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严重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，提出了有效提升分类效果的软件缺陷预测模型（ASRAAdaBoost）。该算法在数据预处理阶段采用特征选择与组合采样技术，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ASRAAdaBoost。实验均采用J48决策树作为基分类器，实验结果表明ASRAAdaBoost算法模型在提升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

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 (ASRAAdaBoost) to effectively improve the classification. The algorithm was based on the 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 attribute redundancy problems, using the AdaBoostensemble algorithm to build a software defect prediction model ASRAAdaBoost eventually. The experimental results show that the ASRAAdaBoost 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，则高风险模块占少数，而且在高风险模块中具体类的错误率远高于低风险模块，由于真实世界中大多数软件系统中低风险模块占绝大多数，所以对高风险模块的预测准确性会无实际意义。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


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（2013BEH19F01）

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 UC-I 数据集进行实验验证。本文的实验结果表明：所提出的方法 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[1]，因此去除数据集中不相关以及冗余属性，能够使得模型算法具有最好性能。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，如向前搜索、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) 生成选子集的特征评估。当候选特征子集生成后，需要评估标准对其评估，评估方法大致分为 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。

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，停止标准的设计与评估准则以及软件的具体应用需求有较大关联，作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)

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 算法是较为经典的一种过采样算法，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。对于类不平衡数据集中的每一个少数类样本，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)，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)

通过去除多数类样本的数量以达到数据集中类的比例平衡。在软件缺陷预测的数据预处理阶段，随机去除正常样本，完全保留有软件缺陷标记的样本，以此解决数据不平衡问题，其缺点是在删除部分少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[11]。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增加一个新的弱分类器，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，分别根据分类错误率选择合适阈值并且调整样本权重。

c) 根据新的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，对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，随后展开本算法的整体实验验证过程，并进行实验对组合采样方法的实验优势，该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独立性，即判断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强，反之，代表两个变量的相互独立性越差。假设自变量有 N 种取值，因变量有 M 种取值，由于卡方值的计算可以表示为

\[ \chi^2 = \sum \frac{(O - E)^2}{E} \]  

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元素独立性去除，而后利用监督学习中的 SMOTE 过抽样方法以增加少数样本样本，并结合使用 resample 简单采样技术去除多余的多数类样本样本。本文的组合采样方法，最终根据 AdaBoost 算法模型的 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。该算法流程图如图 2 所示。
性能指标的各项分类预测性能评价指标基本均优于原始数据。

本实验设计对照组分为以下三种：
- A1，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。
- A2，对原始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。
- A3，对原始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 1 所示。

表 1 设置相应过采样率与欠采样率的数据集信息

<table>
<thead>
<tr>
<th>数据集名称</th>
<th>过采样率/%</th>
<th>欠采样率/%</th>
<th>原样本总个数</th>
<th>现样本总个数</th>
</tr>
</thead>
<tbody>
<tr>
<td>credit-g</td>
<td>55.60</td>
<td>42.88</td>
<td>1 000</td>
<td>499</td>
</tr>
<tr>
<td>diabetes</td>
<td>24.38</td>
<td>46.10</td>
<td>1 000</td>
<td>768</td>
</tr>
<tr>
<td>hepatitis</td>
<td>156</td>
<td>37.7</td>
<td>1 000</td>
<td>155</td>
</tr>
<tr>
<td>ionosphere</td>
<td>19.05</td>
<td>46.79</td>
<td>1 000</td>
<td>351</td>
</tr>
</tbody>
</table>

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 个数据集本身是多类不平衡数据集。本实验根据数据集的属性个数、样本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：
- credit-g，即 Statlog-German Credit Date，它是德国信用卡数据，用于预测个人信用是否良好，信用分为 good 与 bad 两类，代表信用好与坏，其中 bad 类为少数类。
- diabetes，即 Pima Indians Diabetes Dataset，它是使用医疗历史记录来预测比马印第安人近五年的糖尿病发病率，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。
- hepatitis，即肝炎数据集用于预测肝炎发病情况，二分类为 DIE 与 LIVE，代表肝炎病状结果的死亡与存活，其中 DIE 为少数类。
- 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。

实验采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

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

4.2 软件缺陷预测性能评价指标

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价指标，并且对于本文探讨的二分类问题，可将数据样集的输出结果按实际所属类别与分类器预测分类的类别组合划分为正例(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 + FN + FP + TN = 数据集样本总数的数量，其中将少数类作为正例(即代表软件中缺陷的类别)，同时可将分类结果用混淆矩阵表 6 表示。

表 6 混淆矩阵

<table>
<thead>
<tr>
<th>实际情况</th>
<th>预测分类结果</th>
</tr>
</thead>
<tbody>
<tr>
<td>正例(有缺陷)</td>
<td>TP</td>
</tr>
<tr>
<td>反例(无缺陷)</td>
<td>FN</td>
</tr>
</tbody>
</table>

查准率 \( P = \frac{TP}{TP + FP} \) (2)

查全率 \( R = \frac{TP}{TP + FN} \) (3)

在实际情况下，软件缺陷预测模型目标是在现有条件下，最大程度找到含有缺陷的软件模块，最小限度地将无缺陷模块
件模块出现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，才能够全面地反映软件缺陷预测分类准确性的评价，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估指标。其中 F-measure 值与 ROC 值的定义如下：

\[
F \text{- measure} = \frac{1 + \beta^2}{\beta^2 \cdot P + R} \times P \times R
\]

F-measure 值是基于查准率 \( P \) 与查全率 \( R \) 的加权平均值，综合了查准率与查全率的评价结果，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 \( \beta \) 度量的是查准率与查全率的相对重要性，在本文中 \( \beta \) 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性能评估不同的参数值变化，根据不同的分类结果对不同样本进行比较，依次顺序将样本逐一对正例进行预测并计算，将其计算结果作为横坐标区段，即为得到 ROC 曲线。其横坐标轴为假阳性率（false positive rate, FPR），而纵坐标轴为查准率（true positive rate, TPR），两者分别定义为

\[
\text{FPR} = \frac{FP}{TN + FP}
\]

\[
\text{TPR} = \frac{TP}{TP + FN}
\]

在实际情况中，采用 ROC 曲线与坐标轴围成的下方面积 AUC 值作评估的评价指标以代替 ROC 曲线，AUC 值综合了软件缺陷预测方面的误差率与预测概率[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。

![ROC曲线与AUC](image)

4.3 实验设计与结果分析

本文的实验平台采用 weka 看卡托智能分析环境，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 0.05。组合采样技术运用经典过采样 SMOTE 方法与 resample 简单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 算法的阈值设置为 5，本文的组合采样技术将样本总数调整至原数据集数据的一半，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 值来体现在。针对 ASRAdaBoost 算法的实验设计对组成分为以下三种：

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。先对原始样本进行卡方检验特征选择，然后再对执行 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进行分类预测。

以上三组对照实验结果如表 7 - 9 所示，实验结果对比如图 7 - 9 所示。

<table>
<thead>
<tr>
<th>数据集名称</th>
<th>Ori</th>
<th>A2</th>
<th>ASRAdaBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>credit-g</td>
<td>0.511</td>
<td>0.690</td>
<td>0.728</td>
</tr>
<tr>
<td>diabetes</td>
<td>0.632</td>
<td>0.808</td>
<td>0.811</td>
</tr>
<tr>
<td>hepatitis</td>
<td>0.667</td>
<td>0.783</td>
<td>0.778</td>
</tr>
<tr>
<td>ionosphere</td>
<td>0.929</td>
<td>0.953</td>
<td>0.969</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>数据集名称</th>
<th>Ori</th>
<th>A2</th>
<th>ASRAdaBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>credit-g</td>
<td>0.442</td>
<td>0.661</td>
<td>0.728</td>
</tr>
<tr>
<td>diabetes</td>
<td>0.614</td>
<td>0.743</td>
<td>0.814</td>
</tr>
<tr>
<td>hepatitis</td>
<td>0.528</td>
<td>0.783</td>
<td>0.840</td>
</tr>
<tr>
<td>ionosphere</td>
<td>0.874</td>
<td>0.917</td>
<td>0.940</td>
</tr>
</tbody>
</table>

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 值对对比

图 9 Ori, A2 与 ASRAdaBoost 方法的 AUC 值对对比
tis 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 Ovt 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类别不平衡性，再进一步结
合集成算法 AdaBoost，提升了由多个 J48 弱分类器集成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结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，缺陷集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：
[5] Freund Y，Schapire R E。A desicion-theoretic generalization of online learning and an application to boosting [J]。Journal of Com-

puter and System Sciences，1997，55 (1) : 119-139。
[13] Khoshgoftaar T M，Gao Kehan，Hulse J V。Feature selection for high-
ly imbalanced software measurement data [M]//Recent Trends in Information Reuse and Integration。Berlin：Springer，2012；167-189。
[16] Shanah A A，Khoshgoftaar T M，Wald R，et al。Comparison of ap-
proaches to allocate problems with high-dimensional and class-imbal-
[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 ap-
proach 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 Recogni-
tion Letters，2006，27 (8) : 861-874。
[20] 李勇。基于非规则与集成的软件缺陷预测 [J]。计算机应用，2014，34 (8) : 2291-2294，2310。
[21] Bolognesi T，Brinkema E。Introduction to the ISO specification lan-
guage LOTOS [J]。Computer Networks and LSDN Systems，1987，14 (1) : 25-59。
[22] 王同，张健，基于 LOTOS 形式化研究的目标实现 [J]。计算机工
程，2005，31 (12) : 97-99。
[23] Garavel H，Lang F，Mateescu R。Compositional verification of asyn-
chronous concurrent systems using CAPD [J]。Acta Informatica，2015，52 (4-5) : 337-392。
[24] 软件缺陷预测的分
类及应用 [J]。软件学报，2009，20 (6) : 1101-1113。
[25] 王勇，刘建平。基于 SOA 架构的需求管理系统的评估与实
现 [J]。软件学报，2016，27 (8) : 203-204，22。