DOI: 10.16452/j.cnki.sdkjzk.2021.02.010 文章编号:1672-3767(2021)02-0084-11

一种软件缺陷不平衡数据分类新方法

刘文英,林亚林,李克文,雷永秀
(中国石油大学(华东)计算机科学与技术学院,山东 青岛 266580)

摘 要:针对软件缺陷预测数据中的数据不平衡、预测精度低以及特征维度高的问题,提出了一种 RUS-RSMOTE-PCA-Vote 的软件缺陷不平衡数据分类方法。首先通过随机欠采样来减少无缺陷样本的数量;在此基础上进行 SMOTE 过采样,在过采样中综合总体样本的分布状况引入影响因素 posFac 指导新样本的合成;对经过 RUS-RSMOTE 混合采样处理后的数据集进行 PCA 降维,最后应用 Vote 组合 K 最近邻、决策树、支持向量机构造集成分类器。在 NASA 数据集上的实验结果表明,与现有不平衡数据分类方法相比,所提方法在 F-value 值、G-mean 值和 AUC 值上更优,有效地改善了软件缺陷预测数据集的分类性能。

关键词:软件缺陷预测;不平衡数据;混合采样;特征降维;集成分类器
中图分类号:TN929.5 **文献标志码**:A

A novel unbalanced data classification method for software defects

LIU Wenying, LIN Yalin, LI Kewen, LEI Yongxiu
(College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, Shandong 266580, China)

Abstract: To solve the problems of data imbalance, low prediction accuracy and feature dimension in software defect prediction data, a RUS-RSMOTE-PCA-Vote (random under sampling-random synthetic minority oversampling technique-principal components analysis-vote) software defect imbalance data classification method was proposed. Firstly, the number of non-defective samples was reduced by random under sampling. On this basis, SMOTE oversampling was carried out, during which the influence factor posFac (position factor) was introduced into the overall sample distribution to guide the synthesis of the new sample. Then the data set after RUS-RSMOTE sampling was subjected to PCA dimensionality reduction. Finally, an integrated classifier was constructed by using Vote in combination with K nearest neighbor, decision tree, and support vector machine. The experimental results on the NASA (National Aeronautics and Space Administration) data set show that the proposed method is superior to the existing unbalanced data classification methods in terms of F-value, G-mean value and AUC value, thus effectively improves the classification performance of the software defect prediction data set.

Key words: software defect prediction, unbalanced data, hybrid sampling, feature dimensionality reduction, ensemble classifier

对软件缺陷预测的研究表明,80%的缺陷集中发生在 20%的模块中,这说明软件系统中的数据分布是不平衡的,有缺陷模块的数量远远少于无缺陷模块的数量。虽然有缺陷类样本的数量很少,但正确识别有缺陷样本是软件缺陷预测的关键,错误预测有缺陷样本可能会导致遗漏关键错误从而增加软件开发成本。因此,解决不平衡数据问题对于提高软件质量、减少预测误差和成功部署软件具有重要意义。

收稿日期:2020-05-31
基金项目:国家自然科学基金项目(61673396);山东省自然科学基金项目(ZR2017MF032)
作者简介:刘文英(1968—),女,山东蓬莱人,副教授,主要研究领域为计算智能、软件质量与可靠性等.
林亚林(1995—),女,山东青岛人,硕士研究生,主要从事软件缺陷预测研究,本文通信作者.
E-mail:664103127@qq.com

不平衡数据处理[1]是机器学习研究中的热点之一,更是软件缺陷预测方向不可或缺的部分,已有不少学者专注于不平衡数据的研究。目前公认的解决不平衡数据问题的方法是从数据和算法两个层面进行的。数据层面通过采样方法进行处理,算法层面通过应用集成分类算法、代价敏感算法等处理数据。例如,Patel等[2]提出合并自适应 K 最近邻和模糊 K 最近邻的方法来提高不平衡数据的分类性能。Marwa 等[3]提出一种多目标进化方法优化倾斜决策树,并将其应用到不平衡数据分类。Liu 等[4]提出了两阶段成本敏感学习方法,该方法将成本信息处理分为特征选择和分类两个阶段。Rodriguez 等[5]比较了成本敏感算法、采样方法、混合技术和集成技术来处理不平衡数据集,结果表明,不平衡数据经过处理后提高了预测模型的性能。Ruchika 等[6]提出一种新的过采样算法 SPIDER3,并将采样算法与成本敏感分类器结合使用,解决缺陷数据集的不平衡问题,提高预测模型的性能。Siers 等[7]提出成本敏感决策树和成本敏感投票两种技术来处理不平衡数据问题,还结合 SMOTE 过采样方法来优化决策森林进行软件缺陷预测。张忠林等[8]提出一种融合了阈值移动技术和 Bagging 算法的 PT-Bagging 集成算法,相较于 Bagging 算法,所提算法在处理不平衡数据时具有更好的性能。Yuan 等[9]提出一种新的集成学习算法 SE-gcForest,在 gcForest 算法基础上结合 SMOTE 和 EasyEnsemble 来处理不平衡数据。

随着实际软件系统复杂度越来越高,相应的软件缺陷预测面临的数据不平衡问题也越来越突出,目前已出现了一些关于软件缺陷预测与不平衡数据相结合的研究,但是采样技术、特征降维、软件缺陷预测模型等仍然存在很多问题值得去探索。由于混合采样技术理论上同时具备欠采样和过采样的优点,本研究提出一种改进的 RUS-RSMOTE 混合采样方法,克服 SMOTE 算法合成新样本时随机数取值不精确的缺点,引入影响因素 posFac 对随机数进行约束,在此基础上结合随机欠采样技术,对不平衡的软件缺陷数据集进行处理,并利用 F-value、AUC、G-mean 对不平衡的软件缺陷数据分类结果进行评价。

1 相关算法

1.1 随机欠采样(Random under-sampling, RUS)

欠采样技术通过一定的规则或公式减少多数类样本的数量,改变不平衡数据集的样本分布,缓解数据集的不平衡程度,降低计算成本。随机欠采样、压缩最近邻[10]、Tomek links 方法以及邻域清理都是常用的欠采样方法。

随机欠采样通过删除数据集中的多数类样本实现类分布均衡的目的,进而提高分类模型的效率,但是由于删除多数类样本的过程具有随机性、偶然性,容易使重要信息丢失,降低学习器的分类效果。例如,NASA 发布的软件缺陷数据集 PC2 中有 5 589 个样本,其中 23 个(缺陷率 0.41%)为有缺陷样本,5 566 个为无缺陷样本,不平衡率高达 242,若使用随机欠采样技术实现 1∶1(无缺陷样本数∶有缺陷样本数)的数据分布,那么经过随机欠采样处理后的新数据集只有 46 个样本(23 个无缺陷样本和 23 个有缺陷样本)。图 1 是某一不平衡数据集经过随机欠采样前后的样本分布图。

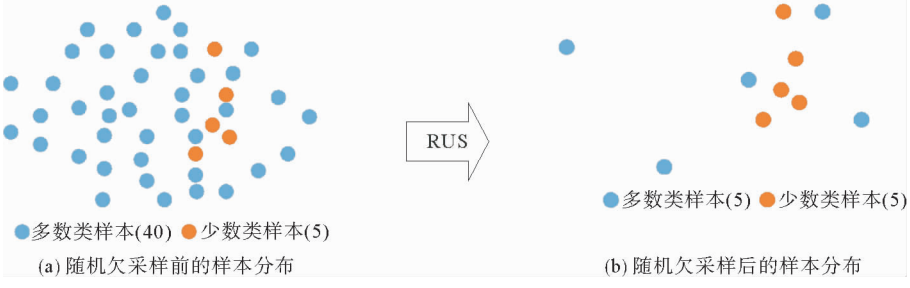


图 1 随机欠采样前后样本分布图

Fig. 1 Sample distribution before and after random under-sampling

1.2 SMOTE

SMOTE 算法是最具代表性的过采样算法,通过相邻少数类样本间线性插值生成新样本[11],从而改变

不平衡数据集的类分布情况。SMOTE 算法通过少数类样本间线性插值的方式生成新样本,克服了随机过采样由于单纯复制少数类样本而导致分类器过拟合的现象,但也存在两个方面的问题,一是增加了数据集样本量,从而增加了分类器训练时间;二是函数 rand(0,1)取值范围过大,未进行精细控制,导致新合成样本质量无法得到保证。

过采样技术和欠采样技术都是通过改变样本数量来改变类分布,从而缓解不平衡数据集的样本分布情况,降低不平衡率。这两种采样技术都有各自的优缺点,目前关于两种技术的比较尚未形成统一的定论。而混合采样技术[16]将欠采样和过采样结合起来,首先对不平衡数据集进行欠采样,然后在此基础上进行过采样。混合采样可以是欠采样和过采样具体方法中任意两种的组合体。

1.3 主成分分析

主成分分析(principal components analysis,PCA)实质是用一组正交向量对原始特征进行变换得到新特征,且新特征间互不相关。数据本身决定着数据从原来的坐标系转换到新坐标系,第一个新坐标轴是由原始数据中方差最大的方向确定的,第二个新坐标轴是由和第一个新坐标轴正交并且方差最大的方向确定的,不断重复该过程,使得重复次数等于原始数据中的特征数 d,得到投影变化后的新坐标轴为$\{w_1,w_2,\cdots,w_d\}$。研究发现,大部分方差都包含在最开始的$\{w_1,w_2,w_3,\cdots,w_{d'}\}$几个坐标轴中,剩下的坐标轴可以忽略,将维度从 d 降低到d'实现特征降维,也可以设置一个重构阈值 t,选择占原始数据的方差中一定百分比的特征向量,t 一般取 95%。PCA 适用于对数值型数据进行处理,优点是仅需识别最重要的多个特征来降低数据的复杂性,缺点是有可能损失重要信息。

1.4 Vote

集成分类器通过某种策略结合多个单分类器来完成学习任务,实际上是一种组合分类器的方法,集成分类器的结构如图 2 所示。如果把单分类器看作是一位决策者,那么集成分类器相当于综合多个决策者的智慧来解决问题。集成分类器可以根据分类器的类型分为同质集成和异质集成,同质集成是由类型相同的单分类器结合而成,例如"决策树集成分类器"中的单分类器都是决策树;异质集成是由类型不同的单分类器结合而成,例如同时包含决策树和朴素贝叶斯的集成分类器。在不平衡的软件缺陷数据集上应用集成分类器,可以使得集成分类器的泛化能力明显高于单分类器,避免过拟合现象的发生,有效地降低了单分类器在不平衡数据分类时所产生的偏差[12]。

Vote 是一种集成分类器结合策略,包括三种投票方法:相对多数投票法即少数服从多数,预测结果是得票最高的类;绝对多数投票法要求最终预测类别所得票数过半;加权投票法将每个类别的票数进行加权求和,结果最大的类别即为预测结果。投票策略既可以集成相同类型的分类器,又可以集成不同类型的分类器,因此可以使用投票策略构建异质集成分类器,综合多个分类器的优势构建预测模型。

本研究利用 5 种集成技术(如表 1)来构建软件缺陷预测模型。

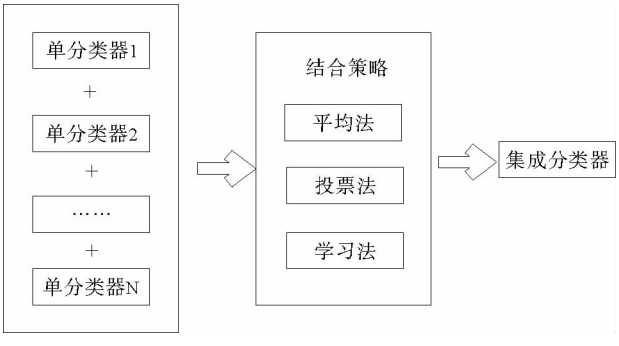


图 2 集成分类器结构

Fig. 2 Structure of the ensemble classifier

2 基于 RUS-RSMOTE-PCA-Vote 的软件缺陷不平衡数据分类方法

利用混合采样技术可以同时具备欠采样和过采样的优点,提出一种改进的 RUS-RSMOTE 混合采样方法。首先对原始缺陷数据集进行随机欠采样来减少无缺陷样本的数量;然后使用 SMOTE 算法进行过采样,同时考虑无缺陷样本分布的影响作用,融入影响因素 posFac 约束随机数 rand(0,1)的取值。为解决不平衡数据集中的数据维度问题,将提出的RUS-RSMOTE与PCA相结合对软件缺陷数据集进行处理。首先对

表 1 本研究采用的五种分类器及相关文献

Tab. 1 Information of 5 classifiers used in this paper

序号	名称(简写)	中文名称	文献
1	Bootstrap aggregating(Bagging)	套袋法	[8,12-13]
2	Adaptive Boosting(AdaBoost)	自适应提升法	[12-14]
3	RandomTree(RT)	随机树	[12,15]
4	RandomForest(RF)	随机森林	[12-13]
5	Vote	投票法	[12,17]

不平衡数据集进行 RUS-RSMOTE 混合采样,根据类别标记将样本划分为有缺陷样本和无缺陷样本,对无缺陷样本进行随机欠采样,对有缺陷样本进行 RSMOTE 过采样;然后对经过 RUS-RSMOTE 混合采样处理后的数据集进行 PCA 降维,将所有样本进行中心化,计算样本的协方差矩阵 XXT 并进行特征值分解,最后取前面最大的 d' 个特征值对应的特征向量进行降维。最后,应用 Vote 集成到的个体学习器,构建适合不平衡数据的软件缺陷预测模型。

2.1 改进的 RSMOTE 过采样方法

针对 SMOTE 算法合成新样本时随机数取值不精确的缺点,引入影响因素 posFac 对随机数进行约束[18],使得新样本合成过程中的随机数取值更有针对性以更加合理地扩展少数类样本,使新数据集趋于平衡,提高分类器对于少数类样本的分类能力。影响因素 posFac 的计算如下:

1) 计算有缺陷样本 x_i 与其 K 个同类近邻的平均欧几里得距离

$$d_{\text{pave-}i}=\frac{\sum_{j=1}^{K}\sqrt{\|x_i-x_j\|^2}}{K}。\tag{1}$$

2) 计算所有有缺陷样本 x_i 与其 K 个同类近邻的平均欧几里得距离之和

$$d_{\text{pave-sum}}=\sum_{i=1}^{m}d_{\text{pave-}i}。\tag{2}$$

其中,m 表示有缺陷样本的数量。

3) 计算 m 个有缺陷样本与同类近邻的平均欧几里得距离和的均值

$$d_{\text{pave}}=\frac{d_{\text{pave-sum}}}{m}。\tag{3}$$

4) 计算有缺陷样本 x_i 与其 K 个无缺陷样本近邻的平均欧几里得距离

$$d_{\text{nave-}i}=\frac{\sum_{j=1}^{K}\sqrt{\|x_i-y_j\|^2}}{K}。\tag{4}$$

5) 计算所有有缺陷样本 x_i 与其 K 个无缺陷样本近邻的平均欧几里得距离之和

$$d_{\text{nave-sum}}=\sum_{i=1}^{m}d_{\text{nave-}i}。\tag{5}$$

6) 计算有缺陷样本与无缺陷样本间的平均欧几里得距离和的均值

$$d_{\text{nave}}=\frac{d_{\text{nave-sum}}}{m}。\tag{6}$$

7) 计算当前被选中的边界样本与其 K 个同类近邻的平均欧几里得距离

$$d_1=(\sum\sqrt{\|x_i-x_j\|^2})/K。\tag{7}$$

8) 计算当前被选中的边界样本与其 K 个无缺陷样本近邻的平均欧几里得距离

$$d_2=\sum_{i=1}^{m}d_i。\tag{8}$$

9) 计算相对距离比

$$u=\frac{d_1/d_{\text{pave}}}{d_2/d_{\text{nave}}}。\tag{7}$$

10) 得到影响因素

$$\text{posFac}=\begin{cases}\text{rand}(0,1), & u<1\\ 0.5+0.5\times\text{rand}(0,1), & 1\leqslant u\leqslant 2\\ 0.8+0.2\times\text{rand}(0,1), & u>2\end{cases}。\tag{8}$$

基于样本分布的 RSMOTE 算法描述:计算任意一个少数类样本 x_i 到数据集中所有同类样本的欧几里得距离,接着寻找样本 x_i 的 K 最近邻,根据采样倍率 N 从 x_i 的 K 最近邻中随机选择 N 个样本与 x_i 进行线性插值合成新样本 x_{new},假设 x_j 为被选中的 x_i 的 K 最近邻样本,新样本合成公式为 $x_{\text{new}}=x_i+\text{posFac}(x_j-x_i)$。

2.2 基于 RUS-RSMOTE-PCA-Vote 的软件缺陷不平衡数据分类方法

基于 RUS-RSMOTE 和 PCA 的特征降维框架如图 3 所示,相应的算法具体过程如下:

输入:DataSet-原不平衡数据集;

输出:带有标记的分类结果。

//RUS-RSMOTE 混合采样阶段:

根据类别标记将 DataSet 划分成 DefectSet 和 NonDefectSet;

对数据集 NonDefectSet 按照预期达到的不平衡率进行随机欠采样,记为数据集 newNonDefectSet;

有缺陷样本数 m=DefectSet.size( );

对数据集 DefectSet 进行随机化处理;

初始化随机变量 $i=0$;

WHILE $i<m$ 计算有缺陷样本 x_i 的 K 个最近邻缺陷样本,同时计算其对应的影响因素 posFac,根据公式 $x_{\text{new}}=x_i+\text{posFac}\times(x_j-x_i)$ 合成新样本,并保存至 newDataSet;

$i++$;

END WHILE;

newDataSet=newDataSet+DefectSet+newNonDefectSet;

//PCA 降维阶段:

将所有样本 x_i 进行中心化;

计算样本的协方差矩阵 XXT 并进行特征值分解;

对特征值按从大到小的顺序进行排序;

取前面最大的 d' 个特征值对应的特征向量 $w_1,w_2,\cdots,w_{d'}$;

利用特征向量实现数据降维得到新数据集;

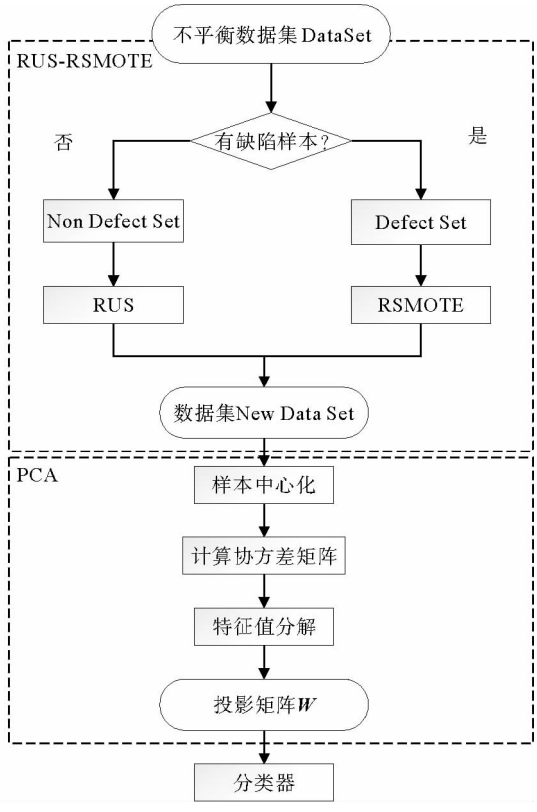


图 3 基于 RUS-RSMOTE 和 PCA 的特征降维框架

Fig. 3 Feature dimensionality reduction framework based on RUS-RSMOTE and PCA

//Vote 构建集成分类器阶段:

将经过前两个阶段处理后的数据集在朴素贝叶斯、决策树、支持向量机、K 最近邻 4 种算法上进行分类;

分析所有数据集在朴素贝叶斯、决策树、支持向量机、K 最近邻 4 种算法上的分类效果,确定最终用于集成的个体分类器,组合规则为“Average of Probabilities”。

3 实验设计与分析

3.1 实验对象

利用数据挖掘工具 WEAK,使用 NASA 发布的软件缺陷数据集进行实验,实验数据集的具体情况见表 2。

表 2 用于实验的 10 个软件缺陷数据集基本信息

Tab. 2 Basic information about 10 software defect data sets used in the experiment

名称	语言	特征数	样本总数	有缺陷样本	无缺陷样本	缺陷率/%	不平衡率
CM1	C	41	505	48	457	9.50	9.52
KC1	C++	22	2 107	325	1 782	15.42	5.48
KC3	Java	41	458	43	415	9.39	9.65
MC1	C++	40	9 466	68	9 398	0.72	138.21
MW1	C	41	403	31	372	7.69	12.00
PC1	C	41	1 107	76	1 031	6.87	13.57
PC2	C	41	5 589	23	5 566	0.41	24.20
PC3	C	41	1 563	160	1 403	10.24	8.77
PC4	C	41	1 458	178	1 280	12.21	7.19
PC5	C++	40	17 186	516	16 670	3.00	32.31

3.2 实验评价指标

通常情况下精准率(precision)和召回率(recall)是评价分类器性能的常用指标,但是对于软件缺陷预测模型而言,由于面临着数据不平衡问题,不适合使用上述两个指标进行模型评价,本研究使用 F-value、AUC、G-mean 作为评价指标。

F-value 是精准率和召回率的调和均值,是不平衡数据分类问题中常用的评价指标,当精准率和召回率的取值都大时,F-value 值才大,且值越大代表预测模型性能越好。计算公式为:

$$F\text{-value}=\frac{2\times recall\times precision}{recall+precision}。\tag{9}$$

AUC(Area Under the Curve)表示 ROC 曲线与坐标轴所围成的面积,取值范围是 0～1,是不平衡数据分类问题中常用的评价指标,AUC 值越大,则预测模型的性能越好。

G-mean 是有缺陷样本召回率和无缺陷样本召回率的几何均值,是衡量不平衡软件缺陷数据集整体分类情况的性能评价指标,只有当有缺陷样本和无缺陷样本的召回率都较大时,G-mean 值才大,同样值越大代表预测模型性能越好。计算公式为:

$$G\text{-mean}=\sqrt{\frac{TP}{TP+FN}\times\frac{TN}{TN+FP}}。\tag{10}$$

数据集中样本的预测结果要么是“有缺陷”要么是“无缺陷”,是一个典型的二分类问题,样本的预测类别与实际类别相比会产生 4 种结果(如表 3),TP 表示正确分类的有缺陷样本数,TN表示正确分类的无缺陷样本数,FP 表示实际为无缺陷类但被预测为有缺类的样本数,FN 表示实际为有缺陷类但被预测为无缺类的样本数。

表 3 分类器情况统计

Tab. 3 Statistics of classification

	预测为 有缺陷	预测为 无缺陷
实际为 有缺陷	TP (True Positive)	FN (False Negative)
实际为 无缺陷	FP (False Positive)	TN (True Negative)

3.3 实验设计与结果分析

实验中的软件缺陷数据集不平衡程度较高,MC1 数据集的不平衡率最高,为 138.21,KC1 数据集的不

平衡率最低,为 5.48。为降低不平衡率且最大限度的保持原始数据分布,设置欠采样后的不平衡率降为 5,过采样的邻域值 K 取 5。

为评估 RUS-RSMOTE 混合采样方法在不平衡数据集处理方面的有效性,将其与 RUS[19]、SMOTE[20]、RSMOTE[18]、RUS-SMOTE[21]进行对比,选用 CM1、KC1、KC3、MC1、MW1、PC2 作为实验数据集,在经过采样处理后的新数据集上使用决策树(J48)构建软件缺陷预测模型,为保证客观性,所有实验采用十折交叉验证进行,选用 F-value、AUC 和 G-mean 作为验证 RUS-RSMOTE 混合采样方法在软件缺陷预测中的有效性的评价指标。

图 4～6 是 6 个软件缺陷数据集经过 5 种采样方法处理后的 F-value、AUC、G-mean 评价指标对应的柱状图。由图 4 可见,所有数据集经过 RUS-RSMOTE 算法处理后的 F-value 值普遍高于其他采样算法,在数据集 KC3 上尤为明显。由图 5、6 可见,除数据集 CM1 外,剩余 5 个数据集经过 RUS-RSMOTE 算法处理后的 AUC、G-mean 值的高于其他采样算法。

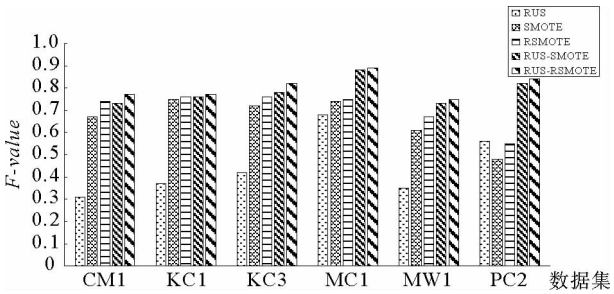


图 4 不同算法上的 F-value 值对比

Fig. 4 Comparison of F-value on different algorithms

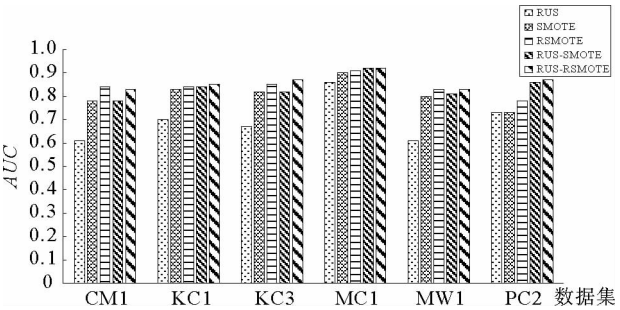


图 5 不同算法上的 AUC 值对比

Fig. 5 Comparison of AUC on different algorithms

综合三个评价指标来看,经过随机欠采样处理的不平衡数据集的性能最差,评价指标取值最低。RUS-RSMOTE 算法与其他采样算法相比,在不平衡软件缺陷数据集上的 F-value、AUC、G-mean 值更高,证实了 RUS-RSMOTE 算法对于处理不平衡的软件缺陷数据集的有效性。

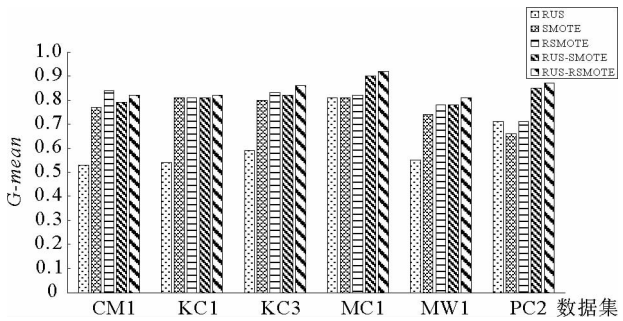
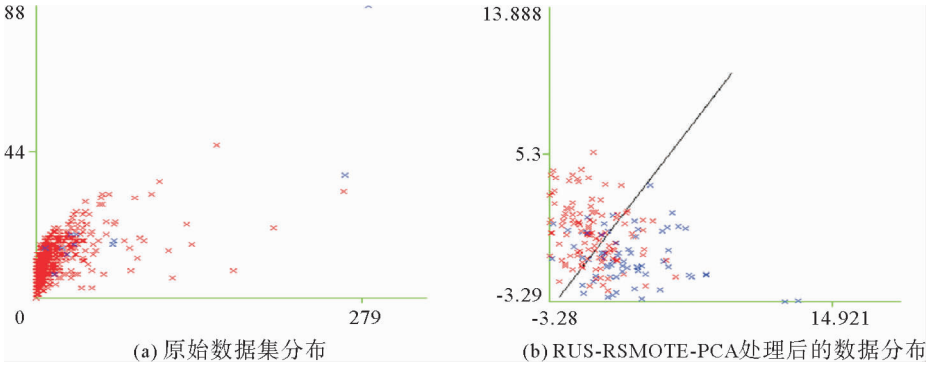


图 6 不同算法上的 G-mean 值对比

Fig. 6 Comparison of G-mean on different algorithms

PC2 数据集中某两种属性的数据分布如图 7 所示,图 7(a)表示原始数据集分布,图 7(b)表示在 RUS-RSMOTE 混合采样的基础上进行 PCA 特征降维后的数据分布。其中,红色标记表示无缺陷样本,蓝色标记表示有缺陷样本,可以看出图 7(a)中的无缺陷样本远远多于有缺陷样本,样本的不平衡程度非常高,无法找到合适的分割线对样本进行分类,图 7(b)的样本分布更适合分类器学习。

表 4 列出了经过 RUS-RSMOTE 混合采样和 PCA 降维后的 10 种数据集在朴素贝叶斯、决策树、支持向量机、K 最近邻 4 种机器学习算法上进行分类的 F-value、AUC、G-mean 指标对比,表格中加粗以及下划线表示的数据代表值最高。F-value、G-mean 值方面,K 最近邻算法在 10 个数据集上均取得最大值;AUC 值方面,K 最近邻算法在除 PC4 之外的 9 个数据集上取得最大值。比较 4 种分类算法在 F-value、AUC、G-mean 3 个指标上的平均值发现,K 最近邻算法的最大,决策树算法次之。综上所述,K 最近邻算法在软件缺陷数据集上的分类性能最好,决策树算法次之,朴素贝叶斯和支持向量机的分类性能相对较差。而支持向量机算法的最终决策函数只由少数的支持向量所确定,鲁棒性较强。因此,将 K 最近邻、决策树、支持向量机利用投票机制 Vote 进行异质集成。



(a) 原始数据集分布 (b) RUS-RSMOTE-PCA处理后的数据分布

图 7 PC2 数据集的二维散点图

Fig. 7 Two-dimensional scatter plots of PC2 dataset

表 4 不同算法上的评价指标对比

Tab. 4 Comparison of evaluation indicators on different algorithms

数据集	F-value				AUC				G-mean			
	NB	J48	SVM	kNN	NB	J48	SVM	kNN	NB	J48	SVM	kNN
CM1	0.44	0.72	0.34	0.84	0.78	0.78	0.58	0.88	0.56	0.77	0.47	0.88.
KC1	0.47	0.67	0.57	0.74	0.76	0.80	0.68	0.82	0.57	0.73	0.65	0.79
KC3	0.61	0.8	0.59	0.85	0.82	0.86	0.70	0.90	0.68	0.85	0.68	0.88
MC1	0.78	0.84	0.76	0.87	0.91	0.88	0.81	0.92	0.82	0.87	0.80	0.90
MW1	0.71	0.75	0.67	0.84	0.84	0.81	0.75	0.88	0.76	0.80	0.73	0.87
PC1	0.46	0.79	0.46	0.89	0.75	0.84	0.64	0.92	0.56	0.84	0.56	0.88
PC2	0.67	0.82	0.57	0.88	0.86	0.87	0.70	0.93	0.74	0.86	0.65	0.90
PC3	0.69	0.78	0.61	0.87	0.80	0.83	0.71	0.90	0.74	0.82	0.68	0.86
PC4	0.76	0.78	0.66	0.81	0.88	0.85	0.74	0.86	0.81	0.83	0.72	0.85
PC5	0.73	0.93	0.86	0.93	0.91	0.95	0.89	0.97	0.77	0.94	0.89	0.94
均值	0.63	0.79	0.61	0.85	0.83	0.85	0.72	0.90	0.70	0.83	0.68	0.88

为了验证 Vote 集成分类器的性能,将其与 Bagging、AdaBoostM1、RandomTree、RandomForest 集成分类器进行对比,比较这些集成分类器在软件缺陷预测性能方面的差异;为了更好地体现分类效果,同时与性能较好的单分类器 K 最近邻(KNN)进行比较。其中,Bagging、AdaBoostM1 为元分类器,使用 K 最近邻作为基分类器。为保证客观性,所有实验采用十折交叉验证进行,同样选用 F-value、G-mean 和 AUC 作为验证集成分类器在软件缺陷预测中的性能评价指标。

表 5 是 10 种软件缺陷数据集在 6 种分类算法上的 F-value 值对比。由表 5 可以看出:

1) 使用 kNN 作为基分类器的 Bagging、AdaBoostM1 在 F-value 值上几乎没有提高,应用 Bagging、AdaBoostM1 后的平均 F-value 值与 kNN 相等,甚至在应用 Bagging 后数据集 KC1、MW1、PC1、PC3 的 F-value 值反而略微降低。

2) Vote 在 9 个数据集上具有最高的 F-value 值,平均 F-value 值为 0.87;RandomForest 次之,值为 0.86。

表 6 是 10 种软件缺陷数据集在 6 种分类算法上的 AUC 值对比。由表 6 可以看出:

1) 使用 kNN 作为基分类器的 Bagging 在 AUC 值上有显著提升,而应用 AdaBoostM1 后数据集 KC1、KC3、MC1、PC1、PC2、PC5 的 AUC 值略微降低,平均 AUC 值也低于 kNN。

2) RandomForest 和 Vote 的 AUC 均值相等,值为 0.94,相比其他分类器,在软件缺陷预测方面具有明显优势。

表 5 不同算法上的 F-value 值对比

Tab. 5 Comparison of F-value on different algorithms

数据集	F-value					
	kNN	Bagging	AdaBoostM1	RandomTree	RandomForest	Vote
CM1	0.84	0.83	0.84	0.74	0.84	0.85
KC1	0.74	0.73	0.74	0.68	0.75	0.77
KC3	0.85	0.85	0.85	0.79	0.86	0.88
MC1	0.87	0.87	0.87	0.82	0.87	0.90
MW1	0.84	0.83	0.84	0.76	0.84	0.87
PC1	0.89	0.88	0.88	0.80	0.88	0.88
PC2	0.88	0.88	0.89	0.82	0.86	0.90
PC3	0.87	0.86	0.87	0.77	0.86	0.87
PC4	0.81	0.82	0.81	0.75	0.86	0.86
PC5	0.93	0.93	0.93	0.91	0.94	0.95
均值	0.85	0.85	0.85	0.78	0.86	0.87

表 6 不同算法上的 AUC 值对比

Tab. 6 Comparison of AUC on different algorithms

数据集	AUC					
	kNN	Bagging	AdaBoostM1	RandomTree	RandomForest	Vote
CM1	0.88	0.89	0.88	0.80	0.91	0.91
KC1	0.82	0.87	0.78	0.74	0.86	0.89
KC3	0.90	0.93	0.89	0.83	0.95	0.93
MC1	0.92	0.95	0.89	0.85	0.95	0.96
MW1	0.88	0.91	0.88	0.81	0.91	0.92
PC1	0.92	0.95	0.92	0.84	0.96	0.93
PC2	0.93	0.95	0.92	0.87	0.95	0.94
PC3	0.90	0.94	0.90	0.82	0.94	0.95
PC4	0.86	0.92	0.85	0.80	0.96	0.95
PC5	0.97	0.98	0.95	0.92	0.99	0.97
均值	0.90	0.93	0.89	0.83	0.94	0.94

表 7 是 10 种软件缺陷数据集在 6 种分类算法上的 G-mean 值对比。通过观察可以得到如下结论：

1) 对比 Bagging、AdaBoostM1 和 kNN 发现，Bagging、AdaBoostM1 几乎没有贡献，与单分类器的平均 G-mean 值相等。

2) 与 AUC 值情况一样，RandomForest 和 Vote 在 G-mean 上的均值相等，值为 0.89，相比其他分类器，这两种集成算法在软件缺陷预测方面具有明显优势。

图 8～10 是采用 Vote、kNN 算法在 10 个数据集上的 F-value、AUC、G-mean 对比。从图 8 可以看出，采用 Vote 进行分类的数据集中除 PC1 外，其他 9 个数据集的 F-value 值普遍高于 kNN，在数据集 PC4 上尤为明显。从图 9 可以看出，两者在 PC5 数据集上的 AUC 值都是 0.97，除此之外，采用 Vote 进行分类的 AUC 值明显优于 kNN。图 10 的柱状图显示两者在 3 个数据集上的 G-mean 值相等，采用 Vote 进行分类的 6 个数据集的 G-mean 值高于 kNN。综上所述，采用 Vote 集成 K 最近邻、决策树、支持向量机的分类器性能远远超过个体分类器 kNN。

表 7 不同算法上的 G-mean 值对比

Tab. 7 Comparison of G-mean on different algorithms

数据集	G-mean					
	kNN	Bagging	AdaBoostM1	RandomTree	RandomForest	Vote
CM1	0.88	0.87	0.88	0.80	0.88	0.89
KC1	0.79	0.78	0.79	0.75	0.80	0.82
C3	0.88	0.89	0.88	0.83	0.89	0.88
MC1	0.90	0.90	0.90	0.86	0.91	0.91
MW1	0.87	0.87	0.87	0.81	0.87	0.88
PC1	0.88	0.88	0.88	0.84	0.91	0.88
PC2	0.90	0.91	0.92	0.86	0.89	0.89
PC3	0.86	0.86	0.90	0.82	0.89	0.90
PC4	0.85	0.86	0.86	0.80	0.89	0.85
PC5	0.94	0.95	0.95	0.93	0.96	0.95
均值	0.88	0.88	0.88	0.83	0.89	0.89

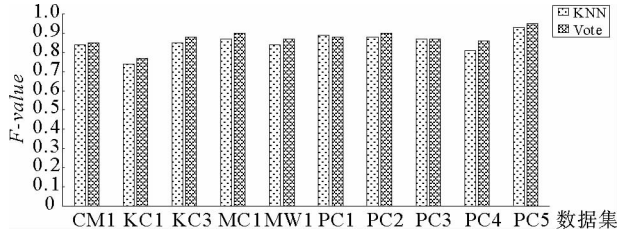


图 8 kNN 和 Vote 的 F-value 值对比

Fig. 8 Comparison of F-value between kNN and Vote

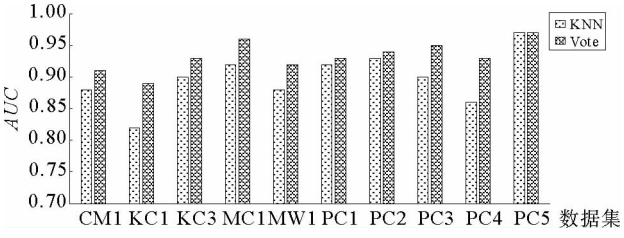


图 9 kNN 和 Vote 的 AUC 值对比

Fig. 9 Comparison of AUC values between kNN and Vote

综合 6 种分类算法在 F-value、AUC、G-mean 3 项指标的表现来看,将 K 最近邻、决策树、支持向量机利用投票机制 Vote 进行异质集成的分类器在软件缺陷预测方面具有显著的性能优势。

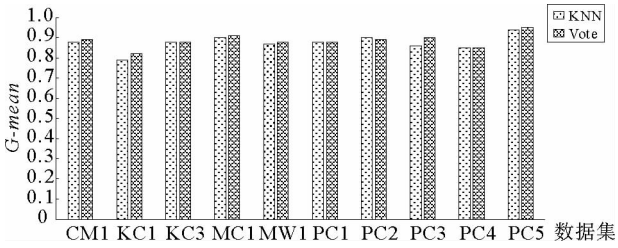


图 10 kNN 和 Vote 的 G-mean 值对比

Fig. 10 Comparison of G-mean values between kNN and Vote

4 总结与展望

为解决软件缺陷预测中存在的数据不平衡、特征维度高以及预测精度低等问题,提出了一种基于 RUS-RSMOTE-PCA-Vote 的软件缺陷不平衡数据分类方法,首先通过随机欠采样来减少无缺陷样本的数量,然后进行 SMOTE 过采样,在过采样中综合总体样本的分布状况引入影响因素 posFac 指导新样本的合成,对经过 RUS-RSMOTE 混合采样处理后的数据集进行 PCA 降维,最后应用 Vote 组合 K 最近邻、决策树、支持向量机构造集成分类器。实验结果表明,所提方法可以有效地解决软件缺陷预测中存在的数据不平衡、特征维度高以及预测精度低等问题。

由于综合考虑了软件缺陷数据存在的数据不平衡、特征维度高以及预测精度低等问题,因此本方法在时间复杂度上稍高于其他方法,接下来将探究不同运行参数对于软件缺陷预测的分类性能影响并提高运行速度。同时结合机器学习、深度学习的新技术进行不平衡数据处理,例如探讨生成对抗网络(generative adversarial networks, GAN)在数据扩充方面的应用等。

参考文献:

[1]于巧.基于机器学习的软件缺陷预测方法研究[D].徐州:中国矿业大学,2017.

YU Qiao.Research on software defect prediction method based on machine learning[D].Xuzhou:China University of Mining and Technology,2017.

[2]PATEL H,THAKUR G S.An improved fuzzy k-nearest neighbor algorithm for imbalanced data using adaptive approach [J].IETE Journal of Research,2018:1-10.

[3]CHABBOUH M,BECHIKH S,HUNG C C,et al.Multi-objective evolution of oblique decision trees for imbalanced data binary classification[J].Swarm and Evolutionary Computation,2019,49:1-22.

[4]LIU M,MIAO L,ZHANG D.Two-stage cost-sensitive learning for software defect prediction[J].IEEE Transactions on Reliability,2014,63(2):676-686.

[5]RODRiGUEZ,DANIEL,HERRAIZ I,et al.Preliminary comparison of techniques for dealing with imbalance in software defect prediction[C]// International Conference on Evaluation & Assessment in Software Engineering.ACM,2014.

[6]MALHOTRA R,KAMA S.An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data[J].Neurocomputing,2019,343:120-140.

[7]SIERS M J,ISLAM M Z.Software defect prediction using a cost sensitive decision forest and voting,and a potential solution to the class imbalance problem[J].Information Systems,2015,51:62-71.

[8]张忠林,吴挡平.基于概率阈值 Bagging 算法的不平衡数据分类方法[J].计算机工程与科学,2019,41(6):1086-1094.
ZHANG Zhonglin,WU Dangping.An imbalanced data classification method based on probability threshold Bagging[J].Computer Engineering and Science,2019,41(6):1086-1094.

[9]YUAN Z,ZHAO P.An improved ensemble learning for imbalanced data classification[C]// IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC).Chongqing,China,2019:408-411.

[10]LU C,CHUNG I,Lin T.The hybrid dynamic prototype construction and parameter optimization with genetic algorithm for support vector machine[J].International Journal of Engineering & Technology Innovation,2015,5(4):220-232.

[11]HU F,WANG L,Zhou Y.An oversampling method for imbalance data based on three-way decision model[J].Acta Electronica Sinica,2018,46(1):135-144.

[12]刘小花,王涛,吴振强.软件缺陷集成预测模型研究[J].计算机应用研究,2013,30(6):1734-1738.
LIU Xiaohua,WANG Tao,WU Zhenqiang.Software defect prediction based on classifiers ensemble[J].Application Research of Computers,2013,30(6):1734-1738.

[13]SUN Z,SONG Q,ZHU X,et al.A novel ensemble method for classifying imbalanced data[J].Pattern Recognition,2015,48(5):1623-1637.

[14]PENG J,AVED A J,SEETHARAMAN G,et al.Multiview boosting with information propagation for classification[J].IEEE Transactions on Neural Networks & Learning Systems,2018,29(3):657-669.

[15]BREIMAN L.Random Forests[J].Machine Learning,2001,45(1):5-32.

[16]古平,欧阳源遊.基于混合采样的非平衡数据集分类研究[J].计算机应用研究,2015,32(2):379-381.
GU Ping,OUYANG Yuanyou.Classification research for unbalanced data based on mixed-sampling[J].Application Research of Computers,2015,32(2):379-381.

[17]袁梅宇.数据挖掘与机器学习[M].北京:清华大学出版社,2014.
YUAN Meiyu.Data mining and machine learning[M].Beijing:Tsinghua University Press,2014.

[18]魏浩,李红,刘小豫.一种改进的 SMOTE 算法[J].河南科学,2018,36(7):1009-1013.
WEI Hao,LI Hong,LIU Xiaoyu.An improved SMOTE algorithm[J].Science of Henan,2018,36(7):1009-1013.

[19]TAHIR M A,KITTLER J,MIKOLAJCZYK K,et al.A multiple expert approach to the class imbalance problem Using Inverse Random under Sampling[J].2009,10(2):82-91.

[20]CHAWLA N V,BOWYER K W,HALL L O,et al.SMOTE:Synthetic minority over-sampling technique[J].Journal of Artificial Intelligence Research,2002,16(1):321-357.

[21]ZHANG X,ZHUANG Y,HU H,et al.3-D laser-based multiclass and multiview object detection in cluttered indoor scenes [J].IEEE Transactions on Neural Networks & Learning Systems,2015,28(1):177-190.

(责任编辑:傅 游)