文章编号：2095-2163(2022)03-0139-04 中图分类号：TP311 文献标志码：A

基于 ExtraTree 的软件缺陷预测方法研究

王馨煜，崔艺凝，段盈盈
（北京信息科技大学 计算机学院，北京 100101）

摘 要：软件缺陷预测技术可以识别出软件存在缺陷的模块，提高软件的质量和安全性能，降低开发成本。针对不同模型预测结果差异性较大的问题，本文对结构复杂和缺乏历史数据的静态软件缺陷模块采用了基于极度随机树的软件缺陷预测方法进行研究，使用合成少数类过采样技术对原始数据集进行基本处理；用 5 种单分类器模型对软件缺陷数据集分别进行预测；最后，基于极度随机树集成各弱分类器，利用集成分类器对软件缺陷模块进行预测。在 NASA MDP 基础数据集上进行验证实验表明，将极度随机树方法应用于软件缺陷预测，具有良好的缺陷预测性能。

关键词：缺陷预测；分类器；极度随机树

Research on software defect prediction based on ExtraTree

WANG Xinyu, CUI Yining, DUAN Yingying
(School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China)

【Abstract】 Software defect prediction technology can identify the software defect modules, improve the quality and safety performance of software, reduce the development cost. Aiming at the problem that the prediction results of different models differ greatly, this paper adopts the software defect prediction method based on ExtraTree to study the static software defect module with complex structure, class imbalance and lack of historical data, and uses the synthetic minority class oversampling technology to perform basic processing on the original data set. Five single classifier models are used to predict software defect data sets respectively. Finally, the weak classifiers are integrated based on the ExtraTree, and the software defect modules are predicted by the ensemble classifier. The validation experiments on NASA MDP data set show that the application of the ExtraTree method to software defect prediction has good performance.

【Key words】 defect prediction; classifier; ExtraTree

0 引 言

随着时代的发展和科技进步，计算机在人们的生活中越来越多地被使用。软件是计算机领域中非常重要的一部分，软件存在的缺陷也不可小觑。软件缺陷预测技术旨在预测出模型中的缺陷数和缺陷倾向性，从而根据预测结果对资源进行合理的分配，是缺陷检测技术的重要辅助手段。早期，研究人员通过经验来估计模型中可能存在的缺陷；后来出现了软件体积度量元和缺陷的关系式，用关系式来计算系统在测试之前存在的缺陷数；有研究者将代码对应具体文档位置，从而给出了缺陷率的公式；也有研究者假设模块规模符合指数分布，给出了缺陷密度的估算公式。

融合多分类器模型对软件缺陷预测技术有重大的研究意义，通过融合多分类器模型，不仅可以发现不同模型之间潜在的联系，还可以度量软件的可靠性。另外，融合多个效果较弱的分类器为一个性能较好的多分类器，还可以提高弱分类器的预测性能。

本文首先通过选择不同的分类器模型对提取的软件模块进行预测并输出结果；其次，对单个分类器模型与融合后的分类器模型的预测结果进行比对；采用基于集成学习的静态软件缺陷预测方法对软件模块缺陷进行预测。

1 相关研究

1.1 背景

软件缺陷预测技术旨在预测出软件模块的缺陷，明确存在缺陷模块的缺陷数和缺陷倾向性。软件模块存在缺陷可能会造成财产损失和安全隐患。如：1996 年 6 月因导航系统的计算机软件故障导致欧洲“阿丽亚娜”号航天飞机坠毁；1999 年美国火星探测飞船坠毁事件，不包括损失时间，其工程成本耗

基金项目：北京信息科技大学 2021 年大学生创新创业训练计划资助(5102110805)。

作者简介：王馨煜(2000-)，女，本科生，主要研究方向：软件工程；崔艺凝(2001-)，女，本科生，主要研究方向：软件工程；段盈盈(2000-)，女，本科生，主要研究方向：软件工程。

收稿日期：2021-11-09

费 3.27 亿美元。软件缺陷预测技术是避免软件运行故障,减少不必要损失的重要手段,该技术自开始研究后就受到了众多的关注。

早期的软件缺陷通过员工的经验来估计,后来 Akiyama 明确给出了最早的软件缺陷与代码行的关系量化式[1],但只是在程序开始前初步对可能存在的软件缺陷进行估算,并不完美。随着测试软件的规模与其复杂度的逐步提高,开发者更加重视的是软件缺陷预测技术的精准度及模块测试的正确率是否能够保证在一个稳定的范围里,是否可以更加高效地完成测试。成熟的软件缺陷预测技术可以在软件发布之前预测出真正有缺陷的程序模块,从而提高软件的质量,减少资源消耗。

1.2 国内外研究现状

目前,国内外研究人员从不同的角度研究了静态缺陷预测和数据驱动缺陷预测等方法。在异常值检测和处理、高维度数据、类不平衡问题和数据差异等方面进行了研究,主要使用机器学习和统计方法来预测缺陷模块[2]。如:Freund 和 Schapire[3] 研究的 Adaboost 迭代算法,可以增强预测模型的精度;Wolpert[4] 提出的 Stacking 算法,可以集成若干基分类器的分类性能,从而提高分类效果等。

越来越多的预测模型的出现使研究者的注意力更多地集中在模型预测精度上,实验数据集的差异性和单一分类器预测性能的局限性是影响软件缺陷预测精度的两大原因。针对数据集的差异性,Sun 等[5] 提出了通过特征选择提高预测精度的方法;Xu 等[6] 提出了 Logistic 方法通过寻找最佳拟合参数来提高预测效率;针对单一分类器预测性能局限性的问题,Zhu[7] 等人提出了无监督的特征选择方法。除此之外,集成学习方法也是解决单一分类器的预测性能不够泛化问题的重要途径,通过将多个弱分类器集成为一个强分类器,进而提高软件缺陷预测的性能。

1.3 本文研究内容

本文针对不同预测模型对软件缺陷预测结果差异性较大的问题,对结构复杂、类别不平衡、缺乏历史数据的静态软件缺陷模块采用基于集成学习的软件静态缺陷预测方法,利用已有的缺陷数据集,选择 Extra-Trees (极度随机树)来将多个弱分类器集成,并通过实验对多个分类模型进行了验证,并对融合前后各个模型的预测结果进行了比对。

在实验中使用 SMOTE 方法(Synthetic Minority Oversampling Technique)对数据集进行预处理,选择 5 种基分类器并结合 Extra-Trees 集成方法进行验证。为了能够有效评价分类结果,本文选择了准确率、召回率、F 值 3 个业界认可的评价指标对预测结果进行评价。

2 相关理论与技术基础

2.1 SMOTE 采样

为解决样本少,特征缺失的问题,Chawla 等人提出了 SMOTE 过采样方法,可以减少模型的过拟合。在训练模型时,样本数量少的类所能提供的信息也比较少,SMOTE 方法通过对少数类样本的分析,将少数类样本合成新的样本并加入数据集中,重复分析、合成过程直到达到数据样本平衡[8]。

生成新样本的方法如式(1)所示。

$$p_j = x + rand(0,1) \times (y - x), j = 1,2,3,\cdots,N \quad (1)$$

其中,P_j 表示新样本;x 是少数类样本点,对于每一个少数类样本 x,从其 k 近邻中随机选择若干个样本,假设选择的近邻为 y;N 表示生成样本数量。

2.2 基于 ExtraTree 的缺陷预测方法

集成学习是解决类不平衡问题的方法之一,从数据中显式或隐式地学习多个模型,将这些模型有效结合,得到可靠、准确的预测。单一分类器模型的测试能力逐渐趋于饱和,并且对缺陷模块预测的范围并不具有广泛性,通过结合多个单一学习器,并聚合其预测结果的学习任务,聚集多个分类方法来提高分类的精度,可以获得比单一学习器更显著的泛化性能,也可以称作多分类系统。

目前,集成学习的主要问题就是如何将多个弱分类器合成一个强分类器,有效提高预测的精度。在实验中采用了 ExtraTree(极度随机树)来集成多分类器模型,但使用这种方法前需要检查样本的数据是否适用 ExtraTree 缺陷预测方法。ExtraTree 具有很少的关键超参数和用于配置这些超参数的合理启发式方法,能够处理很高维度的数据。相比于从训练数据集的引导样本开发每个决策树的随机森林,ExtraTree 更适合整个训练数据集上的每个决策树,每个决策树都采用原始训练集,不会随机采样,训练速度更快。

2.3 分类器模型评价指标

软件缺陷预测模型可用于对软件模块的缺陷情况作分类处理,评价指标用于区分预测模型的优劣。在本次实验中选取了软件缺陷预测常用的评价指标:准确率(*Accuracy*),精确率(*Precision*),召回率(*Recall*) 以及 $F1$。

准确率（*Precision*）又叫查准率，是被正确预测出的有缺陷的样本数量与被预测为无缺陷的样本数量之比，如式(2)：

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

召回率（*Recall* = *TPR*），又叫查全率，也就是被正确预测出的有缺陷的样本数量与实际有缺陷的样本数量之比，如式(3)：

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

*F*1 可以看作是模型精确率和召回率的一种调和平均，如式(4)：

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision} \tag{4}$$

其中，*TP* 表示被正确预测出的有缺陷的样本数量；*FP* 表示被预测为有缺陷的无缺陷样本数量；*FN* 表示被预测为无缺陷的有缺陷样本数量；*TN* 表示被正确预测出的无缺陷样本数量。

3　实验与结果分析

3.1　数据集

本实验采用数据集为 NASA 公布的 MDP 软件缺陷数据集，来自于十三个实际软件项目，数据集的基本信息包括样本集名称、模块总数、缺陷模块数、属性个数以及缺陷所占比例，不同数据集缺陷所占比例不同。从 NASA MDP 数据集中选取缺陷所占比例不同的数据子集 KC1、KC3、MC2、MW1、PC1、PC3、PC4 作为本文的实验数据集，见表 1。

表 1　NASA MDP 数据子集

Tab. 1　NASA MDP Subset of the data

数据集	开发语言	模块总数	缺陷模块数	属性个数	缺陷模块所占比例/%
KC1	C++	2 017	325	22	16.11
KC3	Java	458	43	40	9.39
MC2	C	161	52	38	32.3
MW1	C	403	31	38	7.7
PC1	C	1 031	76	37	7.37
PC3	C	1 563	160	38	10.24
PC4	C	1 458	178	38	12.21

3.2　实验方法

选择的是决策树分类器、随机森林分类器、梯度提升分类器、基于直方图的梯度提升分类器、自适应增强分类器 5 种基础模型，通过极度随机树的集成学习方法融合 5 个基础模型。

为了保证所对模型的数据对比的有效性，每个实验的过程是相同的，5 种基础模型以及极度随机树集成学习方法在 NASA 的数据子集上进行一次十字交叉验证，使 *Recall*、*Precision*、*F*1 3 个对比指标数据进行同一数据集不同模型的数值对比。

3.3　实验结果分析

5 个基础模型以及极度随机树集成学习方法在 7 个数据集中进行实验，得到的指标数据见表 2。从表 2 可以看出，极度随机树集成学习方法在 KC3、MC2、MW1、PC3 这 4 个数据集上达到了比其他 5 种基础模型更好的 *F*1 值，说明极度随机树对于特定数据集可以将弱分类器集成融合成一个较强分类器。随机森林分类器、基于直方图的梯度提升分类器、自适应增强分类器分别在 KC1、PC4、PC1 这 3 个数据集上 *F*1 值达到最佳，该现象与 KC1、PC4、PC1 3 个数据集的类不平衡有一定关系。

表 2　NASA MDP 数据集实验结果

Tab. 2　NASA MDP Experimental results of the data sets

数据集	评价指标	方法					
		决策树	随机森林	梯度提升	基于直方图的梯度提升	自适应增强	极度随机树
KC1	*Recall*	0.455	0.48	0.478	0.443	0.551	0.489
	Precision	0.439	0.499	0.474	0.542	0.363	0.479
	*F*1	0.437	0.481	0.471	0.476	0.433	0.451
KC3	*Recall*	0.336	0.378	0.366	0.372	0.412	0.353
	Precision	0.301	0.361	0.27	0.379	0.275	0.48
	*F*1	0.294	0.327	0.275	0.336	0.291	0.352
MC2	*Recall*	0.358	0.375	0.366	0.372	0.412	0.45
	Precision	0.289	0.387	0.263	0.379	0.275	0.565
	*F*1	0.303	0.328	0.272	0.336	0.291	0.478
MW1	*Recall*	0.3	0.386	0.366	0.372	0.412	0.483
	Precision	0.247	0.388	0.27	0.379	0.275	0.445
	*F*1	0.254	0.336	0.275	0.336	0.291	0.419
PC1	*Recall*	0.455	0.41	0.477	0.36	0.577	0.39
	Precision	0.389	0.54	0.5	0.32	0.445	0.44
	*F*1	0.389	0.393	0.408	0.316	0.483	0.393
PC3	*Recall*	0.365	0.43	0.366	0.372	0.412	0.429
	Precision	0.288	0.414	0.27	0.379	0.275	0.434
	*F*1	0.306	0.374	0.275	0.336	0.291	0.423
PC4	*Recall*	0.618	0.655	0.727	0.691	0.751	0.608
	Precision	0.522	0.602	0.59	0.633	0.546	0.663
	*F*1	0.552	0.619	0.638	0.651	0.622	0.629

（下转第 146 页）