization algorithm and its application in text feature selection[J]. *Applied Soft Computing*, 2015, 35: 629-636. DOI: 10.1016/j.asoc.2015.07.005.
- [8] 邬开俊, 鲁怀伟. 采用并行协同进化遗传算法的文本特征选择[J]. *系统工程理论与实践*, 2012, 32(10): 2215-2220.
- WU K J, LU H W. PCEGA used to solve text feature selection[J]. *Systems Engineering-Theory & Practice*, 2012, 32(10): 2215-2220. DOI: 10.3969/j.issn.1000-6788.2012.10.012.
- [9] LIN K C, ZHANG K Y, HUANG Y H, et al. Feature selection based on an improved cat swarm optimization algorithm for big data classification[J]. *The Journal of Supercomputing*, 2016, 72(8): 3210-3221. DOI: 10.1007/s11227-016-1631-0.
- [10] DADANEH B Z, MARKID H Y, ZAKEROLHOSSEINI A. Unsupervised probabilistic feature selection using ant colony optimization[J]. *Expert Systems with Applications*, 2016, 53: 27-42. DOI: 10.1016/J.ESWA.2016.01.021.
- [11] 李志鹏, 李卫忠. 基于可拓小生境量子粒子群算法的特征选择[J]. *数据分析与知识发现*, 2017(7): 82-89.
- LI Z P, LI W Z. Feature selection based on modified QPSO algorithm[J]. *Data Analysis and Knowledge Discovery*, 2017(7): 82-89.
- [12] MAFARJA M, MIRJALILI S. Whale optimization approaches for wrapper feature selection[J]. *Applied Soft Computing*, 2018, 62: 441-453. DOI: 10.1016/J.ASOC.2017.11.006.
- [13] GUHA R, GHOSH M, CHAKRABARTI A, et al. Introducing clustering based population in binary gravitational search algorithm for feature selection[J]. *Applied Soft Computing*, 2020, 93: 106341. DOI: 10.1016/J.ASOC.2020.106341.
- [14] RANI A S S, RAJALAXMI R R. Unsupervised feature selection using binary bat algorithm[C]//2015 2nd International Conference on Electronics and Communication Systems (ICECS). Coimbatore: IEEE, 2015. DOI: 10.1109/ECS.2015.7124945.
- [15] EMARY E, ZAWBAA H M, HASSANIEN A E. Binary ant lion approaches for feature selection[J]. *Neurocomputing*, 2016, 213: 54-65. DOI: 10.1016/J.NEUCOM.2016.03.101.

作者简介:

郭嘉女, (1985-), 博士研究生, 讲师. 研究方向为大数据与先进计算. E-mail: performance@aliyun.com