面向类不平衡数据集的软件缺陷预测模型*

李 冉，周丽娟，王 华
（首都师范大学 信息工程学院，北京 100048）

摘 要： 软件缺陷数据的类不平衡问题会影响缺陷预测分类的准确性，为解决类不平衡数据对预测分类的影响，针对如何优化数据预处理的算法执行顺序进行了研究，提出了一种有效提升分类效果的软件缺陷预测模型（ASRAdaBoost）。该算法模型在根据对照实验确定数据预处理最优顺序后，采用特征选择卡方检验算法，再执行SMOTE过采样与简单采样方法，解决数据类不平衡和属性冗余同时存在的问题，最后结合AdaBoost集成算法，构建出软件缺陷预测模型ASRAdaBoost。实验均采用J48决策树作为基分类器，实验结果表明ASRAdaBoost算法模型有效地提高了软件缺陷预测的准确性，得到了更好的分类效果。

关键词： 软件缺陷预测；类不平衡数据；特征选择；集成算法
中图分类号： TP311.5 **文献标志码：** A **文章编号：** 1001-3695(2018)09-2806-05
doi:10.3969/j.issn.1001-3695.2018.09.057

Software defect prediction model based on class imbalanced datasets

Li Ran, Zhou Lijuan, Wang Hua
(*College of Information Engineering, Capital Normal University, Beijing* 100048, *China*)

Abstract: The problem of class imbalanced data of software defect will affect the accuracy of defect predictive classification. In order to solve the problem of classification, this paper discussed the order of algorithm execution of optimized data preprocessing and developed a software defect prediction model (ASRAdaBoost) to effectively improve the classification. The algorithm was based on the comparison experiment to determine the optimal sequence of data preprocessing, using the chi-square test of attribute selection, and then performed SMOTE oversampling and resample method to solve the imbalanced data and attri-bute redundancy problems, using the AdaBoostensemble algorithm to build a software defect prediction model ASRAdaBoost eventually. The experimental results show that the ASRAdaBoost model can effectively improve the accuracy of software defect prediction and get a better classification effect.

Key words: software defect prediction; class imbalanced data; attribute selection; ensemble algorithm

软件的可靠性是软件工程领域中最重要的性能指标。为了最大程度地提升软件质量，应对软件缺陷进行预测。软件缺陷预测[1]的关键在于如何对软件模块进行缺陷预测分类，若将软件缺陷所在模块划分为高风险模块(fault-prone)与低风险模块，则其重点、难点即是发现一个软件产品中存在的高风险模块，如何将有限资源合理分配至高风险模块中，且达到预测分类准确性，从而找出其中存在的软件缺陷。

在近年的软件缺陷预测研究中，如何解决类不平衡问题成为目前研究的热点方向。由于真实世界中大多数软件系统中的数据集是不平衡的，即高风险模块数量远远少于低风险模块数量，即高风险模块占少数，也被称为少数类[2]，低风险模块被称为多数类。传统的分类学习算法中，最注重整体分类准确率的最优值，在具有类不平衡特征的数据集方面，分类器会对多数类分类的准确率具有更高值，将导致大量高风险模块被误分类至低风险模块，降低了少数类分类的准确度，从而会影响整个软件系统的开发过程。所以，分类实验结果会无实际意义。本文将针对如何提升类不平衡数据集的软件缺陷预测分类准确性进行探讨研究与实验分析。

1 研究背景

目前针对数据集类不平衡问题，可用数据层面与算法层面来解决该问题。在数据层面，主要包含各种采样技术，如SMOTE[3]、RUS(random under sampling)[4]等采样抽样技术；算法层面主要包含集成学习(如Boosting[5]、Bagging[6])、代价敏感学习(如AdaCost[7]、MetaCost[8])以及单类别学习[9](少数类样本作为训练样本)等算法，运用并优化各分类算法以处理不平衡数据来提升分类准确性。目前国内外研究人员在软件缺陷预测领域已提出诸多算法以解决类不平衡问题，例如：张晓风等人[10]提出的基于PCA分布抽样的软件预测分析模型，利用分布函数以合成新数据样本的过抽样与随机向下抽样相结合来解决数据不平衡问题；熊婧等人[11]首次提出将神经网络作为AdaBoost集成算法的分类器以构成更强的级联分类器；Liu等人[12]首次提出二级代价敏感学习方法以解决类不平衡问题；Khoshgoftaar等人[13]根据特征选择算法解决极度不平衡数据的问题，并进行了相关实验验证，但对于数据预处理的方法顺序选择上不具有一定代表性，且引入了人工噪声，分类效果没有明显提升。

基于以上研究背景，本文提出新的解决思路：以面向类不平衡的软件缺陷预测数据集为研究对象，提出优化数据预处理顺序的基于ASRAdaBoost算法的软件缺陷预测模型，以解决样本中的数据不平衡问题，提升分类的准确性与可靠性。该算法是在数据预处理阶段采用特征选择与组合采样技术，验证分析

收稿日期：2017-10-19；修回日期：2017-11-29 基金项目：国家自然科学基金资助项目(61601310)；高可靠嵌入式系统技术北京市工程研究中心资助项目(2013BAH19F01)

作者简介：李冉(1994-)，女，北京人，硕士，主要研究方向为机器学习、数据挖掘(1281276560@qq.com)；周丽娟(1969-)，女，教授，博士，主要研究方向为数据挖掘、大数据处理；王华(1964-)，女，副教授，博士，主要研究方向为软件工程、数据库系统.

其预处理方法的最佳采用顺序,再融合集成算法 AdaBoost 的 J48 分类器,对符合类不平衡条件的多个 UCI 数据集进行实验验证。本文的实验结果均表明: 所提出的 ASRAdaBoost 算法模型的分类准确性有明显提升。

2　相关理论与技术

2.1　特征选择

特征选择(attributeSelection) 是在数据集中根据某种评价标准选择出最优属性的特征集[14] ,因此去除数据集中不相关以及冗余属性,能够使得相应模型和算法具有最好性能。至今为止,国内外不少学者均对特征选择步骤进行了定义,本文整理出特征选择算法的步骤如下:

a) 生成候选特征子集。首先根据不同起点和方向开始搜索,如前向搜索、后向搜索、双向搜索、随机搜索; 然后依据特定的搜索策略生成子集,如穷举式搜索、序列搜索、随机搜索。

b) 进行生成子集的特征评估。当候选特征子集生成后,需要评估标准对其评估,模型大致分为 wrapper、embedding、filter 三类。Wrapper 称为包装器模型,通过基于子集的算法而得出的分类结果作为特征评价依据; embedding 称为嵌入式模型,即特征选择的过程与分类器训练过程在同一个优化阶段完成,将两者的执行过程较好地融合在一起; filter 称为过滤器模型,与具体的分类算法无关,是通过子集本身的数据内在特点进行特征评估。因此,为避免后续实验选用的分类器对预测分类结果的影响,本文实验中的特征选择阶段将采用 filter 过滤器模型特征选择方法,该模型框架如图 1 所示。

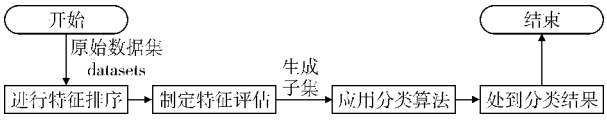

图 1　Filter 特征选择模型框架

c) 停止标准。以上步骤经过多次迭代过程,直到该步骤停止标准的满足。停止标准的设置与评估准则以及软件的具体应用需求有较大关联,较为常见的停止标准有特征子集的大小、错误阈值以及算法的运算次数。

2.2　采样技术

2.2.1　过采样(over-sampling)

即通过增加少数类样本的数量以达到数据集中类的比例平衡。其中,SMOTE 算法是最为经典的一种过采样算法,即通过在相近的少数类样本间进行线性插值以生成新数据实例。即对于类不平衡数据集中的每一个少数类样本 x,从其最近邻的欧氏距离中选择 k 个少数类样本(k 一般取值为 5) ,再在 x 与 k 个最近邻样本的连线中随机选择其中的 n 个样本,对每个最近邻样本再进行线性插值以生成新数据样本作为合成的少数类样本,从而使得数据集中的多数类样本与少数类样本比例达到平衡,但其缺点在于随着样本数量的增加,训练时间会逐渐增加,在一定程度上降低了软件缺陷预测模型的效率。

2.2.2　欠采样(under-sampling)

即通过去除多数类样本的数量以达到数据集中类的平衡。在软件缺陷预测的数据预处理阶段,随机去除正常样本,完全留存有软件缺陷标记的缺陷样本,以此解决数据不平衡问题,其缺点是在删除部分多数类的信息时会导致部分关键数据的丢失。

2.3　AdaBoost 算法概述

集成学习是当今机器学习领域中应用最为广泛的技术之一,其中 AdaBoost 算法较为常用。AdaBoost 算法是由 Boosting 算法发展而来的,也是由 Freund 和 Schapire 等人提出的一种迭代算法[15] 。该算法旨在将弱学习算法提升为强学习算法,即通过逐步增强前一个分类器分类错误的样本权重,再将加权后的所有样本进行训练。同时在每一轮迭代过程中再新增一个新的弱分类器,直到满足最小错误率或最大迭代次数的条件,因此有效保证了分类的错误率极大值随着训练次数的增加而有所递减。具体算法步骤如下:

a) 初始化训练集中每个样本相同的权重为 $1/n$,其中 n 是训练集中的样本个数,并且给定最大迭代次数为 N 轮。

b) 执行多轮迭代,根据分类错误率选择合适阈值并且调整样本权值。

c) 根据新的各样本权重为下一迭代的分类器生成训练数据集,最终经过 N 次迭代,生成预测性能更好的集成分类器——强分类器。

3　ASRAdaBoost 算法内容

为降低数据集中的类不平衡问题对软件缺陷预测结果的影响,本文提出 ASRAdaBoost 算法模型。利用 weka 平台,在数据预处理阶段针对特征选择与数据采样的顺序进行探讨,先执行特征选择后进行组合采样方法的预处理顺序,随即展开本算法的整体实验验证过程,并设计实验对照组验证分析本文算法的优势。本算法首先采用卡方检验特征选择算法,对选用的软件缺陷预测类不平衡数据集中的属性进行特征选择,去除冗余及不相关属性。卡方检验是一种假设性检验的方法,通常用来检验两个变量的相关性,即评价两个事件是否独立或相关。在描述实际值与理论值的偏差程度时,卡方值越大,代表两个变量的相互独立性越弱; 反之,代表两变量的相互独立性越强。假设自变量有 N 种取值,因变量有 M 种取值可能,即对偏差程度的判定可以表示为

$$\chi^2 = \sum \frac{(A-E)^2}{E} \tag{1}$$

在依据特征选择的卡方检验判断变量独立性后,设置阈值将不相关的冗余特征去除,而后利用监督学习中的 SMOTE 过抽样方法以增加少数类样本实例,再结合使用 resample 简单采样技术去除多余的多数类样本实例,即本文的组合采样方法,最后根据 AdaBoost 集成算法的 J48 分类器进行结果分类,得到的软件缺陷预测模型实验结果的分类效果明显得到提升。该算法流程如图 2 所示。

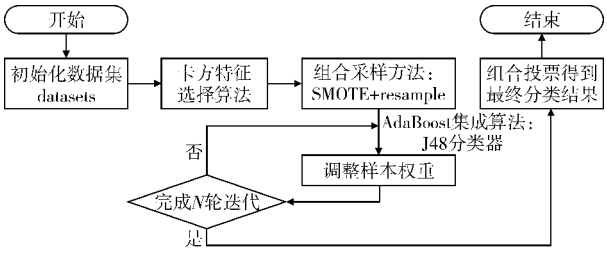

图 2　ASRAdaBoost 算法流程

在上述算法内容探讨的数据预处理优化阶段,本文实验应用机器学习中的特征选择方法与过采样、简单采样技术,将过采样技术与简单采样技术结合成本文提出的组合采样方法,但特征选择与组合采样方法的相关执行顺序是否对于类不平衡数据集的分类效果有影响,目前仅有 Shanab 等人[16] 对于数据预处理执行顺序与数据维度进行了相关研究,由于其在研究中引入了人工噪声,对于分类效果有一定的影响。基于此,本文将对软件缺陷预测中的数据预处理顺序进行简要探讨,比较特

征选择与组合采样的执行顺序的效果优劣后再进行之后的实验操作。实验采用 UCI 数据集的其中四个类不平衡数据集进行预处理执行顺序的对比实验，均采用十折交叉验证法，具体实验设计对照组分为以下三种：

a) Ori。对原始数据集直接使用 C4.5 决策树的 J48 分类器进行预测分类。

b) A1。对原数据集先执行组合采样技术后执行特征选择，再使用 J48 分类器进行分类。

c) A2。对原数据集先执行特征选择后执行组合采样技术，再使用 J48 分类器进行分类。

本实验将少数类比例提升至样本总数的 40%，利用简单采样将实验样本数量降至原总数的一半，以提升预测分类速度。根据以上比例计算出每个数据集所对应的不同的过采样率以及欠采样率，执行组合采样方法的实验信息与结果如表14所示，实验结果对比如图 35 所示。

表 1 设置相应过采样率与欠采样率的数据集信息

数据集名称	过采样率/%	欠采样率/%	原样本总个数	现样本总个数
credit-g	55.60	42.88	1 000	499
diabetes	24.38	46.10	768	384
hepatitis	156	37.7	155	76
ionosphere	19.05	46.79	351	175

表 2 三种方法在四个数据集上的查准率 P 对比

数据集名称	Ori	A1	A2	数据集名称	Ori	A1	A2
credit-g	0.511	0.717	0.723	hepatitis	0.667	0.733	0.783
diabetes	0.632	0.804	0.808	ionosphere	0.929	0.949	0.953

表 3 三种方法在四个数据集上的 F-measure 值对比

数据集名称	Ori	A1	A2	数据集名称	Ori	A1	A2
credit-g	0.442	0.649	0.661	hepatitis	0.528	0.746	0.783
diabetes	0.614	0.695	0.743	ionosphere	0.874	0.875	0.917

表 4 三种方法在四个数据集上的 AUC 值对比

数据集名称	Ori	A1	A2	数据集名称	Ori	A1	A2
credit-g	0.639	0.72	0.734	hepatitis	0.708	0.822	0.802
diabetes	0.751	0.837	0.852	ionosphere	0.892	0.865	0.921

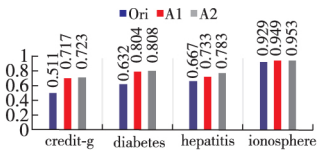

图 3 Ori、A1 与 A2 方法的查准率 P 对比

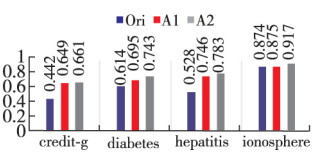

图 4 Ori、A1 与 A2 方法的 F-measure 值对比

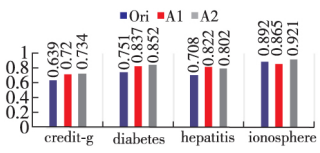

图 5 Ori、A1 与 A2 方法的 AUC 值对比

由表 24 的数据以及图 35 的柱形图可反映出三个对照组在四个 UCI 数据集上的不同实验结果，其中 A1 与 A2 组的各项分类预测性能评价指标基本均优于原始数据 Ori 组的性能指标。即首先肯定，对于类不平衡的数据集，数据预处理的特征选择与采样方法均能够提升分类预测的性能。在本文所探讨的特征选择算法与组合采样方法的顺序方面，其中 A2 组的查准率 P，F-measure 值与 AUC 值均高于 Ori 组与 A1 组，即 A2 组的软件缺陷预测分类性能优于其他两个对照组。尽管本实验中 A1 与 A2 的实验效果差别不大，但从以上各性能评价指标来看，本算法先执行卡方检验特征选择后执行 SMOTE 过采样与简单采样的组合采样算法，使得软件缺陷预测的分类效果更优秀。因此，本文后续的实验与整体 ASRAdaBoost 算法模型将采用先特征选择后组合采样的数据预处理顺序。

4 实验设计与验证分析

4.1 软件缺陷预测实验数据集选取

本实验选取美国加州大学欧文分校用于机器学习的 UCI 数据集，其中来自分类任务的 279 个数据集大部分是多类不平衡数据集。本实验根据数据集的属性个数、样本总个数以及多数类与少数类的比例进行条件筛选，由于数据的预处理方法是基于 weka 平台实现，其中算法性能对高维数据效果不佳。所以，本文从中选取四个二类不平衡的 UCI 数据集作为本次软件缺陷预测的实验数据集，分别为：

a) credit-g，即 Statlog-German Credit Dataset。它是德国信用数据，用于预测个人信用是否良好，信用分类为 good 与 bad 两类，代表信用好与坏，其中 bad 类为少数类。

b) diabetes，即 Pima Indians Diabetes Dataset。它是根据医疗历史记录来预测比马印第安人近五年内的糖尿病发病率，二分类分别为 tested-negative 与 tested-positive，代表测试结果阴性与阳性，其中 tested-positive 类为少数类。

c) hepatitis，即肝类数据集用于预测肝炎发病情况，二分类为 DIE 与 LIVE，代表肝炎病例结果的死亡与存活，其中 DIE 为少数类。

d) ionosphere，即电离层数据集。根据给定的电离层中的自由电子雷达回波以预测大气结构，二分类分别为 g 与 b，代表好与坏，其中 b 为少数类。

实验选用的数据集信息如表 5 所示。

表 5 实验采用的四个 UCI 数据集信息

数据集名称	样本个数	属性个数	少数类个数	少数类占比/%
credit-g	1 000	21	300	30.0
diabetes	768	9	268	34.9
hepatitis	155	20	32	20.6
ionosphere	351	35	126	35.9

4.2 软件缺陷预测实验性能评估指标

对于软件缺陷预测分类模型，本实验采用传统的分类器性能评估指标，并且对于本文探讨的二分类问题，可将数据集样例根据其实际所属类别与分类器预测分类的类别组合划分为真正例(true positive，即 TP)、假正例(false positive，即 FP)、真反例(true negative，即 TN)、假反例(false negative，即 FN)四种情况，即 $TP+FP+FN+TN=$ 数据集样本总数[17]，其中将少数类作为正例(即代表软件中有缺陷的类别)，同时可将分类结果用混淆矩阵(表 6)表示。

表 6 混淆矩阵

实际情况	预测分类结果	
	正例(预测有缺陷)	反例(预测无缺陷)
正例(真实有缺陷)	TP	FN
反例(真实无缺陷)	FP	TN

$$\text{查准率 } P=\frac{TP}{TP+FP} \tag{2}$$

$$\text{查全率 } R=\frac{TP}{TP+FN} \tag{3}$$

在实际情况中，软件缺陷预测模型目标是在现有条件内最大程度找到含有缺陷的软件模块，最小限度地将无缺陷软件模块误分类至有缺陷软件模块中。在针对本文探讨的类

不平衡问题中,诸多研究人员发现仅有查准率 P 与查全率 R 两种性能评估指标,不能够全面地反映软件缺陷评测分类准确性的情况,因此本文采用查准率 P、F-measure 值、AUC(ROC area 值)作为本文实验中 ASRAdaBoost 算法模型的软件缺陷预测分类的性能评测指标。其中 F-measure 值与 ROC 值的定义如下:

$$F\text{-measure} = \frac{(1+\beta^2) \times P \times R}{(\beta^2 \times P) + R} \tag{4}$$

F-measure 值是基于查准率 P 与查全率 R 的加权调和平均值,综合了查准率与查全率的评测结果,有利于作为软件缺陷预测中不平衡数据集分类结果的预测性能评估指标。其中 β 度量的是查准率与查全率的相对重要性,在本文中 β 取值为 1。

ROC 曲线[18]全称为受试者工作特征曲线(receiver operating characteristic curve),在机器学习领域中用于描述不同分类器的学习性能评估比较[19],根据不同分类器的结果对测试样例进行排序,按此顺序将样本逐一作为正例进行预测并计算,将其计算结果作为横纵坐标作图,即得到 ROC 曲线。其横坐标轴是假正例率(false positive rate,FPR),纵坐标轴为真正例率(true positive rate,TPR),两者分别定义为

$$\mathrm{FPR} = \frac{FP}{TN + FP} \tag{5}$$

$$\mathrm{TPR} = \frac{TP}{TP + FN} \tag{6}$$

在实际情况中,采用 ROC 曲线与坐标轴围成的下方面积 AUC 值作为实用的评价指标以代替 ROC 曲线,AUC 值综合了软件缺陷预测方面的误报率与预测概率[20]。AUC 值越大,软件缺陷预测模型的性能越好,其中,随机预测模型的 AUC = 0.5,理想模型的 AUC = 1。其中,ROC 曲线与 AUC 值的关系如图 6 所示。

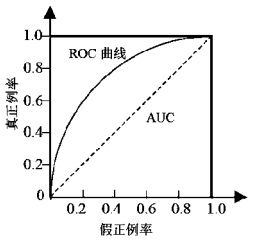

图 6　ROC 曲线与 AUC

4.3　实验设计与结果分析

本文的实验平台采用 weka 怀卡托智能分析环境,该平台能够较为便捷地使用本文模型算法进行实验。在对比特征选择与组合采样技术的执行顺序后,本文实验的模型算法中特征选择部分继续采用卡方检验,通过假设检验的思想减少数据集的特征数量及训练时间,其中阈值设置为 0。组合采样技术运用经典过采样 SMOTE 方法与 resample 简单欠采样方法将少数类比例提升至 40%(此时少数类的分类精度最为理想),其中 SMOTE 算法的邻域值设置为 5,本实验的组合采样技术将样本总数调整至原数据集数量的一半,最后结合集成算法 AdaBoost,选用 C4.5 决策树的 J48 分类器作为基分类器,构建出基于 ASRAdaBoost 算法的软件缺陷预测模型。

本文所有实验采用十折交叉验证法,将文中所用数据集划分为 10 个子集,其中 9 个作为训练集,1 个作为测试集以保证最终预测分类测试结果的客观性与可靠性,最后比较仅使用 J48 分类器进行分类的实验结果,经过数据预处理阶段的实验结果,以及本文提出的 ASRAdaBoost 模型训练后的三种软件缺陷预测分类效果,使用查准率 P、F-measure 值以及 AUC 值来体现。针对 ASRAdaBoost 算法的实验设计对照组分为以下三种:

a) Ori。直接使用 C4.5 决策树的 J48 分类器进行分类。

b) A2。先执行特征选择后执行组合采样技术,再使用 J48 分类器进行分类。

c) ASRAdaBoost,即本文提出的 ASRAdaBoost 算法。先对原始总样本执行卡方检验特征选择,再针对其执行 SMOTE 过采样方法、简单采样技术,最后融合 AdaBoost 集成算法,采用 J48 分类器进行十折交叉验证法对其进行分类预测。

以上三组对照实验结果如表 79 所示,实验结果对比如图 79 所示。

表 7　三种方法在四个数据集上的查准率 P 对比

数据集名称	Ori	A2	ASRAdaBoost
credit-g	0.511	0.723	0.728
diabetes	0.632	0.808	0.811
hepatitis	0.667	0.783	0.778
ionosphere	0.929	0.953	0.969

表 8　三种方法在四个数据集上的 F-measure 值对比

数据集名称	Ori	A2	ASRAdaBoost
credit-g	0.442	0.661	0.728
diabetes	0.614	0.743	0.814
hepatitis	0.528	0.783	0.840
ionosphere	0.874	0.917	0.940

表 9　三种方法在四个数据集上的 AUC 值对比

数据集名称	Ori	A2	ASRAdaBoost
credit-g	0.639	0.734	0.857
diabetes	0.751	0.852	0.923
hepatitis	0.708	0.802	0.970
ionosphere	0.892	0.921	0.959

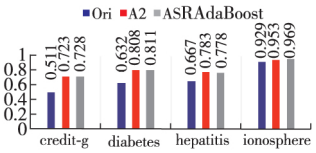

图 7　Ori、A2 与 ASRAdaBoost 方法的查准率 P 对比

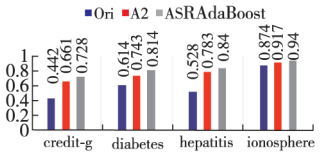

图 8　Ori、A2 与 ASRAdaBoost 方法的 F-measure 值对比

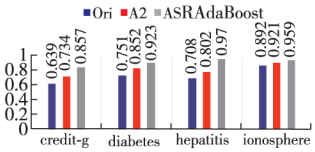

图 9　Ori、A2 与 ASRAdaBoost 方法的 AUC 值对比

表 7 9 以及图 7 9 列出了分别使用查准率 P、F-measure 值以及 AUC 值进行分类预测性能评估的三种方法评测效果柱形图,即仅使用 J48 分类器分类的方法、使用组合采样方法处理的 J48 分类器分类的方法以及本文提出的 ASRAdaBoost 模型算法。从上述的实验结果可发现,本文的 ASRAdaBoost 模型算法在四个数据集上的 F-measure 值与 AUC 值均为最高。在查准率 P 方面,只有 hepatitis 数据集的查准率未有明显提升,较之实验对照组 A2 的查准率 P,结果较差。进行实验结果分析后发现,由于 hepatitis 数据集的样本数量比其他数据集的样本数量少,所以在 ASRAdaBoost 算法的最后一步融合 AdaBoost 集成算法时,本文提出的模型无法准确地获取数据集样本特征以进行分类预测,导致数据欠拟合的现象发生,使得实验效果较之其他数据集表现不够理想,查准率未得到提升。除 hepatitis 数据集以外,该算法在其他三个数据集上的查准率 P 达到

最高值，因此可看出 *ASRAdaBoost* 算法在三个性能评估指标上，较之 Ori 组与 A2 组有更为优秀的结果，提高了软件缺陷预测的分类学习性能。在针对本实验选用的四个类不平衡数据集的解决上，由于 ASRAdaBoost 算法首先采用特征选择与组合采样结合的数据预处理方法对不平衡数据集进行少数类与多数类的比例平衡，降低了整体数据集的不平衡度，再进一步结合集成算法 AdaBoost，提升了由多个 J48 弱分类器集合而成的强分类器的准确性。因此，上述实验预测分类结果表明，本文 ASRAdaBoost 算法的软件缺陷预测分类效果优于其设计对照组，在针对类不平衡的数据集上处理效果明显，缺陷预测的分类性能更加优秀。

5 结束语

本文在针对软件缺陷预测中类不平衡数据集方面，构建了结合特征选择、组合采样以及集成算法 AdaBoost 的软件缺陷预测模型 ASRAdaBoost。该算法模型在融合卡方检验特征选择、经典过采样 SMOTE 方法、简单采样 resample 方法，通过去除冗余属性、增加少数类实例，有效平衡了数据集的类比例。在选取的四个 UCI 类不平衡数据集上进行实验，实验结果表明 ASRAdaBoost 模型的软件缺陷预测性能有明显提升，对于类不平衡问题有较好的解决能力。本文实验中均采用 J48 分类器作为基分类器，如今广泛适用的集成算法有很多种，如 Bagging 算法等。因此进一步研究将侧重于探讨在软件缺陷预测分类中应用不同分类器进行学习与比较，以及如何结合其他经典集成算法以提升软件缺陷预测性能。

参考文献:

[1] 王青，伍书剑，李明树. 软件缺陷预测技术 [J]. 软件学报，2008，19(7)：1565-1580.

[2] Weiss G M. Mining with rarity: a unifying framework [J]. ACM SIGKDD Explorations Newsletter，2004，6(1)：7-19.

[3] 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.

[4] Tahir M A，Kittler J，Yan Fei. Inverse random under sampling for class imbalance problem and its application to multi-label classification [J]. Pattern Recognition，2012，45(10)：3738-3750.

[5] Freund Y，Schapire R E. A desicion-theoretic generalization of on-line learning and an application to boosting [J]. Journal of Computer and System Sciences，1997，55(1)：119-139.

[6] Breiman L. Bagging predictors [J]. Machine Learning，1996，24(2)：123-140.

[7] Fan Wei，Stolfo S J，Zhang Junxin，*et al*. AdaCost: misclassification cost-sensitive boosting [C]//Proc of the 16th International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers Inc，1999：97-105.

[8] Kai M T. An empirical study of MetaCost using boosting algorithms [C]//Proc of the 11th European Conference on Machine Learning. Berlin: Springer，2000：413-425.

[9] 张栋，王勇，蔡立军. 基于单类别学习的自适应数据流分类算法 [J]. 西北工业大学学报，2010，28(5)：713-717.

[10] 张晓风，张德平. 基于不平衡数据集的软件缺陷预测 [J]. 计算机应用研究，2017，34(7)：2027-2031.

[11] 熊婧，高岩，王雅瑜. 基于 AdaBoost 算法的软件缺陷预测模型 [J]. 计算机科学，2016，43(7)：186-190.

[12] Liu Mingxia，Miao Linsong，Zhang Daoqiang，*et al*. Two-stage cost-sensitive learning for software defect prediction [J]. IEEE Trans on Reliability，2014，63(2)：676-686.

[13] Khoshgoftaar T M，Gao Kehan，Hulse J V. Feature selection for highly imbalanced software measurement data [M]// Recent Trends in Information Reuse and Integration. Berlin: Springer，2012：167-189.

[14] Guyon I，Elisseeff A. An introduction to variable feature selection [J]. Journal of Machine Learning Research，2003，3：1157-1182.

[15] 曹莹，苗启广，刘家辰，等. AdaBoost 算法研究进展与展望 [J]. 自动化学报，2013，39(6)：745-758.

[16] Shanab A A，Khoshgoftaar T M，Wald R，*et al*. Comparison of approaches to alleviate problems with high-dimensional and class-imbalanced data [C]// Proc of IEEE International Conference on Information Reuse and Integration. Piscataway，NJ: IEEE Press，2011：234-239.

[17] 周志华. 机器学习 [M]. 北京：清华大学出版社，2016.

[18] Gao Sheng，Lee C H，Lim J H. An ensemble classifier learning approach to ROC optimization [C]// Proc of the 18th International Conference on Pattern Recognition. Piscataway，NJ: IEEE Press，2006：679-682.

[19] Fawcett T. An introduction to ROC analysis [J]. Pattern Recognition Letters，2006，27(8)：861-874.

[20] 李勇. 结合欠抽样与集成的软件缺陷预测 [J]. 计算机应用，2014，34(8)：2291-2294，2310.

(上接第 2805 页)

[7] Fischer-Hellmann K P，Bleimann U，Fuhrmann W，*et al*. Analysis of security-relevant semantics of BPEL in cross-domain defined business processes [J]. Information Management & Computer Security，2007，15(2)：116-127.

[8] Ferrara A. Web services: a process algebra approach [C]//Proc of the 2nd International Conference on Service Oriented Computing. 2004：242-251.

[9] 王莉，刘厚泉，吴雪峰. 基于 BPEL 的业务流程管理系统架构的研究与应用 [J]. 计算机工程与设计，2006，27(18)：3507-3510.

[10] Sapiecha K，Grela D. Test scenarios generation for a class of processes defined in the BPEL language [J]. Annales UMCS Informatica，2008，8(2)：75-87.

[11] 赵会群，孙晶. 网购软件体系结构代数模型 [J]. 中国科学：信息科学，2013，43(1)：161-177.

[12] Alves A，Arkin A，Askary S，*et al*. Web services business process execution language version 2.0 [J]. Medicina，2007，44(1)：64-71.

[13] Bolognesi T，Brinksma E. Introduction to the ISO specification language LOTOS [J]. Computer Networks and LSDN Systems，1987，14(1)：25-59.

[14] 王继曾，张键. 基于 LOTOS 形式规范的目标实现 [J]. 计算机工程，2005，31(12)：97-99.

[15] Garavel H，Lang F，Mateescu R. Compositional verification of asynchronous concurrent systems using CADP [J]. Acta Informatica，2015，52(4-5)：337-392.

[16] 蔡文华，徐洪珍. 面向服务的构件可信演化策略 [J]. 计算机应用与软件，2016，33(4)：11-13，55.

[17] 李越甲. 基于 SOA 架构的需求管理系统的设计与实现 [D]. 南京：南京大学，2016.

[18] 王权于，吕国斌，应时，等. 一种策略驱动的 BPEL 流程异常处理框架 [J]. 计算机科学，2015，42(1)：180-186，192.

[19] 张亚. 服务组合 BPEL 测试序列生成研究 [J]. 计算机科学，2017，44(1)：203-204，22.

[20] 孟若. 基于 AOP 的 Web 服务演化方法 [D]. 武汉：武汉工程大学，2015.