计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(1):242−248 [doi: 10.15888/j.cnki.csa.008284]

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

面向软件缺陷预测的过采样方法①

纪兴哲, 邵培南

(中国电子科技集团第三十二研究所, 上海 201808)
通信作者: 纪兴哲, E-mail: xingzhej@163.com

摘　要: 为了缓解软件缺陷预测的类不平衡问题, 避免过拟合影响缺陷预测模型的准确率, 本文提出一种面向软件缺陷预测的基于异类距离排名的过采样方法 (HDR). 首先, 对少数类实例进行 3 类实例区分, 去除噪声实例, 减少噪声数据导致的过拟合的情况, 然后基于异类距离将实例进行排名, 选取相似度高的实例两两组合产生新实例, 以此来提升新实例的多样性, 之后将有价值的被删除的少数类实例恢复. 实验将 HDR 算法与 SMOTE 算法和 Borderline-SMOTE 算法进行比较, 采用 RF 分类器在 NASA 的 8 个实际项目数据集上进行, 结果显示在 *F*1-measure 和 G-Mean 两项指标上分别有 7.7% 和 10.6% 的性能提升, 实验表明 HDR 算法在处理数据量大并且不平衡率高的软件缺陷预测数据集上明显优于其他两种算法.

关键词: 软件缺陷预测; 类不平衡; 过采样; SMOTE; 异类距离

引用格式: 纪兴哲,邵培南.面向软件缺陷预测的过采样方法.计算机系统应用,2022,31(1):242–248. http://www.c-s-a.org.cn/1003-3254/8284.html

Oversampling Method for Software Defect Prediction

JI Xing-Zhe, SHAO Pei-Nan

(The 32nd Research Institute of China Electronics Technology Group Corporation, Shanghai 201808, China)

Abstract: To alleviate the class imbalance problem of software defect prediction and avoid the influence of overfitting on the accuracy of the defect prediction model, this study proposes an oversampling method for software defect prediction based on heterogeneous distance ranking (HDR). First, a minority of instances are distinguished by three classes to remove noise instances and reduce overfitting caused by noise data. Then, instances are ranked based on heterogeneous distances and paired with highly similar ones to generate new instances for the improvement of new instance diversity. Valuable minority instances that were deleted are restored afterward. The experiment compares the HDR algorithm with the SMOTE and the Borderline-SMOTE algorithms, and the RF classifier is used on the eight actual project data sets of NASA. The results show that there are 7.7% and 10.6% performance improvements on the *F*1-measure and G-Mean indicators respectively. Experimental results show that the HDR algorithm is significantly better than other algorithms in processing software defect prediction data sets with large data volumes and high imbalance rates.

Key words: software defect prediction; class imbalance; oversampling; SMOTE; heterogeneous distance

随着软件技术的发展, 软件项目中的缺陷越来越受开发人员的重视, 占据大量开发和维护时间, 降低开发人员的工作效率, 而且给企业带来许多财物损失. 目前, 软件缺陷预测 (SDP)[1,2] 已经成为了一个快速发展的研究领域, 许多研究[3,4] 大都通过机器学习方法来挖掘版本控制系统数据集, 运用历史数据训练预测模型来预测软件项目的缺陷. 在整个数据集达到平衡的前提下, 机器学习中许多算法可以解决这类问题; 然而,

① 收稿时间: 2021-04-01; 修改时间: 2021-04-29; 采用时间: 2021-05-11; csa 在线出版时间: 2021-12-17

现在的绝大多数软件项目数据集都极为不平衡, 有缺陷的样本都远远少于无缺陷的样本 (前者通常被称为少数类, 后者称为多数类), 这在 SDP 中称为类不平衡问题, 类不平衡问题存在于众多现实领域中, 例如生物医学诊断[5]、多媒体数据分类[6]、垃圾邮件识别[7]、信用卡及电信诈骗检测[8] 等, 这类问题都具有鲜明的类不平衡问题.

许多优秀的用于构建分类预测模型的方法[9] 已经被提出, 包括常用的决策树 (DT), k 最近邻 (KNN) 朴素的贝叶斯, 逻辑回归, 多层感知 (MLP), 支持向量机 (SVM), 极限学习机 (ELM) 和深度神经网络 (DNN) 等, 在对 SDP 问题进行建模预测时, 它们的整体分类准确率通常都不错. 但是当面对类不平衡数据集时, 尤其是例如癌症患者预测[5] 这类高失衡率的问题中, 即只有极少数样本需要正确的预测时, 这些分类算法只能对大多数不患病的样本进行分类, 而忽略了少数患病样本, 这从算法应用的本质上与出发点是相违背的.

解决类不平衡问题的广泛被采用的方法是二次采样法, 代价敏感法和集合学习法[10], 集合学习方法是一种通过叠加多个弱监督分类器以获得更好的强监督分类器的学习方法, 常用的组装学习方法是装袋, 提升和堆叠. 代价敏感法是给每一类实例分配不同的权重, 以使在分类时少数类比多数类更有被分类的价值. 二次采样法包括欠采样和过采样. 欠采样是通过丢弃部分数据集中的多数类来平衡数据集, 例如 RUS, CNN, TL, OSS, 以及 SBC[3] 等, 缺点是丢弃的部分实例可能包含潜在的有用信息; 过采样是合成新实例添加到少数类中, 例如 ROS, SMOTE, 以及 SMOTE 算法的拓展等. 一般情况下, 过采样方法是优于欠采样方法的, 因为过采样避免了信息的丢失, 但也有数据复制导致拟合度过高和多样性不足的困扰.

针对以上缺点, 本文提出一种基于异类距离排名的过采样方法 (HDR), 通过衡量少数类实例与最近多数类之间距离的排名关系, 选出与边界联系最强的实例组合生成新实例, 缓解数据分布的不均衡性并且提升数据的多样性, 在此数据集基础上继而提高软件缺陷预测的准确性. 同时与 SMOTE 算法[11] 和 Borderline-SMOTE 算法[12] 等作比较, 通过对三者算法的研究和对相同数据集进行实验, 挖掘出新方法的优越性. 本文结构如下: 在第 1 节讨论了软件缺陷预测的技术现状, 在第 2 节中讨论了不同的采样方法; 在第 3 节中详细介绍了本文提出的 HDR 算法; 在第 4 节中介绍了实验对象和评估方法以及实验结果; 在第 5 节中对本文进行了总结并展望了未来的工作.

1 软件缺陷预测

为了减少软件故障并提高软件质量, 开发人员往往会采用许多软件质量保证活动 (例如缺陷预测, 代码审查和单元测试等). 根据文献了解到软件故障在软件项目中通常遵从八二定律[13], 即 80% 的缺陷存在于 20% 的模块中, 而软件质量保证活动需要对整个软件项目进行覆盖, 往往会在无缺陷的大量模块上花费若干时间. 为了最大程度的降低成本, 软件开发人员需要知道哪些模块包含更多缺陷, 并首先检查此模块. 由此, 研究人员提出了软件缺陷预测技术.

软件缺陷预测技术的过程包含 3 个主要步骤: 收集历史缺陷数据集; 使用历史数据通过机器学习或深度学习技术来训练分类或回归模型; 将训练后的模型应用与预测软件缺陷的数量或可能性. 随着 PROMISE 知识库[10], NASA 缺陷样本集[14] 和 AEEEM 等数据集相继被公开, 软件缺陷预测技术得到了快速的发展, 并出现了多种不同的分类方法. 从不同的数据集粒度来看, 软件缺陷预测技术可分为 4 类: 包级别, 文件级别, 方法级别, 代码行级别. 从不同指标来看, 分为两类: 静态和动态. 静态缺陷预测采用静态软件指标来预测缺陷数量或缺陷分布, 指标包括 McCabe 度量和 Halstead 度量. 动态缺陷预测采用缺陷生成来预测系统缺陷随时间的分布状况. 采用机器学习方法研究软件缺陷预测分为 3 种不同的方向, 分别是项目间缺陷预测 (WSDP), 跨项目缺陷预测 (CSDP), 以及非均匀缺陷预测 (HDP)[15], 比起相较而言已经成熟的 WSDP, 当前研究人员更关注 CSDP 的发展进程.目前研究人员已经提出了许多缺陷预测方法, 但是大多数缺陷预测研究对软件模块的缺陷倾向性描述较多, 而忽视了模块本身的缺陷数及缺陷分布, 并且机器学习的方法都采用默认参数, 没有探究修改参数带来的影响.

2 软件缺陷预测的二次采样方法

关于 SDP 中的类不平衡问题已有一些研究, 相关文献多是采用是二次采样方法[16]. 二次采样方法分为欠采样和过采样两种.

欠采样是通过删除数据集中的部分多数类, 缩小

多数类的规模达到类平衡, 目前经常作为过采样方法的补充技术.

过采样技术是通过复制现有少数类或生成新的合成少数类来平衡数据集的一种采样方法, 随机过采样(ROS)[16] 是在少数类中随机复制部分插入数据集中, 虽然这种方法能够简单有效的解决类不平衡问题, 但由于重复采样, 数据缺乏多样性, 常常导致严重的过拟合. Chawla 等人在 2002 年提出了 SMOTE 算法[11], 作为迄今为止该研究领域内影响最大的过采样方法, 它基于最近邻思想, 提出根据两个少数类样本在方向上人工生成新的少数类的解决方法, 实验证明在处理类不平衡时能有效缓解数据的不平衡分布. Han 等认为 SMOTE 算法忽视了多数类的分布, 有导致类间重叠的风险, 提出了 Borderline-SMOTE 算法[12], 认为算法应着眼于多数类与少数类的决策边界上. 后人的研究多是依据 SMOTE 算法的优越性和特殊性对其进行拓展, 例如: Bunkhumpornpat 等提出了 Safe-Level-SMOTE 算法[17], 用安全等级系数来保证 K 邻居中其他少数类实例数量; Feng 等人提出了基于复杂度的过采样技术(COSTE)[3], 显著提高了生成实例的质量; He 等人基于密度分布提出了改进算法 ADASYN (自适应合成采样)[18]; Ding 等人在 He 工作的基础上结合 EasyEnsemble 集成方法提出了 KA-Ensemble[9], 动态改善多数类和少数类的平衡分布, 并且在解决多分类问题中依然有效; Romero 等人提出了 KINOS 算法[19], 先删除部分少数类, 对余下数据过采样, 最后将删除部分恢复, 既通过生成新实例拓展了少数类, 又保留了少数类的高价值数据; Tarawneh 等人基于最远邻提出了 SMOTEFUNA 算法[20]; Paria 等人提出了 RCSMOTE[4], 该方法能够改进 SMOTE 算法的泛化问题, 避免对无意义的少数类进行过采样, 同时解决决策边界的重叠问题.

3 基于异类距离排名的过采样方法

本文参考 SMOTE 算法和 Borderline-SMOTE 算法生成新合成实例的方法提出一种新的基于异类距离排名的过采样方法 (HDR). 由于多数类和少数类的边界数据分布情况对分类问题的解决至关重要. HDR 将过采样的重点着眼于边界实例[12] 上, 在结合了噪声过滤, 基于排名和高值数据保留等思想和方法后, 提出了以下方法步骤. 首先, 对样本数据集采用实例分类方法进行区分, 剔除噪声数据, 然后计算边界实例与最近的多数类的距离, 之后 HDR 基于距离以升序排列所有边界实例, 并且将排名相邻的边界实例成对生成新合成实例. 对距离排名的目的是, 较高排名的边界实例更贴合边界, 在数据分类时更具有区分度, 将排名相邻的边界少数实例两两组合进而生成新实例, 实则是通过组合区分度高、代表性强的实例来创造更具多样性, 更有利于均衡类分布的新实例. 在之后, 删除新生成的噪声合成实例并且恢复最开始删除的少数类噪声数据以避免丢失有价值的信息. 以下是 SMOTE 算法, Borderline-SMOTE 以及 HDR 算法的详细描述.

3.1 SMOTE 算法

SMOTE 算法是在随机过采样 (ROS) 基础上改进的一种合成少数类过采样方法, 不同于 ROS 采取简单复制的方法进行过采样, SMOTE 的基本思想是依据少数类实例样本人工生成新的合成实例. SMOTE 如图 1, 具体内容由文献 [11] 详细介绍, 算法流程如下.

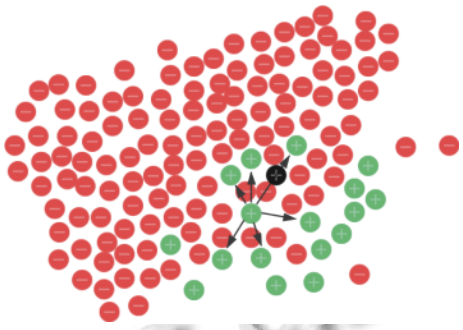

图 1 SMOTE 算法基本思想

1) 在少数类中, 为每一个少数类实例x使用 KNN 算法依据欧氏距离确定 K 个最近邻居;

2) 根据数据集的不平衡率设置一个过采样比例 M, 根据 M 确定采样倍率 N, 对每一个实例x, 从x的 K 近邻中随机选择 T 个实例 $(x_1, x_2, \cdots, x_T)$, T 的大小由 N 决定. 对于选择的每一个实例x_T, 执行以下公式生成少数类新实例x_{new};

3) 将新生成的实例x_{new}添加到原始数据集中, 使训练数据集的不平衡度达到要求.

$$x_{\text{new}} = x_i + n(x_t - x_i), n \in (0,1) \tag{1}$$

SMOTE 算法的缺陷有以下几个方面: 首先对所有的少数类一视同仁, 会导致多数类与少数类的边界会随着新添加的生成实例的增多而越加模糊, 其次对于噪声数据的过采样会增加数据集的不平衡性, 最后对于控制 SMOTE 算法的两个重要参数 K 和 N 需要反复调整.

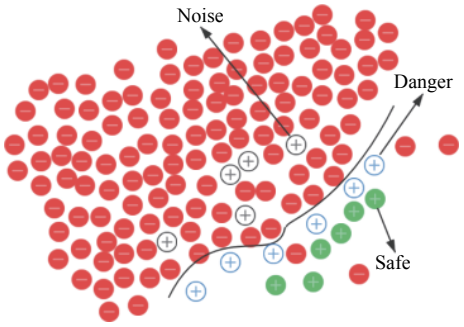

图 2 Borderline-SMOTE 算法技术要点

3.2 Borderline-SMOTE 算法

Borderline-SMOTE 算法是针对 SMOTE 算法的缺陷提出的一种改良算法. SMOTE 对所有的少数类实例都是同等对待. 然而有些实例远离边界, 对分类并没有帮助, 这部分实例在过采样时应该被丢弃, 同时需要强化边界生成更多的边界实例, 来提升数据分布不均衡中的少数类分布比例, Borderline-SMOTE 算法基于此思想被提出. 算法的基本思想是将少数类实例分为 3 类, 分别为 safe, danger, noise, 如图 2, 仅对分类为 danger 的少数类实例进行过采样. 通过类似 Borderline-SMOTE 算法的缩小样本采样策略, 许多研究者相继提出了诸如 SMOTEBoost, AND-SMOTE[21] 和 ADASYN 等算法, 本文提出的 HDR 也基于此思想.

3.3 HDR 算法

HDR 算法可分为 3 个步骤: 样本预处理, 过采样和样本后处理.

1) 样本预处理

样本数据预处理, 首先将样本数据根据标签分为多数类和少数类, 算法的目的是对少数类的过采样进行优化, 基本思想是在不引入嘈杂数据的情况下扩充少数类实例数量. 正如在第 3.2 节中提到的 SMOTE 算法的缺陷之一就是对噪声实例的过采样加剧了数据集的不平衡, 而安全实例的过采样产生的新实例对分类器的学习过程也不会有大的影响. 考虑到上述事实, 首先需要将少数类实例进行分类, 包括边界实例, 安全实例, 噪声实例, 考虑边界实例 (这在步骤 2 中运用) 并且去除噪声实例. 少数类实例的分类方法[4] 类似 Borderline-SMOTE 如算法 1 所示.

算法 1. 少数类分类方法

输入: 数据集$data$, 包含少数类实例d和多数类实例D, $d\in(d_1,d_2,d_3,\cdots,d_{nnum})$, $nnum$为d的数量.
输出: 边界实例 B, 安全实例 S, 噪声实例 N.

方法:
1. 针对d中的每一个d_i, 计算d_i在数据集$data$中的m近邻, 获得m近邻中的多数类实例数量m'.
2. 如果$m'=m$, 即d_i所有的m近邻都是多数类实例, d_i可被判定为噪声实例, $d_i\in N$; 如果$\frac{m}{2}<m'<m$, 即d_i的m近邻中有超过一半的实例是多数类实例, d_i可被判定为边界实例, $d_i\in B$; 如果$m'<\frac{m}{2}$, 即d_i的m近邻中有少于一半的多数类实例, d_i可被判定为噪声实例, $d_i\in S$.

依据算法 1 区分了 3 种不同类型的实例, 在去除了噪声实例 N 后, 在步骤 2) 中不会再有噪声实例生成新实例干扰数据集的平衡性以及分类器的性能. 根据已有研究[12] 表明, 调整 m 的取值, 当边界实例 B 的数量达到少数类实例数量的一半, 此时 m 为最佳值.

2) 过采样

在预处理步骤之后, 过采样步骤确定如何生成新合成少数实例以及生成的数量. 这一步骤的目的是平衡类之间的分布, 提高数据集的多样性, 扩充边界实例, 这将有利于分类器更好的发挥作用. 通过计算边界实例的多数类最近邻的欧式距离, 然后将计算结果升序排名, 将距离相似的两个数类成对组合在一起,之后参考 SMOTE 算法, 在二者之间的连线上随机选择一点作为新生成的少数类实例. 计算的每一个结果, 都是在寻找对分类更有效的边界实例, 越接近多数类的少数类, 具有的分类代表意义越深, 通过组合排名相似的实例, 可以提升生成数据的多样性, 缓解类不平衡问题. HDR-过采样算法如算法 2 所示.

算法 2. HDR-过采样算法

输入: 边界实例 B, 多数类实例 D, 少数类实例d, $B\in(b_1,b_2,b_3,\cdots,b_i)$, $D\in(D_1,D_2,D_3,\cdots,D_i)$
输出: 平衡数据集$newdata$

方法:
1. 定义 x_i:b_i与其多数类最近邻的欧氏距离
2. 定义 $random(a,b)$:a和b之间任取一点作为结果
3. 定义 M:异类距离集合, $M\in(m_1,m_2,m_3\cdots m_i)$
4. 定义 C:生成新实例集合, $C\in(c_1,c_2,c_3,\cdots,c_i)$, $L=length(C)$ 根据文献 [22] 实验表明, 当生成新实例后少数类总量达到总体数量的 40%, L 则为最佳值
5. *for* i=1,2,3$\cdots L$ *then*:
$dism=dist(b_i,D_i)$, D_i为属于多数类的b_i的最近邻
$x_i=dism$, 存入 M 集合
6. 对 M 进行升序排序, 得到$m_1,m_2,m_3,\cdots,m_i$, m_1最小
7. *for* j=1;j=j+2;$j\le length(M)$ *then*:
repeat:
$random(m_i,m_{i+1})\rightarrow c_i$, $length(c_i)=L/length(B)$
$newdata=C+D+d$

3) 样本后处理

虽然新实例是通过一组异类距离相似度高的实例

生成而来, 这意味着大多数情况它都存在于少数类群体内部, 如图 3, 虚线为边界线, 蓝色圆形对勾代表的是根据黑实线两侧异类距离相似的实例随机生成, 新合成实例是安全的, 但不排除有黑色圆形对勾的情况, 呈凹形的边界线分布两侧分布着异类距离相似的实例, 通过 HDR 算法生成的新实例存在于多数类内部, 这将形成噪声, 影响分类. 因此采用 KNN 算法来去除新生成的噪声实例. KNN 算法的核心思想是通过判断当前样本的邻居所属类别来推断样本所属的类别. 首先, 应用欧几里得距离来计算每个实例的 K 近邻, 在其 K 邻域内, 如果少数类的数量小于多数类的数量, 就意味着当前实例是噪声实例, 并且应该被删除.

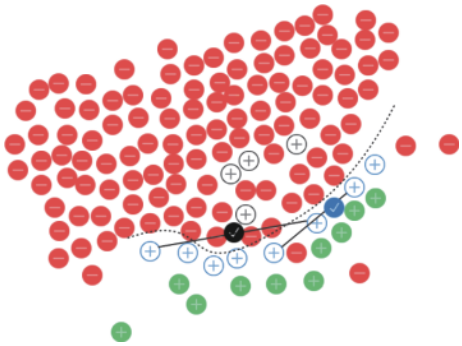

图 3 HDR-过采样算法生成新合成实例

由于有缺陷实例部分通常较少, 每一个实例都对应了真实数据, 删除任意一个都是对数据样本的污染, 因此采取与诸多文献不同的后处理方式 在预处理步骤中去除的少数类噪声样本需要被添加回过采样后的数据集. 同时为了研究恢复实例对整个算法的影响, 将未进行恢复操作的 HDR 算法命名为 HDR-half 算法, 并通过实验进行研究.

4 实验分析

4.1 实验数据集

为了验证 HDR 算法的有效性, 实验选用了 NASA MDP 数据集, MDP 数据集包含了 13 个 NASA 实际项目, 在 SDP 研究文献内被多次运用. 表 1 选取了 MDP 8 个具有代表性的数据集的详细信息, 包括项目名, 度量属性数, 总实例数量和有缺陷实例所占比例. 其中度量属性包含了 McCabe 度量, Halstead 度量, 代码行数, 操作数复杂度等, 图中可以观察到, 有缺陷实例所占比例都远低于 50%, 这表明 MDP 数据集可以用做评估解决类不平衡方法的实验对象.

表 1 NASA 的 MDP 部分数据集

Dataset	Examples	Features	Defect class	Ratio (%)
CM1	505	37	48	9.5
JM1	10878	21	2102	19.3
KC1	2 107	21	325	15.4
KC3	458	39	43	9.4
MC1	9 466	38	68	0.7
MC2	161	39	52	32.3
PC1	1107	37	76	6.9
PC2	5 589	36	23	0.4

4.2 实验评价指标

在 SDP 中, 通常基于混淆矩阵计算性能评价指标, 如表 2 所示. 其中真阳性 (TP) 表示正确预测为缺陷实例的数量, 假阳性 (FP) 表示错误预测为缺陷实例的数量, 真阴性 (TN) 表示正确预测为无缺陷实例的数量, 假阴性 (FN) 表示错误预测为无缺陷实例的数量.

表 2 混淆矩阵

类别	预测为真	预测为假
实际为真	*TP*	*FN*
实际为假	*FP*	*TN*

为了有效评估 HDR 算法的性能, 实验选用了平衡率 (*balance*), *F* 测量 (*F*1-measure, *F*1) 和 G 均值 (G-Mean, *GM*)3 项性能指标, *balance* 是召回率和准确率之间的平衡, *balance* 值越大, 证明实验方法性能越好. *F*1-measure 体现模型的稳定性, 当 *F*1-measure 较高时表明实验方法有更好的性能. G 均值也可以用来评价不平衡数据的模型表现. 根据混淆矩阵, *balance*, *F*1, *GM* 定义如下:

$$balance = 1 - \frac{\sqrt{\left(1-\frac{TP}{TP+FN}\right)^2+\left(\frac{TP}{TP+FP}\right)^2}}{\sqrt{2}} \tag{2}$$

$$F1 = \frac{2TP}{2TP+FP+FN} \tag{3}$$

$$GM = \sqrt{\frac{TP}{TP+FP}\cdot\frac{TP}{TP+FN}} \tag{4}$$

4.3 实验设计

为验证 HDR 方法的有效性, 实验分别与 SMOTE、Borderline-SMOTE 实验作比较; 为研究算法中步骤 3) 处理方法的影响, 实验将 HDR-half 算法与 HDR 算法进行比较. 实验中所涉及的近邻参数均按照文献设置为 5, 分类器采用随机森林 (RF), RF 作为决策树 (DT) 的集成方式, 目前已被广泛地应用在分类问题中, 它的特点是在处理数据量较大或度量属性较多的项目时训

练速度快, 训练成本低. 实验采用 5 折交叉验证, 将数据集的 80% 作为训练集, 20% 作为测试集, 训练过程重复 20 次, 性能结果取 20 次 5 折交叉验证平均值.

4.4 实验结果分析

表 3, 表 4 和表 5 分别列出了采用 4 种方法关于 MDP 数据集的 *balance*, *F*1-measure 和 G-Mean 三项指标. 从表中可以看到, HDR-half 算法相比于 HDR 算法在 *balance* 指标上有相似的表现, 而在 *F*1-measure 和 G-Mean 指标上则得到了相比 HDR 算法较差的结果, 由此, 本文采用的在过采样之后恢复其原始部分少数类数据的独特方法, 减少了宝贵数据的浪费, 丰富了数据集和数据分布, 并且对实验结果有巨大的提升.

从表 3 中可以看到, 在 HDR 算法采用 RF 分类器得到的 *balance* 值相对于 SMOTE 和 Borderline-SMOTE 取得了全面的优势, 尤其在 JM1 数据集上相比于 SMOTE 具有 23.3% 的巨大提升, 最低提升发生在 KC3 数据集上, 提升了 3.5%, 平均有 8.76 的提升. 这些提升都源自新算法在数据集上进行了预处理+后处理的操作; 同样, 从表 4 和表 5 来看, 本文提出的新算法结果明显较好, 相较于 SMOTE 算法在 *F*1-measure 上和 G-Mean 上分别有 8.76% 和 16.89% 的性能提升, 和 Borderline-SMOTE 相比也分别有 6.64% 和 4.24% 的性能提升. 表 4 中 Borderline-SMOTE 在 KC3 数据集和 MC2 数据集上结果较好. 经过多次实验分析, 原因是两者的数据量太小导致新算法对数据集的优化不如 Borderline-SMOTE 明显. 总体而言, 相较于同类型算法 Borderline-SMOTE, HDR 采用异类距离排名的方法稳定性更强, 对不平衡数据集预测性能的提升更明显.

表 3 3 种方法在数据集上的 *balance* 值

数据集	CM1	JM1	KC1	KC3	MC1	MC2	PC1	PC2
SMOTE	0.603 6	0.618 7	0.664 0	0.732 8	0.714 8	0.708 7	0.651 3	0.647 2
Borderline-SMOTE	0.634 7	0.675 4	0.678 8	0.754 1	0.701 2	0.751 4	0.696 2	0.657 4
HDR-half	0.650 3	0.756 5	0.738 9	0.749 9	0.761 9	0.768 4	0.775 7	0.710 0
HDR	**0.648 4**	**0.762 9**	**0.735 6**	**0.758 7**	**0.762 5**	**0.771 2**	**0.772 6**	**0.702 2**

表 4 3 种方法在数据集上的 *F*1-measure 值

数据集	CM1	JM1	KC1	KC3	MC1	MC2	PC1	PC2
SMOTE	0.283 6	0.356 3	0.394 5	0.358 8	0.124 7	0.542 9	0.285 4	0.203 5
Borderline-SMOTE	0.261 4	0.384 5	0.362 8	0.392 4	0.140 2	0.572 7	0.296 4	0.211 6
HDR-half	0.303 3	0.392 5	0.440 2	0.384 2	0.137 8	0.556 3	0.309 7	0.224 3
HDR	**0.354 5**	**0.412 9**	**0.451 7**	0.381 2	**0.156 5**	0.563 5	**0.354 1**	**0.226 3**

表 5 3 种方法在数据集上的 G-Mean 值

数据集	CM1	JM1	KC1	KC3	MC1	MC2	PC1	PC2
SMOTE	0.557 5	0.596 1	0.580 3	0.621 7	0.612 7	0.605 3	0.634 6	0.492 1
Borderline-SMOTE	0.645 2	0.605 3	0.610 8	0.627 8	0.724 1	0.624 2	0.690 0	0.706 2
HDR-half	0.653 9	0.612 4	0.578 5	0.639 5	0.773 1	0.650 6	0.732 9	0.684 9
HDR	**0.714 2**	**0.623 1**	0.578 7	**0.650 4**	**0.781 2**	**0.673 9**	**0.737 1**	**0.707 1**

综上所述, 本文在 SMOTE 算法和 Borderline-SMOTE 算法的影响下提出的 HDR 算法, 在解决 SDP 的类不平衡问题时, 有效缓解了数据分布的不平衡性, 并且其生成的多样性新合成实例也大大提升了分类器的性能, 提高了模型预测的准确率.

5 结论与展望

由于已有的过采样技术往往导致不同程度的过拟合, 并且算法的处理过程通常覆盖绝大多数实例, 提高了模型预测的成本. 针对以上缺点, 本文提出一种基于异类距离排名的过采样方法 (HDR), HDR 利用边界实例与最近多数类的欧氏距离, 帮助生成更加多样性的新合成实例, 同时噪声过滤步骤大大降低了新实例导致的过拟合的可能性. 本文通过在 NASA 的 MDP 实际项目数据集上采用 RF 分类器与两种过采样算法进行比较来评估 HDR 的性能, 在用来评估的 3 个指标 (*balance*, *F*1-measure, G-Mean) 上可以看到经过 HDR 算法处理过的不平衡数据集在软件预测模型的性能上体现了较大的优势, 然而同时也看到了 HDR 面对小样本数据集的性能缺陷. 因此, 如何调整 HDR 算法使之能较好的解决各种不同大小的数据集以及采取更多的分类器验证其性能将是接下来的研究重点.

参考文献

1 刘童. 基于机器学习算法的软件缺陷预测技术研究 [硕士学位论文]. 武汉: 华中师范大学, 2018.1.

2 Sun ZB, Song QB, Zhu XY. Using coding-based ensemble learning to improve software defect prediction. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2012, 42(6): 1806–1817. [doi: 10.1109/TSMCC.2012.2226152]

3 Feng S, Keung J, Yu X, *et al*. COSTE: Complexity-based oversampling technique to alleviate the class imbalance problem in software defect prediction. Information and Software Technology, 2021, 129: 106432. [doi: 10.1016/j.infsof.2020.106432]

4 Soltanzadeh P, Hashemzadeh M. RCSMOTE: Range-controlled synthetic minority over-sampling technique for handling the class imbalance problem. Information Sciences, 2021, 542: 92–111. [doi: 10.1016/j.ins.2020.07.014]

5 Zięba M, Tomczak JM, Lubicz M, *et al*. Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. Applied Soft Computing, 2014, 14: 99–108. [doi: 10.1016/j.asoc.2013.07.016]

6 Li YJ, Guo HX, Zhang QP, *et al*. Imbalanced text sentiment classification using universal and domain-specific knowledge. Knowledge-Based Systems, 2018, 160: 1–15. [doi: 10.1016/j.knosys.2018.06.019]

7 Irtazal A, Adnan SM, Ahmed KT, *et al*. An ensemble based evolutionary approach to the class imbalance problem with applicationsinCBIR.AppliedSciences,2018,8(4):495.[doi:10.3390/app8040495]

8 Pun J, Lawryshyn Y. Improving credit card fraud detection using a meta-classification strategy. International Journal of Computer Applications, 2012, 56(10): 41–46. [doi: 10.5120/8930-3007]

9 Ding H, Wei B, Gu ZR, *et al*. KA-Ensemble: Towards imbalanced image classification ensembling under-sampling and over-sampling. Multimedia Tools and Applications, 2020, 79(21): 14871–14888. [doi: 10.1007/s11042-019-07856-y]

10 Cai XJ, Niu Y, Geng SJ, *et al*. An under-sampled software defect prediction method based on hybrid multi-objective cuckoo search. Concurrency & Computation Practice & Experience, 2020, 32(5): e5478. [doi: 10.1002/cpe.5478]

11 Gong LN, Jiang SJ, Jiang L. Tackling class imbalance problem in software defect prediction through cluster-based over-sampling with filtering. IEEE Access, 2019, 7: 145725–145737. [doi: 10.1109/ACCESS.2019.2945858]

12 Han H, Wang WY, Mao BH. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. International Conference on Intelligent Computing. Hefei: Springer, 2005. 878–887.

13 Bejjanki KK, Gyani J, Gugulothu N. Class Imbalance Reduction (CIR): A novel approach to software defect prediction in the presence of class imbalance. Symmetry, 2020, 12(3): 407. [doi: 10.3390/sym12030407]

14 NezhadShokouhi MM, Majidi MA, Rasoolzadegan A. Software defect prediction using over-sampling and feature extraction based on Mahalanobis distance. The Journal of Supercomputing, 2020, 76(1): 602–635. [doi: 10.1007/s11227-019-03051-w]

15 Xia X, Lo D, Pan SJ, *et al*. HYDRA: Massively compositional model for cross-project defect prediction. IEEE Transactions on Software Engineering, 2016, 42(10): 977–998. [doi: 10.1109/TSE.2016.2543218]

16 García V, Sánchez JS, Marqués AI, *et al*. Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. Expert Systems with Applications, 2020, 158: 113026. [doi: 10.1016/j.eswa.2019.113026]

17 Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. Safe-Level-SMOTE: Safe-level-synthetic minority over-sampling TEchnique for handling the class imbalanced problem. 13th Pacific-Asia Conference, PAKDD 2009. Bangkok: Springer, 2009. 475–482.

18 He HB, Bai Y, Garcia EA, *et al*. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). Hong Kong: IEEE, 2008. 1322–1328.

19 de Morais RFAB, Vasconcelos GC. Boosting the performance of over-sampling algorithms through under-sampling the minority class. Neurocomputing, 2019, 343: 3–18. [doi: 10.1016/j.neucom.2018.04.088]

20 Tarawneh AS, Hassanat ABA, Almohammadi K, *et al*. SMOTEFUNA: Synthetic minority over-sampling technique based on furthest neighbour algorithm. IEEE Access, 2020, 8: 59069–59082. [doi: 10.1109/ACCESS.2020.2983003]

21 Yun J, Ha J, Lee JS. Automatic determination of neighborhood size in SMOTE. Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication. Danang: ACM, 2016. 1–8. [doi: 10.1145/2857546.2857648]

22 李冉, 周丽娟, 王华. 面向类不平衡数据集的软件缺陷预测模型. 计算机应用研究, 2018, 35(9): 2806–2810. [doi: 10.3969/j.issn.1001-3695.2018.09.057]