类不平衡对软件缺陷预测模型稳定性和预测性能的影响分析方法

张艳梅[1,2],植胜林[3],姜淑娟[1,2],袁 冠[1,2]
(1. 中国矿业大学矿山数字化工程研究中心,江苏徐州 221116;2. 中国矿业大学计算机科学与技术学院,江苏徐州 221116;
3. 科华数据股份有限公司,广东深圳 518055)

摘 要: 本文提出一种类不平衡对软件缺陷预测模型稳定性和预测性能的影响分析方法. 首先,使用欠采样方法将原数据集构造成一组不平衡率小于原数据集本身不平衡率的新数据集. 其中,在构造数据集时使用固定种子,保证同一个数据集构造的同一个不平衡率的数据集中的数据相同,以减少每次运行结果的随机性. 其次,以MCC值作为预测模型的性能评价指标,将每次产生的新数据集放入模型中的分类算法进行训练预测评价,获得当前数据集不同不平衡率下的MCC值,并提出稳定性评价指标. 实验结果表明:与AUC相比,MCC更适合作为类不平衡情况下软件缺陷预测模型稳定性的评价指标;对于软件缺陷预测性能稳定性,代价敏感模型表现优于集成模型.

关键词: 类不平衡;缺陷预测;稳定性;预测性能;评价指标
基金项目: 国家自然科学基金(No.61673384, No.71774159);中国博士后科学基金特别资助(No.2021T140707)
中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112(2023)08-2076-12
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210911

Influence Analysis Method of Class Imbalance on Software Defect Prediction Model Stability and Prediction Performance

ZHANG Yan-mei[1,2], ZHI Sheng-lin, JIANG Shu-juan[1,2], YUAN Guan[1,2]
(1. *Mine Digitization Engineering Research Center of the Ministry of Education, China University of Mining and Technology, Xuzhou, Jiangsu* 221116, *China*;
2. *School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu* 221116, *China*;
3. *KeHua Data CO., LTD, Shenzhen, Guangdong* 518055, *China*)

Abstract: The paper proposes a method for analyzing the influence of class imbalance on software defect prediction model stability and prediction performance. Firstly, the original data set is constructed into a set of new data sets whose unbalance rate is less than the original data set's unbalance rate by using the undersampling method. Where, fixed seeds are used in the construction of the data set to ensure that the data in the same unbalanced rate data set constructed by the same data set is the same, so as to reduce the randomness of the results of each run. Secondly, the MCC value is taken as the performance evaluation indicator of the prediction model, and the new data set generated each time is put into the classification algorithm of the model for training and prediction evaluation, so as to obtain the MCC value at different unbalanced rate for the current data set. We also propose a performance stability evaluation indicator. The experimental results show that, MCC is more suitable as the stability evaluation indicator of software defect prediction model under the condition of class imbalance compared with AUC. For the stability of software defect prediction performance, the cost sensitive model performs better than the ensemble model.

Key words: class imbalance; defect prediction; stability; prediction performance; evaluation indicator
Foundation Item(s): National Natural Science Foundation of China (No.61673384, No.71774159); Postdoctoral Foundation of China (No.2021T140707)

收稿日期:2021-07-14;修回日期:2022-08-17;责任编辑:王天慧

1 引言

软件测试作为软件工程的重要步骤之一,对减少软件缺陷和提高软件质量具有极为重要的意义.要提高软件质量,减少软件设计缺陷、开发和维护成本,就需要更有效的手段及时发现软件中的缺陷,因此软件缺陷预测应运而生.

软件缺陷预测技术是通过机器学习[1]的理论和算法,从软件历史数据中获取有用的软件缺陷信息,将挖掘出的缺陷信息分布模式应用于预测未知软件模块中是否会出现缺陷.

研究发现,在实际应用中分类不平衡问题普遍存在[2],其中包括软件缺陷预测.类不平衡问题是指样本内各类别数量分布不均匀的现象.软件缺陷数据的分布是高度倾斜的,往往由少数有缺陷模块和多数无缺陷模块组成.传统学习模型在有类不平衡的数据集中学习效果比较差[3],原因是大多数机器学习算法的设计是假设分类平衡或者错分类代价相等[4],无法正确表示数据不均匀的分布特征.

在实际应用中,分类模型使用场景不仅是训练集和测试集的样本分布不同,其样本特征也不一定相同.因此训练出来的分类模型往往用于跨项目缺陷预测和跨公司缺陷预测[5],分类模型的预测性能代表着分类结果的好坏,而稳定性代表着训练出来的模型在进行跨项缺陷预测和跨公司缺陷预测中可能表现出的分类能力.因此,在跨项目缺陷预测和跨公司缺陷预测时,当训练模型的稳定性较高时,就越容易得到符合预期的分类效果;相反,模型的稳定性越差,意味着分类结果不稳定,难以得到符合预期的分类效果,从而导致对模型的效果评价不一致.可以看出,稳定性也严重影响着分类模型的实际应用.因此,在评估分类模型的预测性能时,稳定性也是需要考虑的因素.此外,一方面,虽然类不平衡问题解决方法能够提高预测模型性能,但是这些方法是否对各个预测模型的预测性能都起着积极的作用仍有待商榷;另一方面,预测模型本身可能会受到类不平衡的影响,如果预测模型的性能稳定性好,那么即便受到噪声数据的干扰,仍能保持较好的预测效果.因此,如果能够了解不同预测模型在类不平衡时的稳定程度以及各分类算法在类不平衡时的预测性能,则可在实际应用中有针对性地选择合理的预测模型,从而更好地指导软件缺陷预测.

2 相关工作

目前,许多研究人员在类不平衡问题的解决方面取得了一定的成果.类不平衡问题解决方法主要包括采样法[6,7]、代价敏感学习[8,9]和集成学习[10,11].类不平衡问题解决的关键要点在于数据集的处理、机器学习方法的选择以及合理的评价指标.

在数据集处理方面,Gao等人[12]将采样法和特征选择方法相结合,以解决类不平衡问题和高维问题.在代价敏感学习方面,Michael等人[8]提出结合代价敏感的决策树方法,并证明了其在处理类不平衡对软件缺陷预测性能的影响上有不错的效果.在集成学习方面,Laradji等人[13]将集成学习与特征选择结合,并证明其效果较单一模型效果更好.Wang等人[14]比较了重采样、阈值移动的代价敏感学习和集成学习三种技术,发现集成学习性能更具优势.Wahono等人[15]将粒子群优化(Particle Swarm Optimization, PSO)和袋装技术(Bagging)结合,PSO用于特征选择,Bagging用于处理类不平衡问题.Malhotra等人[16]对采用各种基于boosting的集成方法开发的软件缺陷预测模型进行了实证比较.结果表明,与经典的boosting模型相比,在使用集成方法进行分类前使用重采样技术显著改善了模型预测的性能.为了进一步提高在采样过程中新合成的样本质量,王子健[17]首先对数据中的异常值进行检测,并与原本存在的缺失值统一用均值进行填充,从而避免新生成的样本点围绕原数据集中的噪音点而建立,进一步采用欠采样与过采样相结合的思想,重点关注分布在分类决策边界上的类别重叠问题.通过对每个样本的最近邻少数类周围的多数类样本进行清洗,进而提高了数据质量,从而实现数据集的总体平衡.

分类算法主要有人工神经网络(Artificial Neural Network)、贝叶斯网络(BayesNet)、决策树(C4.5)、随机森林(Random Forest, RF)、逻辑回归(Logistic Regression, LR)和支持向量机(Support Vector Machine, SVM)等.Lessmann等人[18]对22个分类算法在多个软件缺陷数据集上的预测性能进行了比较,发现这些算法都有较优的效果,这是由于实验数据集特征、预测模型参数和评价指标的设置,使模型之间效果差异并不明显.

在类不平衡处理对软件缺陷预测的性能影响这一领域中,于巧[5]从预测模型性能稳定性出发,分析了类不平衡对软件缺陷预测性能的影响,并对不稳定的预测模型使用代价敏感学习和集成学习,验证了这两种方法可以提高预测模型的稳定性.Song等人[19]从软件度量元、类不平衡处理方法、分类算法、评价指标等方面,综合研究了类不平衡处理在软件缺陷预测中的作用,发现只有52%的不平衡算法和预测模型结合显著地提升了性能.Balogun等人[20]改进了于巧[5]的方法,评估预测模型在软件缺陷预测中的性能稳定性.首先,实验结果表明,分类不平衡对预测模型的性能有负面影响,过采样方法(Synthetic Minority Oversampling TEchnique,SMOTE)提高了预测模型的性能.其次,平衡数据集的过采样方法优于欠采样方法,欠采样方法

由于数据集中有用实例的随机删除因而性能较差. 最后,在使用的预测模型中,Logistic 回归、Naïve 贝叶斯和随机森林各自的变异系数(Coefficient of Variation,CV)值是更稳定的预测模型,在不平衡的数据集上也能很好地工作. Eldho[21]提出一个非平衡数据分类模型,将数据集分成两个新的数据集,并通过预测模型对创建的不平衡数据集进行测试. 该方法从PROMISE库中提取不平衡缺陷数据集,用于性能评估. 实验结果表明,现有的K近邻(K-Nearest Neighbor,KNN)、朴素贝叶斯(Naive Bayesian,NB)和反向传播(Back Propagation,BP)三种预测分类器模型的性能容易受到类不平衡的影响,而支持向量机(Support Vector Machine,SVM)和极限学习机(Extreme Learning Machine,ELM)更稳定.

通过以上研究发现:大多研究在评价预测模型的预测性能时忽略了稳定性这一指标;评价预测模型的稳定性时,缺少更加合理的评价指标.

本文的贡献如下:

(1)在评价预测模型的预测性能时同时也考虑稳定性这一指标;

(2)对软件缺陷预测性能影响分析方法进行改进,并提出软件缺陷预测性能稳定性的评价指标以及预测性能指标.

3 相关知识

本部分介绍类不平衡处理方法的相关知识,包括软件度量、类不平衡处理方法、常用的分类算法和评价指标等.

3.1 软件度量

软件度量是量化软件缺陷数据时的指标,也常把度量元描述为软件特性,也称特征. 常用的软件缺陷度量有面向对象的CK度量(Chidamber and Kemerer,CK)、社会网络分析度量(Social Network Analysis,SNA)和过程度量(Process,PROC). Song等人[19]研究发现输入度量的不同对预测性能带来的影响并不显著. 因此,本文并不关注数据集的输入度量选择,统一使用CK度量作为输入度量.

3.2 类不平衡处理方法

为了解决类不平衡问题,研究者们提出了许多解决类不平衡的方法. 解决类不平衡问题的方法中采样法是对数据集的优化,代价敏感学习和集成学习则是对分类算法的优化[5]. 采样法从数据的角度提出解决方案,而代价敏感学习和集成学习则从算法的角度解决问题.

3.3 典型分类算法

在软件缺陷预测中,机器学习充当着重要的角色. 为了研究类不平衡问题对软件缺陷预测模型性能的影响,本文使用6个经典机器学习分类算法:决策树、K近邻、逻辑回归、朴素贝叶斯、随机森林和SVM序列最小优化算法[5]. 表1给出了算法的英文名称(缩写)、中文名和WEKA中对应的分类算法类别.

表1 分类算法简介

编号	英文名称(缩写)	中文名称	分类算法类别
1	C4.5	C4.5决策树	trees
2	K-Nearest Neighbor(KNN)	K近邻	lazy
3	Logistic Regression(LR)	逻辑回归	functions
4	Naïve Bayes(NB)	朴素贝叶斯	bayes
5	Random Forest(RF)	随机森林	trees
6	SMO	SVM序列最小优化算法	functions

3.4 评价指标

虽然随着分类结果确定,模型的稳定性和预测性能也随之确定,但是合理的评价指标能够合理体现实验结果的准确性. 分类性能通常基于混淆矩阵. 在二分类中表示4种可能的结果,如表2所示.

表2 二分类混淆矩阵

	样本为正例	样本为负例
预测为正例	TP	FP
预测为负例	FN	TN

一个好的评价指标可以准确判断构建的预测模型的好坏. 评价指标通常分为3类:缺陷倾向评价指标、缺陷数评价指标和缺陷严重度评价指标[22]. 本文研究的软件缺陷预测技术是指预测模块有无缺陷(即缺陷倾向问题),属于二分类问题. 在二分类问题中常用的性能评价指标主要有AUC(ROC曲线下面积)和MCC.

3.4.1 AUC(ROC曲线下方面积)

AUC[5]的定义是随机给定一个正样本和负样本,分类器输出正样本预测为正的概率高于分类器输出负样本预测为正的概率的可能性. AUC又被称为计算ROC曲线下方的面积. 在比较预测模型时,一条ROC曲线上的点必须完全在另一条ROC曲线上方,才认为前者优于后者.

3.4.2 MCC(Matthews相关系数)

MCC是一个针对二分类的性能指标,特别是在类别间数量不相等时,MCC定义[19]为

$$\mathrm{MCC}=\frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}\quad(1)$$

MCC衡量的是预测值和实际值间的关系,取值为[–1,1]. 取值为1时表示完全正相关(即完全预测),取值为0时表示无关联(随机预测),取值为–1时表示完全负相关.

4 分类不平衡对软件缺陷预测性能影响的分析方法

为了研究类不平衡数据对缺陷预测模型性能的影响程度,本文在文献[5]提出的分类不平衡影响分析方法的基础上进行改进.

4.1 方法描述

分类不平衡对软件缺陷预测性能影响的分析方法流程图如图1所示,对应的算法如算法1所示.

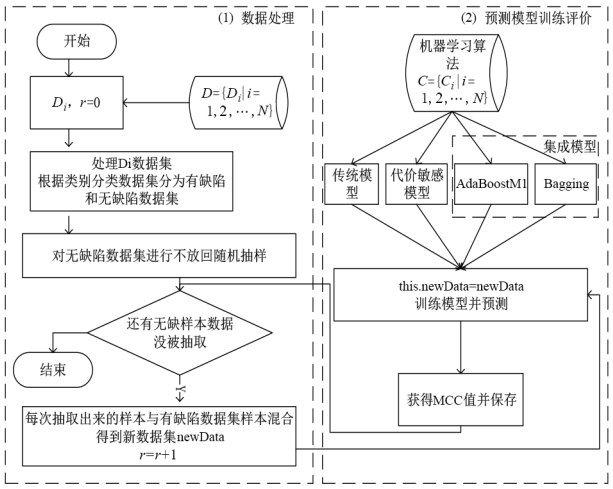


图1　分类不平衡对软件缺陷预测性能影响的分析方法流程图

类不平衡对软件缺陷预测性能影响的分析算法主要包含3个步骤.

(1)构造数据集:使用欠采样方法将原数据集构造出一组不平衡率为1,2,…,r(r为原数据集的不平衡率)的新数据集. 与文献[5]不同的是,本文在构造数据集时使用固定的种子,以保证同一个数据集构造出的同一个不平衡率的数据集中的数据相同,以减少每次运行结果的随机性.

(2)将产生的一组新数据集分别放入模型中的分类算法进行训练预测评价,假设数据不平衡率为r,可以构造r个新数据集,这样一个模型将得到r个不同不平衡率下MCC的值,获得一组对应的MCC值,以MCC值作为预测模型性能指标.

(3)循环步骤(1)和(2),直至所有的模型、分类算法和数据集都经过组合使用,训练测试并获得对应的MCC值.

本实验选取6个经典的分类算法(C4.5,KNN,LR,NB,RF,SMO)以及传统模型(不进行任何处理,直接使用分类算法)、代价敏感模型和集成模型三大模型,以"模型+分类算法"的组合方式区别分类算法有无经过不平衡学习算法处理. 具体地,将传统模型(Traditional,T)、代价敏感模型(Cost sensitive,C)、集成模型(AdaBoostM1,A)、集成模型(Bagging,B)分别与分类算法进行组合. 因此,本实验可以产生$\sum_{i=1}^{6}r_i\times4\times6=3\,816$个MCC值,每一个MCC值都经由10次十折交叉验证法的平均得到.

算法1　分类不平衡对软件缺陷预测性能影响的分析算法

输入: 模型数组arrayModel={传统模型,代价敏感模型,集成模型AdaBoostM1,集成模型Bagging},机器学习算法array$C$={C4.5,KNN,LR,NB,RF,SMO},原不平衡数据集arrayData={jedit-3.4,lucene,MC1,PC1,PC2,tomcat}

输入: arrayModel[i]+ arrayC[j]+ arrayData[k]下的r个MCC值,不平衡率r

```
FOR(int i=0 ; i<arrayModel.length; i++)
    FOR(int j=0 ; j<arrayC.length; j++)
        FOR(int k=0 ; k<arrayData.length; k++)
        //初始化资源
        根据数据集的类别将arrayData[k]分为DefectiveData和NonDefectiveData
        arrayNewData[0]=DefectiveData
resNondefectiveData=NonDefectiveData
        arrayMCC=null
        //构造不平衡率从1,2,…的一组不平衡数据集保存在arrayNewData中
        FOR(int r=1; resNondefectiveData != null; r++)
        使用固定种子重排过滤器将resNondeffectiveData打乱
        IF resNondefectiveData.size >= 2*DefectiveData.size
            从resNondeffectiveData提取DefectiveData.size个样本到arrayNewData[r−1]中
            将提取的样本从resNondeffectiveData中移除
        ELSE
            将resNondeffectiveData剩余样本全部提取至arrayNewData[r−1]中
            将提取的样本从resNondeffectiveData中移除
        END IF
        IF r−1 ≥ 1
            arrayNewData[r−1]= arrayNewData[r−1]+ arrayNewData[r−2]
        END IF
        //将arrayNewData中的数据集放入到arrayModel[i]+arrayC[j]分类算法中训练测试并获取一组MCC值
        FOR(int p=0; p<arrayNewData.length; p++)
            将arrayNewData[p]数据集放入arrayModel[i]+arrayC[j]训练测试和评估获得MCC值
            将MCC存入arrayMCC[p]
        输出arayyMCC中的值
```

由于同一数据集中不同预测模型的平均性能是有差异的,直接使用标准差去衡量预测模型性能的离散程度并不合理[5]. 因此,本文使用变异系数(Coefficient of Variation,CV)[23](标准差和平均值的比值)获得预测模型的稳定性信息,消除平均值差异对数据离散程度的影响,并获得离散程度较大的预测模型信息. 变异系数对比的是数据组之间的相对变异程度,并没有固定的阈值. 在此,我们选取适合本实验的评价准则:当变异系数小于0.1时属于弱变异;当变异系数在0.1~1.0时属于中等变异;当变异系数大于1.0时为强变异. 预测模型性能离散程度很大,预测模型性能表现很不稳定. 因此,首先收集每个"模型+分类算法"的一组值($MCC=\{MCC_i|i=1,2,\cdots,r\}$),再统计其平均值(Average,AVG)、标准方差(Standard Deviation,STD)和变异系数(Coefficient of Variation,CV),计算公式为

$$AVG=\frac{\sum_{i=1}^{r}MCC_i}{r} \tag{2}$$

$$STD=\sqrt{\frac{\sum_{i=1}^{r}(MCC_i-AVG)^2}{r}} \tag{3}$$

$$CV=\frac{STD}{AVG}\times 100\% \tag{4}$$

4.2 分类算法和预测模型参数设置

由于在weka中进行预测模型训练时一般会使用分类算法默认参数进行实验,本实验使用的分类算法中的参数值采用weka中默认的参数值.

传统模型不经任何不平衡学习方法处理的分类算法,直接对分类算法进行训练评估. 代价敏感模型实验中,我们将FN代价设为FP的10倍,TP和TN代价为0,得到代价矩阵$\boldsymbol{C}$(Cost Matrix),矩阵表示为

$$\boldsymbol{C}=\begin{bmatrix}0 & 10\\ 1 & 0\end{bmatrix} \tag{5}$$

为了探究集成模型在分类不平衡时的性能稳定程度,集成模型实验选择了结合代价敏感思想的集成模型AdaBoostM1,以及常用的经典集成模型Bagging. 因此,可以将代价敏感模型和集成模型的稳定性进行比较,来研究代价敏感模型和集成模型对传统模型的稳定性的提升效果.

5 实验

本文的目的是研究类不平衡情况下,代价敏感学习和集成学习对传统模型预测性能和稳定性的影响.

5.1 研究问题

为了分析类不平衡对软件缺陷预测性能的影响,本文提出以下问题:

问题(1):在类不平衡情况下,传统模型的稳定性如何?

问题(2):类不平衡情况下,代价敏感学习是否对传统模型的稳定性和预测性能带来提升?

问题(3):类不平衡情况下,集成学习是否对传统模型的稳定性和预测性能带来提升?

问题(4):在类不平衡情况下,代价敏感模型和集成模型分别对传统模型的稳定性和预测性能带来怎样的提升?

5.2 实验对象

由于不平衡学习方法对低不平衡率软件缺陷数据集预测性能提升并没有明显的效果[19],本实验选取60个软件缺陷数据集中不平衡率在中高水平的6个典型的软件缺陷数据集,而没考虑低不平衡水平的数据集. 其中,选取的PC2,jedit-3.4和MC1这3个高度不平衡数据集的不平衡率在整个软件缺陷数据集不平衡率中位数的10倍以上($>10\times 2^{1.79}\approx 35$);C1,tomcat和lucene这3个中度不平衡的数据集的不平衡率是整个软件缺陷数据集不平衡率中位数的3倍($3\times 2^{1.79}\approx 10$). 数据集信息如表3所示.

表3 软件缺陷数据集信息

数据集	特征数	样本总数	有缺陷样本	无缺陷样本	不平衡率
PC2	36	745	16	729	45
jedit-4.3	20	492	11	481	43
MC1	38	1 988	46	1 942	42
PC1	37	705	61	644	10
tomcat	20	858	77	781	10
lucene	7	691	64	627	9

5.3 实验结果与分析

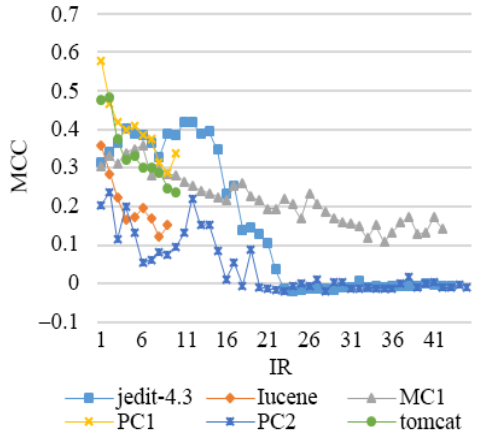
5.3.1 传统模型的稳定性

对于"问题(1):在类不平衡情况下,传统模型的稳定性如何?",我们从获取到的60个软件缺陷数据集中选取了6个不平衡率在中高水平的数据集. 步骤如下:首先将数据集构造为不平衡率递增的新数据集;然后对新数据集进行分类训练,得到预测模型;最后,计算"传统模型+分类算法"在各数据集上训练得到的预测模型的MCC值和变异系数,按分类算法划分成6组,通过变异系数分析预测模型的稳定性.

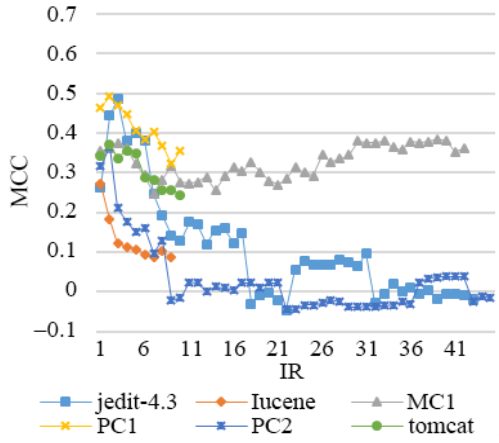
按照实验设计的步骤,首先构造一组不平衡率递增的数据集;然后,将这组数据集分别放到"传统模型+对应分类算法",训练预测获得评价结果;最后,得到一组随不平衡率递增的MCC值. 各传统预测模型的预测性能按数据集划分统计在折线图中,如图2所示. 其中,横轴表示不平衡率(Imbalance Ratio,IR),纵

轴表示MCC值.

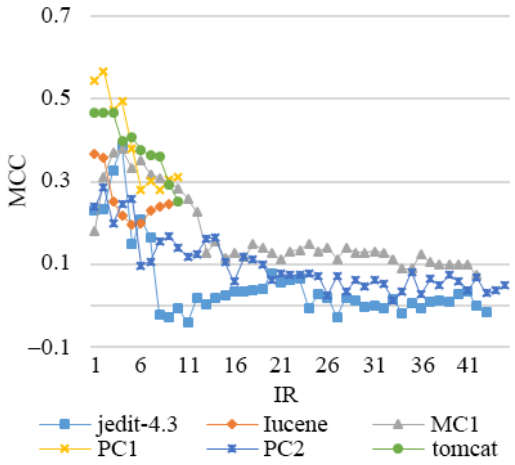
由图2可以看出:“传统模型+SMO”随着不平衡率IR的增大分类性能下降非常明显,在不平衡率IR到达10时对所有的软件缺陷数据集分类基本没有效果,接近随机预测值.“传统模型+C4.5”“传统模型+KNN”“传统模型+LR”和“传统模型+RF”这4个分类算,随着



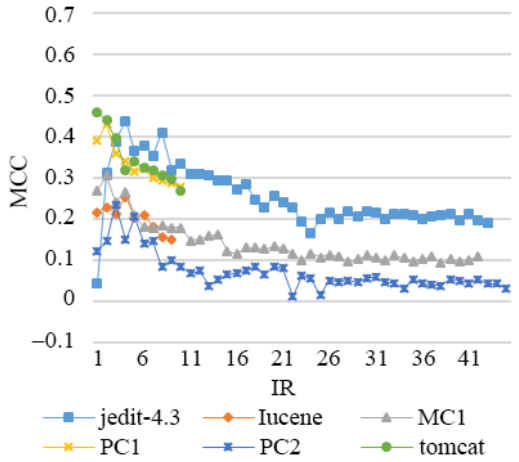
(a) 传统模型+C4.5



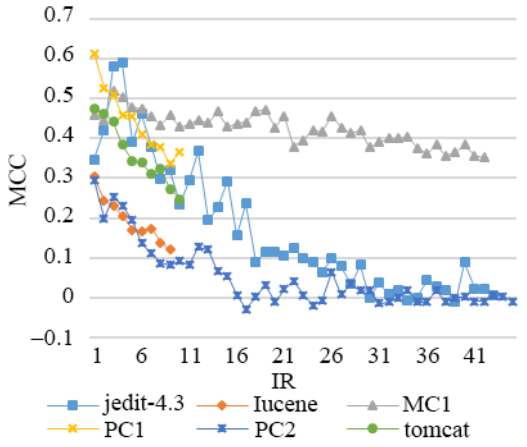
(b) 传统模型+KNN



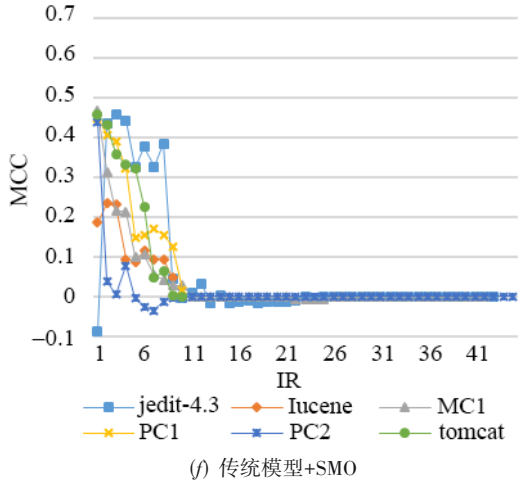
(c) 传统模型+LR



(d) 传统模型+NB



(e) 传统模型+RF



(f) 传统模型+SMO

图2　传统模型+分类算法对分类不平衡的敏感性比较图

不平衡率增大,预测能力都明显下降.当不平衡率IR大于20后,3个高不平衡率的数据集在“传统模型+C4.5”训练预测下,有2个数据集已经接近随机预测,没有正向预测.而在其他3种“传统模型+分类算法”情况中,数据集

还是会出现正向预测的情况.

为了进一步比较“传统模型+各分类算法”的稳定性,我们通过计算变异系数的值并从该值中获得的分类算法的稳定性信息,计算“传统模型+各分类算法”在数据集上的MCC的平均值,标准差和变异系数.为了便于查看使用MCC评价指标时传统分类算法的总体稳定性效果,我们将各个传统分类算法在数据集上MCC的变异系数求平均值,得到如表4所示的“传统模型+各分类算法”在各数据集上MCC预测性能的变异系数.当某个预测模型的变异系数的值大于1时,采用“加粗”标记.

表4 “传统模型+各分类算法”在各数据集上MCC的变异系数

算法	jedit-4.3	lucene	MC1	PC1	PC2	tomcat	平均值
C4.5	**1.168 7**	0.340 8	0.305 7	0.197 2	**1.664 4**	0.239 4	0.652 7
KNN	**1.275 3**	0.440 0	0.125 2	0.126 8	**2.855 3**	0.146 5	0.828 2
LR	**1.868 8**	0.233 0	0.516 0	0.277 0	0.680 8	0.181 3	0.626 2
NB	0.294 3	0.155 6	0.365 1	0.141 0	0.623 8	0.173 6	0.292 2
RF	0.982 0	0.277 9	0.096 1	0.183 7	**1.559 5**	0.207 7	0.551 2
SMO	**2.453 0**	0.486 7	**2.605 5**	0.590 4	**6.196 3**	0.762 7	**2.182 4**

由表4可以看出:在高度不平衡数据集上,“传统模型+SMO”的变异系数都大于1,说明SMO的稳定性很差.传统模型与C4.5,KNN,LR和RF的组合在一些数据集上的变异系数大于1,说明预测模型性能表现并不是很稳定.“传统模型+NB”在高度不平衡数据集中的变异系数依旧小于1,表现相对稳定.在中度不平衡数据集上,基本所有“传统模型+所有分类算法”的变异系数都为0.1~1.0,表明中度的类不平衡水平对“传统模型+所有分类算法”的稳定性有影响,但是没有高度类不平衡水平的影响那么大.总体上看“传统模型+NB”的变异系数的平均值最低,比较稳定,变异系数平均值为0.2922.传统模型与C4.5,KNN,LR和RF的组合在一些数据集上预测性能表现不稳定,变异系数平均值为0.5~1.0.而“传统模型+SMO”的变异系数平均值为2.182 4,远大于1.0,因此稳定性很差.

为了比较MCC和AUC的精确性以及在高度不平衡数据集和中度不平衡数据集上变异系数的变化情况,我们还使用了AUC指标进行评价.AUC指标包括计算出的“传统模型+各分类算法”在数据集上AUC的平均值,标准差和变异系数.同样,为了便于查看使用AUC评价指标时的传统分类算法的总体稳定性效果,我们将各个传统分类算法在数据集上AUC的变异系数求平均值.根据文献[5]的变异系数阈值标准,当变异系数大于0.05时采用“加粗”标记,得到如表5所示的“传统模型+各分类算法”在各数据集上AUC预测性能的变异系数.

表5 “传统模型+各分类算法”在各数据集上AUC的变异系数

算法	jedit-4.3	lucene	MC1	PC1	PC2	tomcat	平均值
C4.5	**0.146 3**	0.040 7	0.042 3	**0.057 2**	0.146 7	0.043 6	**0.079 5**
KNN	0.035 0	0.037 1	0.023 8	0.038 2	**0.079 6**	0.020 5	0.039 0
LR	**0.073 3**	0.024 0	0.041 1	0.015 6	**0.058 2**	0.013 2	0.037 6
NB	0.043 0	0.011 6	0.012 6	0.014 8	0.029 8	0.013 3	0.020 9
RF	0.037 8	0.039 2	0.017 5	0.017 3	0.043 0	0.015 1	0.028 3
SMO	**0.075 1**	0.033 0	**0.082 2**	**0.134 9**	**0.064 1**	**0.134 0**	**0.087 2**

由表4和表5可以发现:在高度不平衡数据集和中度不平衡数据集上,AUC的变异系数表现并没有太大的区别,而MCC在高度不平衡数据集上有较大的变异系数,与中度不平衡水平数据集的变异系数区别更加明显.总体来看,MCC的变异系数偏大.当预测模型没有分类能力(随机预测)时,分类模型并不符合我们的预期.此时,MCC的值为0,而AUC的值为0.5.平均值越接近0,变异系数越大,MCC可以更好地反馈预测模型性能,更符合软件缺陷预测的目的,MCC更适合作为软件缺陷预测预测模型的评价指标.

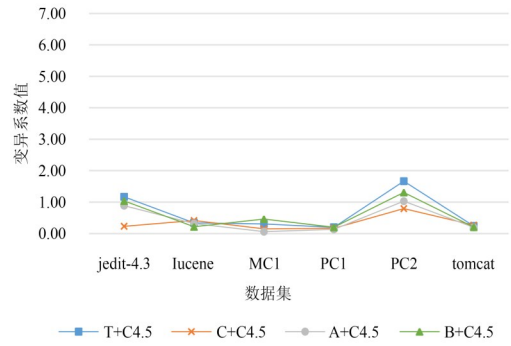
回答问题(1):在类不平衡情况下,“传统模型+NB”稳定性最好,其他预测模型的稳定性稍差.其中,“传统模型+SMO”稳定性最差.

5.3.2 代价敏感学习和集成学习对缺陷预测模型稳定性和预测性能的影响分析

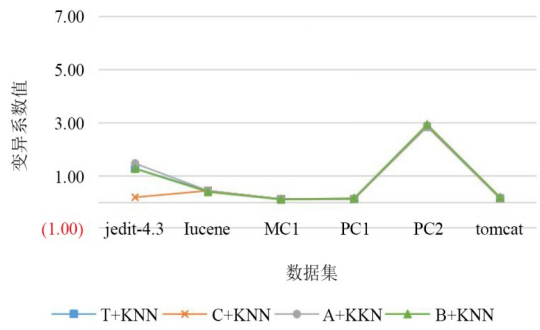
对于“问题(2):类不平衡情况下,代价敏感学习是否对传统模型的稳定性和预测性能带来提升?”,首先将传统模型换成代价敏感模型,然后对数据集进行训练预测获得预测模型的MCC值和变异系数.实验对6个分类算法采用代价敏感模型学习,将传统模型替换为代价敏感模型,再进行训练预测获得评价结果.

对于“问题(3):在类不平衡情况下,集成学习是否对传统模型的稳定性和预测性能带来提升?”,将传统模型替换为集成模型,再进行训练预测获得评价结果.

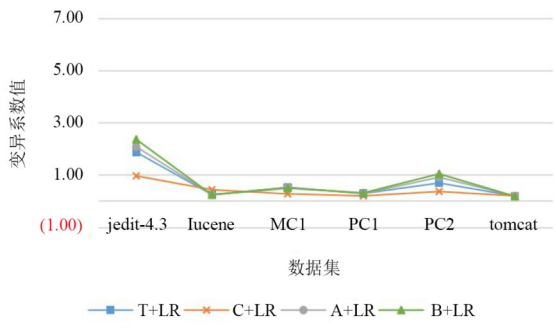
为了验证在类不平衡情况下代价敏感学习和集成学习是否对传统模型的稳定和预测性能带来提升,我们按“模型+分类算法”在各个数据集上的变异系数绘制成折线图,如图3所示.



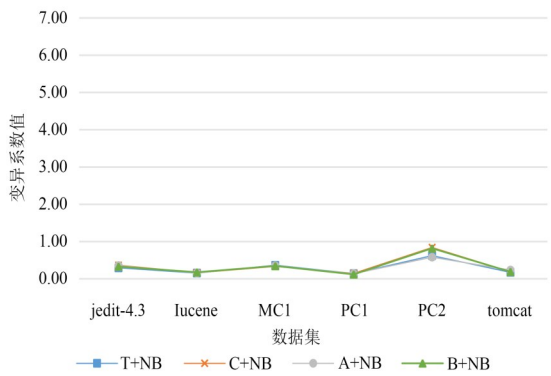
(*a*) “各模型+C4.5”变异系数



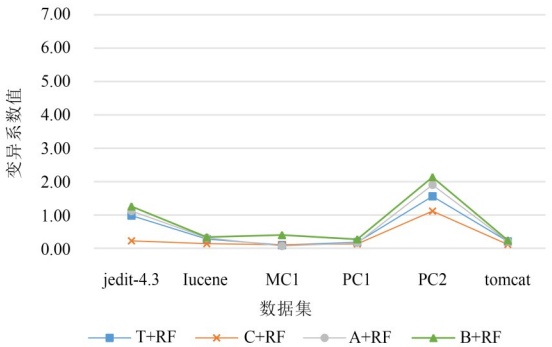
(*b*) “各模型+KNN”变异系数



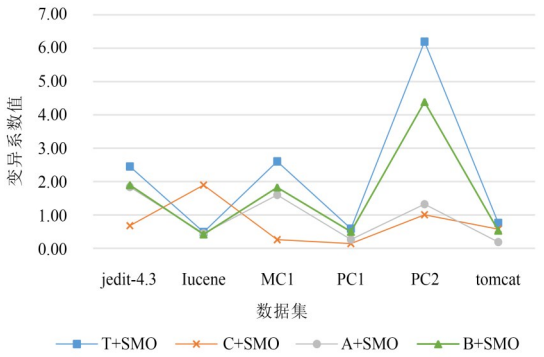
(*c*) “各模型+LR”变异系数



(*d*) “各模型+NB”变异系数



(*e*) “各模型+RF”变异系数



(*f*) “各模型+SMO”变异系数

图 3　各分类模型+各算法变异系数统计折线图

由图 3 可以发现:“代价敏感模型+SMO”相对于“传统模型+SMO”的稳定性提升效果最好;“代价敏感模型+C4.5”“代价敏感模型+LR”“代价敏感模型+RF”的稳定性提升效果一般;“代价敏感模型+KNN”和“代价敏感模型+NB”稳定性基本没有提升效果.“集成模型(A)+C4.5”和“集成模型(A)+SMO”相对于传统模型下分类算法的稳定性有提升效果,“集成模型(A)+SMO”稳定性提升效果非常明显,而“集成模型(*A*)+C4.5”的稳定性提升效果一般;“集成模型(A)+KNN”“集成模型(A)+LR”“集成模型(A)+NB”和“集成模型(A)+RF”的稳定性提升并不明显.“集成模型(B)+SMO”的稳定性相较于传统模型的稳定性也有比较明显的提升效果;“集成模型(B)+C4.5”“集成模型(B)+KNN”和“集成模型(B)+NB”对稳定性的提升效果不明显;而“集成模型(B)+LR”和“集成模型(B)+RF”对稳定性的改变更是出现了负效果,即加入不平衡学习算法 LR 或 RF 后,模型的稳定性变得更差.

为了验证在类不平衡情况下代价敏感学习和集成学习是否对传统模型分类性能有所提升,我们按分类算法划分成 6 组箱型图,每组有 4 个箱型图,分别代表

“传统模型+分类算法”(蓝色)、“代价敏感模型+分类算法”(橙色)、“集成模型(A)+分类算法”(灰色)、“集成模型(B)+分类算法”(绿色),如图4所示.其中,横轴表示分类算法,纵轴表示MCC值.

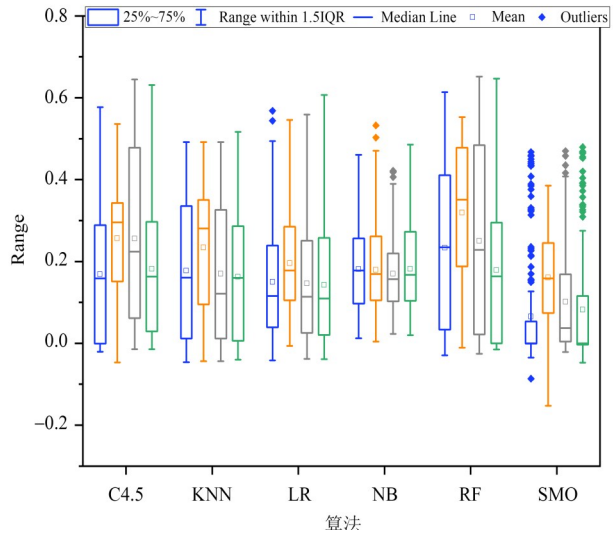


图4 “各模型+各分类算法”MCC值预测性能比较统计箱型图

从图4可以发现:代价敏感模型和传统模型比较预测性能时,除了“代价敏感模型+NB”与“传统模型+NB”相比没有明显的提升效果之外,对其他分类算法都有明显提升,其中对SMO的性能有明显的提升效果.“代价敏感模型+SMO”的下四分位数都在“传统模型+SMO”的上四分位数之上,其次是“代价敏感模型+C4.5”相对于“传统模型+C4.5”来说性能提高效果也比较明显,中位数在“传统模型+C4.5”的上四分位数之上.“集成模型(A)+C4.5”“集成模型(A)+RF”和“集成模型(A)+SMO”的性能相对于传统模型下的C4.5、RF和SMO都有提升.“集成模型(A)+LR”较于传统模型下的LR提升效果并不明显.“集成模型(A)+KNN”和“集成模型(A)+NB”的性能反而下降,即出现了负效果.值得注意的是,“集成模型(A)+C4.5”性能提升和表现一直很突出的RF在集成模型下的性能相差不大.“集成模型(B)+C4.5”和“集成模型(B)+SMO”的性能相对于传统模型下的C4.5和SMO都有提升.集成模型(B)+LR对传统模型下的分类算法提升效果并不明显.“集成模型(B)+KNN”“集成模型(B)+NB”和“集成模型(B)+RF”较于传统模型性能均出现了负效果,“集成模型(B)+KNN”“集成模型(B)+NB”较于传统模型性能出现的负效果较小,而“集成模型(B)+RF”负效果较为明显.

回答问题(2):在类不平衡情况下,“代价敏感模型+SMO”相对于“传统模型+SMO”的稳定性提升效果最好,“代价敏感模型+C4.5”“代价敏感模型+LR”“代价敏感模型+RF”分类算法的稳定性提升效果一般,“代价敏感模型+KNN”和“代价敏感模型+NB”稳定性基本没有提升效果.“代价敏感模型+SMO”提升预测性能最佳.“代价敏感模型+C4.5”“代价敏感模型+KNN”“代价敏感模型+LR”和“代价敏感模型+RF”不同程度地提升了预测性能,而代价敏感模型+NB并没有提升预测性能.

回答问题(3):在类不平衡情况下,“集成模型(A)+SMO”“集成模型(A)+C4.5”相对于传统模型下的分类算法的稳定性有提升效果,“集成模型(A)+SMO”稳定性提升效果明显,而“集成模型(A)+C4.5”的稳定性提升效果一般;“集成模型(A)+KNN”“集成模型(A)+LR”“集成模型(A)+NB”和“集成模型(A)+RF”的稳定性提升并不明显.“集成模型(A)+SMO”提升预测性能最佳;“集成模型(A)+C4.5”提升预测性能效果一般;“集成模型(A)+RF”仅仅提升预测性能,但是没有提升预测性能的稳定性;“集成模型(A)+LR”没有提升预测性能;“集成模型(A)+KNN”和“集成模型(A)+NB”均出现了不同程度的负效果.“集成模型(B)+SMO”和“集成模型(B)+C4.5”相对于传统模型下的分类算法的稳定性有提升效果;“集成模型(B)+KNN”和“集成模型(B)+NB”的稳定性提升并不明显;“集成模型(B)+LR”和“集成模型(B)+RF”的稳定性出现了负效果.“集成模型(B)+SMO”提升预测性能也很好;“集成模型(B)+C4.5”提升预测性能效果一般;“集成模型(B)+LR”没有提升预测性能;而“集成模型(A)+KNN”“集成模型(A)+NB”和“集成模型(A)+RF”有不同程度的负效果.

5.3.3 代价敏感学习和集成学习的比较

对于“问题(4):在类不平衡情况下,代价敏感模型和集成模型分别对传统模型的稳定性和预测性能提升如何?”,我们在对传统模型预测性能提升的基础上,分析代价敏感模型和集成模型中的哪个模型对预测性能提升效果更佳.

通过从图3中统计折线上的点,点与(传统模型上的)点基本重合视为无提升;在下方表示有提升,若提升大于0.5则表示提升明显;点在上方则表示负效果;采用点计数,以类别中点个数最多的类别作为此“模型+分类算法”下的稳定性.考虑到在实际应用中,训练出的模型常用于进行跨项目缺陷预测和跨公司缺陷预测,稳定性是评估预测模型非常重要的决定性因素,因此更希望模型的稳定性越高越好.因此,当各模型的稳定性出现类别(有提升、无提升或负效果)计数点相同时,选择淘汰的优先级别是:负效果>无提升>有提升.

在评价性能时,主要看中位数和平均值位置,同时也考虑高、低四分位数和最大值.中位数和平均值与传统模型相差不大则视为无提升;中位数和平均值高于则视为有提升,当高于传统模型性能值的上四分位数

值时(该“模型+分类算法”的中位数和平均值完全位于传统模型箱子上方),视为提升效果明显;若是中位数和平均值低于传统模型,则出现负效果. 代价敏感模型和集成模型相对于传统模型提升效果如表6所示. 其中,“++”代表提升效果明显,“+”提升效果一般,“0”代表没有提升效果,“-”代表出现了负效果.

表6　代价敏感模型和集成模型相对于传统模型的提升效果

	模型	C4.5	KNN	LR	NB	RF	SMO
稳定性	代价敏感模型	+	0	+	0	+	++
	集成模型(A)	+	0	0	0	0	++
	集成模型(B)	+	0	-	0	-	++
性能	代价敏感模型	+	+	+	0	+	++
	集成模型(A)	+	-	0	-	+	+
	集成模型(B)	+	-	0	-	-	+

通过图3和表6可以发现:对于算法C4.5和SMO,代价敏感模型、集成模型(A)和集成模型(B)对传统模型+C4.5和SMO稳定性都有提升效果. 其中,集成模型(A)+SMO对算法SMO的稳定性提升更好;对于算法KNN和NB,代价敏感模型、集成模型(A)和集成模型(B)较于对应的传统模型其稳定性都没有提升;对于算法LR和RF,代价敏感模型、集成模型(A)和集成模型(B)较于对应的传统模型稳定性效果不同,其中代价敏感模型表现为提升了稳定性,集成模型(A)表现为无提升,集成模型(B)则表现出了负效果.

通过图4和表6可以发现:对于算法C4.5和SMO,代价敏感模型和集成模型(A)和集成模型(B)相较于其对应的传统模型的预测性能都有提升. 代价敏感模型对传统模型+SMO预测性能提升效果明显且优于集成模型(A)和集成模型(B)+SMO的提升效果;在代价敏感模型、集成模型(A)和集成模型(B)+KNN、LR、NB和RF这12个组合的预测性能分别较于传统模型+KNN、LR、NB和RF上,除代价敏感模型对传统模型+KNN、LR和RF和集成模型(A)+RF的预测性能这4个组合有提升外,其余8个组合表现为对预测性能无提升,甚至有的组合出现负效果. 在所有的组合中代价敏感模型+RF预测性能表现最优.

回答问题(4):总体上看,“传统模型+NB”和“代价敏感模型+NB”稳定性最好,而排除集成模型(A)和集成模型(B)+NB,由于其稳定性好,但是预测性能出现了负效果;“代价敏感模型+RF”“集成模型(A)+RF”和“集成模型(A)+C4.5”预测性能最好. 如果在实际应用中进行跨项目缺陷预测或跨公司缺陷预测,应该要考虑整体的稳定性和性能两方面,才能获得最好的效果. 因此,这时可以使用“代价敏感模型+RF”,这是由于它在保证稳定性的同时,性能效果表现也有明显优势.

6　结束语

本文探讨类不平衡问题对软件缺陷预测性能的影响,从算法层面研究了代价敏感学习中的元代价模型过程和集成学习的代价敏感集成学习两个方法对类不平衡问题的预测性能提升效果. 首先,分析传统机器学习算法在类不平衡问题下的敏感程度;其次,从稳定性和性能两个方面分析传统机器算法的预测性能;然后,从稳定性和性能分析代价敏感模型和集成模型对传统分类算法的提升效果;最后,分析代价敏感模型和集成模型提升效果的优势之处.

(1)实验结果表明:在不平衡率增大时,MCC的值明显下降,AUC的变异系数差别却不大,表明了MCC值更适合作为分类算法之间预测性能的对比指标.

(2)NB分类算法对分类不平衡问题敏感性比较低,但在不平衡率增大时预测性能仍然有所下降. C4.5,KNN,LR和RF分类算法敏感性则较高. 而SMO分类算法敏感性很高,几乎没有能力应对类不平衡问题. 这说明传统分类算法遇到类不平衡问题时总会受到影响,预测性能会下降. NB分类算法稳定性最高,RF分类算法性能最好.

(3)总体上,在稳定性和性能方面,代价敏感模型表现优于集成模型. 但在实际应用中选择分类模型时,还是要考虑训练集和测试集样本的分布和样本特征. 如果只进行样本分布和特征相同的项目内缺陷预测,即仅考虑分类算法预测性能最好即可,这时“代价敏感模型+RF”“集成模型(A)+RF”或“集成模型(A)+C4.5”都是较好的选择;如果进行跨项目缺陷预测或跨公司缺陷预测,应该要考虑整体的稳定性和性能两方面,才能获得最好的效果,这时可以使用“代价敏感模型+RF”,这是由于它在保证稳定性的同时,性能效果表现也有明显优势.

虽然代价敏感模型和集成模型能够对部分算法有提升效果,但是并未提出解决类不平衡问题的改进算法. 因此在未来的工作中将对以下内容进行深入探究. (1)本文分类算法使用的是默认参数,参数选择对软件缺陷预测性能也有影响,需要在参数变化情况下进行更深入的研究. 此外,还需要深入研究代价敏感模型的特征矩阵参数不同带来的影响. 加入参数后,模型的选择变得多样,参数变化容易引起组合爆炸问题,因此从大量的数据中寻找到合适的特征矩阵,以获得最优效果的组合. (2)文中没有研究从数据处理层面解决类不平衡问题的方法,未来工作中也值得对之深入探究.

参考文献

[1] CATAL C, DIRI B N. Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem[J]. Information Sciences, 2009, 179(8): 1040-1058.

[2] KRAWCZYK B. Learning from imbalanced data: Open challenges and future directions[J]. Progress in Artificial Intelligence, 2016, 5(4): 221-232.

[3] CAI Q, HE H B, MAN H. Imbalanced evolving self-organizing learning[J]. Neurocomputing, 2014, 133: 258-270.

[4] POWERS D M W . Evaluation: From precision, recall and *F*-measure to ROC, informedness, markedness and correlation[EB/OL]. (2020-10-11) [2021-07-14]. https://arxiv.org/abs/2010.16061.

[5] 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 Q. Research on Software Defect Prediction Method Based on Machine Learning[D]. Xuzhou: China University of Mining and Technology, 2017. (in Chinese)

[6] 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: 321-357.

[7] TAHIR M A, KITTLER J, YAN F. Inverse random under sampling for class imbalance problem and its application to multi-label classification[J]. Pattern Recognition, 2012, 45(10): 3738-3750.

[8] 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.

[9] ZHOU Z H, LIU X Y. Training cost-sensitive neural networks with methods addressing the class imbalance problem[J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(1): 63-77.

[10] LÓPEZ V, FERNÁNDEZ A, GARCÍA S, et al. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics [J]. Information Sciences, 2013, 250: 113-141.

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

[12] GAO K H, KHOSHGOFTAAR T M. Assessments of feature selection techniques with respect to data sampling for highly imbalanced software measurement data[J]. International Journal of Reliability, Quality and Safety Engineering, 2015, 22(2): 1550010.

[13] LARADJI I H, ALSHAYEB M, GHOUTI L. Software defect prediction using ensemble learning on selected features[J]. Information and Software Technology, 2015, 58: 388-402.

[14] WANG S, YAO X. Using class imbalance learning for software defect prediction[J]. IEEE Transactions on Reliability, 2013, 62(2): 434-443.

[15] WAHONO R S, SURYANA N. Combining particle swarm optimization based feature selection and bagging technique for software defect prediction[J]. International Journal of Software Engineering and Its Applications, 2013, 7(5): 153-166.

[16] MALHOTRA R, JAIN J. Handling imbalanced data using ensemble learning in software defect prediction[C]// 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence). Piscataway: IEEE, 2020: 300-304.

[17] 王子健 . 基于主动集成学习的软件缺陷预测模型研究[D]. 秦皇岛: 燕山大学, 2021.
WANG Z J. Research on Software Defect Prediction Model Based on Active Integrated Learning[D]. Qinhuangdao: Yanshan University, 2021. (in Chinese)

[18] LESSMANN S, BAESENS B, MUES C, et al. Benchmarking classification models for software defect prediction: A proposed framework and novel findings[J]. IEEE Transactions on Software Engineering, 2008, 34(4): 485-496.

[19] SONG Q B, GUO Y C, SHEPPERD M. A comprehensive investigation of the role of imbalanced learning for software defect prediction[J]. IEEE Transactions on Software Engineering, 2019, 45(12): 1253-1269.

[20] BALOGUN A, BASRI S, ABDULKADIR S J, et al. Software defect prediction: Analysis of class imbalance and performance stability[J]. Journal of Engineering Science and Technology, 2019, 14(6): 3294-3308.

[21] ELDHO K J. Impact of unbalanced classification on the performance of software defect prediction models[J]. Indian Journal of Science and Technology, 2022, 15(6): 237-242.

[22] 宫丽娜, 姜淑娟, 姜丽 . 软件缺陷预测技术研究进展[J]. 软件学报, 2019, 30(10): 3090-3114
GONG L N, JIANG S J, JIANG L. Research progress of

software defect prediction[J]. Journal of Software, 2019, 30(10): 3090-3114. (in Chinese)

[23] QIAO X Y, LIU Y F. Adaptive weighted learning for unbalanced multicategory classification[J]. Biometrics, 2009, 65(1): 159-168.

作者简介

张艳梅　女,1982年生,唐山人. 博士. 副教授, CCF专业会员. 主要研究方向为软件分析与测试、软件缺陷预测等.
E-mail: ymzhang@cumt.edu.cn

植胜林(通讯作者)　男,1997年生, 梧州人. 学士. 软件研发工程师. 主要研究领域为应用软件研发、软件分析与测试、软件缺陷预测等.
E-mail: zhilincumt@163.com

姜淑娟　女,1966年生,莱阳人. 博士. 教授, 博士生导师,CCF专业会员. 主要研究领域为软件分析与测试、编译技术等.
E-mail: shjjiang@cumt.edu.cn

袁　冠　男,1982年生,徐州人. 博士. 教授,博士生导师,CCF高级会员. 主要研究领域为时空大数据技术、计算智能和软件工程等.
E-mail: yuanguan@cumt.edu.cn