Journal of Computer Applications
计算机应用,2018,38(3):824-828,858
ISSN 1001-9081
CODEN JYIIDU
2018-03-10
http://www.joca.cn

文章编号:1001-9081(2018)03-0824-05
DOI:10.11772/j.issn.1001-9081.2017081935

回归算法对软件缺陷个数预测模型性能的影响

付忠旺[1,2,3],肖 蓉[1,2],余 啸[2*],谷 懿[1]
(1.湖北大学 计算机与信息工程学院,武汉 430062; 2.软件工程国家重点实验室(武汉大学),武汉 430072;
3.湖北省教育信息化工程技术研究中心,武汉 430062)
(* 通信作者电子邮箱 xiaoyu-whu@yahoo.com)

摘 要:针对已有研究在评价软件缺陷个数预测模型性能时没有考虑到软件缺陷数据集存在数据不平衡的问题而采用了评估回归模型的不合适的评价指标的问题,提出以平均缺陷百分比作为评价指标,讨论不同回归算法对软件缺陷个数预测模型性能的影响程度。利用PROMISE提供的6个开源数据集,分析了10个回归算法对软件缺陷个数预测模型预测结果的影响以及各种回归算法之间的差异。研究结果表明:使用不同的回归算法建立的软件缺陷个数预测模型具有不同的预测效果,其中梯度Boosting回归算法和贝叶斯岭回归算法预测效果更好。

关键词:软件缺陷个数预测;数据不平衡;回归算法
中图分类号:TP181 **文献标志码**:A

Impact of regression algorithms on performance of defect number prediction model

FU Zhongwang[1,2,3], XIAO Rong[1,2], YU Xiao[2*], GU Yi[1]
(1. *School of Computer Science and Information Engineering, Hubei University, Wuhan Hubei* 430062, *China*;
2. *State Key Laboratory of Software Engineering (Wuhan University), Wuhan Hubei* 430072, *China*;
3. *Educational Informationalization Engineering Research Center of HuBei Province, Wuhan Hubei* 430062, *China*)

Abstract: Focusing on the issue that the existing studies do not consider the imbalanced data distribution problem in defect datasets and employ improper performance measures to evaluate the performance of regression models for predicting the number of defects, the impact of different regression algorithms on models for predicting the number of defects were explored by using Fault-Percentile-Average (FPA) as the performance measure. Experiments were conducted on six datasets from PROMISE repository to analyze the impact on the models and the difference of ten regression algorithms for predicting the number of defects. The results show that the forecast results of models for predicting the number of defects built by different regression algorithms are various, and gradient boosting regression algorithm and Bayesian ridge regression algorithm can achieve better performance as a whole.

Key words: defect number prediction; imbalanced data distribution; regression algorithm

0 引言

软件缺陷预测指的是通过从历史软件数据中学习出缺陷预测的模型,然后对新的软件模块进行预测,预测其是否有缺陷。如果预测该软件模块有缺陷则对该软件模块分配更多的软件测试人员,这样可以合理地分配测试资源。研究者已经提出了很多软件缺陷预测的方法[1-3]:陈翔等[4]总结了国内外在该研究领域取得的主要成果,但这些研究者提出的软件缺陷预测方法都是基于分类模型,即预测软件模块是否有缺陷;文献[5-6]指出,如果采用回归方法预测一个软件模块存在多少个缺陷时,可以优先测试缺陷个数多的模块,这样能够更好地分配测试资源。

举例来说,假如一个软件公司开发了一个包含有100个软件模块的新项目。由于项目交付时间提前,测试人员有限,在项目交付之前只能测试20个软件模块。因此,测试人员首先基于软件仓库中的历史软件模块数据建立了一个软件缺陷预测模型或者软件缺陷个数预测模型;然后利用预测模型预测这100个软件模块是否有缺陷或有多少个缺陷。假设缺陷预测模型预测这100个软件模块中30个软件模块有缺陷,由于在项目交付之前测试人员只能测试20个软件模块,因此测试人员不清楚应该测试这30个被预测为有缺陷的软件模块中的哪20个软件模块;但如果根据软件缺陷个数预测模型的预测结果,测试人员能够基于这100个软件模块的缺陷个数的预测值对这100个软件模块进行降序排序,优先测试前20个软件模块,即优先具有更多缺陷的软件模块,因此,预测软件缺陷个数相比单纯的预测软件模块是否有缺陷更利于优化软件测试资源的分配[7]。

目前在软件缺陷个数预测方面已有大量研究。Rathore等[8]探究了决策树回归算法在本项目缺陷个数预测模型和跨项目缺陷个数预测模型的预测性能,实验结果表明在采用绝对误差和相对误差作为评估指标时,决策树回归算法有很好的预测性能。Wang等[9]提出了利用历史数据构造缺陷状态转换模型,然后利用马尔可夫链预测将来每种状态下的缺陷个数。Afzal等[10]提出了利用基因编程算法来预测缺陷个

收稿日期:2017-08-07;**修回日期**:2017-09-22;**录用日期**:2017-10-18。

作者简介:付忠旺(1993—),男,山东聊城人,硕士研究生,主要研究方向:数据挖掘、软件工程; 肖蓉(1980—),女,湖北宜昌人,讲师,博士研究生,主要研究方向:软件工程; 余啸(1994—),男,湖北汉川人,博士研究生,主要研究方向:软件工程、深度学习; 谷懿(1996—),男,云南大理人,主要研究方向:机器学习。

数。Rathore 等[11] 提出了利用遗传算法和决策树回归算法来预测给定软件系统的缺陷个数的方法,在 PROMISE 提供的开源数据集上的实验结果表明该方法有较好的预测性能。

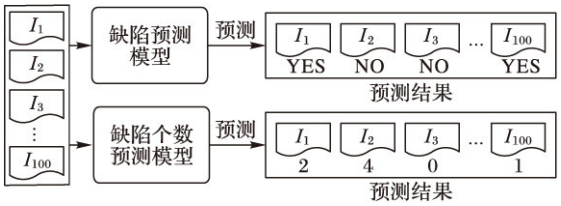

图 1　缺陷预测模型与缺陷个数预测模型的差异性

Fig. 1　Difference between a defect prediction model and a model for predicting the number of defects

Gao 等[12] 比较了泊松回归算法、零膨胀泊松回归算法和负二项回归算法对于预测软件缺陷个数上的能力,实验采用平均绝对误差(Average Absolute Error, AAE) 和平均相对误差(Average Relative Error, ARE) 作为评价指标,实验结果表明泊松回归算法实现了最好的预测性能。Chen 等[13] 比较了线性回归算法、贝叶斯岭回归算法、支持向量机回归算法、最近邻回归算法、决策树回归算法和梯度 Boosting 回归算法的预测性能; 实验采用均方回归误差和精度作为评价指标,实验结果表明在本项目缺陷个数预测和跨项目缺陷个数预测两种情形下,决策树回归算法都实现了最好的预测性能。在另一个比较相似的研究中,Rathore 等[14] 基于 PROMISE 提供的开源数据集上比较了遗传算法、多层感知机回归算法、线性回归算法、决策树回归算法、零膨胀泊松回归算法和负二项回归算法等六种回归算法对于预测软件缺陷个数的能力; 实验采用平均绝对误差和平均相对误差作为评价指标,实验结果表明线性回归算法和决策树回归算法对于大多数数据集而言产生的错误率最低并且相对于其他四种缺陷预测算法预测精准度更高。

但这些论文一般以评估回归模型的度量指标如均方根误差(Root Mean Square Error, RMSE) 、平均绝对误差或平均相对误差来评价软件缺陷个数预测模型的预测性能。这些度量指标反映了预测值偏离真实值的程度,其值越小,表示预测准确率越高。如平均绝对误差的计算公式为:

$$AAE = \frac{1}{n}\sum_{i=1}^{n} | y_{i,\text{predicted}} - y_{i,\text{actual}} |$$

其中: n 为测试集中软件模块的个数, $y_{i,\text{predicted}}$ 为测试集中第 i 个软件模块的缺陷个数预测值, $y_{i,\text{actual}}$ 为测试集中第 i 个软件模块的缺陷个数真实值。

但是由于软件缺陷数据集是极度数据不平衡的,即大多数软件模块的缺陷个数为 0,只有少数软件模块的缺陷个数大于 0。仅用评价回归模型的度量指标评估软件缺陷个数预测模型性能的好坏是不合适的。以本文 2.1 节中 Ant 1.3 这个软件缺陷数据集为例,该数据集包含 125 个软件模块,其中 20 个软件模块是有缺陷的,总共有 33 个缺陷。假如一个预测模型在预测该数据集时预测这 125 个软件模块的缺陷个数均为 0, AAE 值为 0.264(=33/125) 。这个预测模型取得了很低的 AAE, 但是这样的预测模型不能应用到实际的应用场景中,因为它不能预测出任何有缺陷的软件模块的缺陷个数。

文献[5]指出,由于缺乏高质量的训练数据,准确地预测出一个模块包含几个缺陷是比较困难的。实际上,一般这些软件缺陷个数预测方法都是利用预测出的缺陷个数来对软件模块进行排序,优先测试包含更多缺陷的软件模块。因此 Weyukers 等[15] 提出采用平均缺陷百分比(Fault Percentile Average, FPA) 评价指标来评估软件缺陷个数预测模型的性能。

针对文献[12 - 14]在比较不同的回归算法对软件缺陷个数预测模型性能影响的研究中采用了均方根误差、平均绝对误差或平均相对误差等不合适的评价指标,有可能产生错误的结论的问题,本文提出了以平均缺陷百分比为评价指标,利用 PROMISE 提供的 6 个开源数据集,分析了线性回归、决策树回归、贝叶斯岭回归、自相关决策回归、支持向量回归、梯度 Boosting 回归、高斯过程回归、最近邻回归、随机梯度下降回归和 Huber 回归这 10 个常用的回归算法对软件缺陷个数预测模型预测结果的影响以及各种回归算法之间的差异。实验结果表明: 梯度 Boosting 回归算法和贝叶斯岭回归算法预测效果最好。

本文主要工作为:

1) 以平均缺陷百分比为评价指标分析了回归算法对软件缺陷个数预测模型预测性能的影响。

2) 对比分析了 10 种常用的回归算法的差异,发现梯度 Boosting 回归算法和贝叶斯岭回归算法建立软件缺陷个数预测模型时具有最好的预测效果。

1　软件缺陷个数预测

1.1　软件缺陷个数预测流程

软件缺陷个数预测的流程如图 1 所示。第一步为从软件历史数据中提取出有用的软件模块,然后标记这些模块的特征和具有多少个缺陷。第二步为基于这些软件模块利用回归模型建立软件缺陷个数预测模型。第三步为对新的软件模块提取出特征,利用第二步中得到的软件缺陷个数预测模型预测这个新的软件模块的缺陷个数。

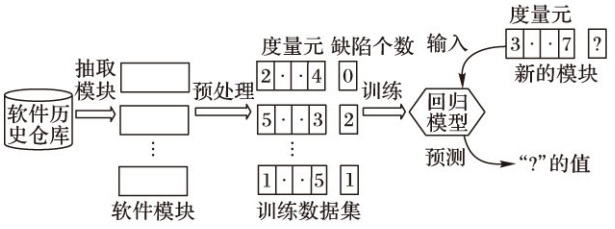

图 2　软件缺陷个数预测流程

Fig. 2　Flow chart of predicting the number of software defects

1.2　回归算法

1.2.1　线性回归

线性回归(Linear Regression, LR) [16] 是一种用于对因变量与一个或多个独立变量之间的线性关系进行建模的统计方法。一个线性回归方程为 $Y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n$, 其中: Y 是因变量, $x_1, x_2, \cdots, x_n$ 是独立变量, $b_1, b_2, \cdots, b_n$ 是独立变量的回归系数, b_0 是误差项。针对软件缺陷个数预测模型, Y 为缺陷个数, $x_1, x_2, \cdots, x_n$ 为软件模块的度量元。一般来说,线性回归都可以通过最小二乘法求出其方程。

1.2.2　决策树回归

决策树回归(Decision Tree Regression, DTR) [17] 通过学

习从数据特征推断的简单决策树来预测目标变量的值。决策树从根节点自上而下构建,并使用分割标准将数据分成包含具有相似值的实例的子集。选择最大化减少预期误差的属性作为根节点。该过程在非叶分支上递归运行,直到所有数据被处理。

1.2.3 贝叶斯岭回归

贝叶斯岭回归(Bayesian Ridge Regression, BRR)[18]假设先验概率、似然函数和后验概率都是正态分布。先验概率是假设模型输出 Y 是符合均值为 $X\theta$ 的正态分布,正则化参数 α 被看作是一个需要从数据中估计得到的随机变量。回归系数 θ 的先验分布规律为球形正态分布,超参数为 λ。贝叶斯岭回归通过最大化边际似然函数来估计超参数 α 和 λ,以及回归系数 θ。

1.2.4 自相关决策回归

自相关决策回归(Automatic Relevance Determination Regression, ARDR)[19]和贝叶斯岭回归很像,唯一的区别在于对回归系数 θ 的先验分布假设。自相关决策回归假设 θ 的先验分布规律为与坐标轴平行的椭圆形高斯分布。自相关决策回归也是通过最大化边际似然函数来估计超参数 α 和 λ 向量,以及回归系数 θ。

1.2.5 支持向量回归

支持向量回归(Support Vector Regression, SVR)[20]是支持向量在函数回归领域的应用。支持向量回归不同于支持向量机,支持向量回归的样本点只有一类,所寻求的最优超平面不是使两类样本点分得"最开",而是使所有样本点离超平面的总偏差最小,这时样本点都在两条边界线之间。

1.2.6 梯度 Boosting 回归

梯度 Boosting 回归(Gradient Boosting Regression, GBR)[21]以弱预测模型(通常是决策树)的形式产生预测模型,类似于其他 Boosting 方法,都是以阶段性方式构建模型,但与其他 Boosting 方法不一样的是,梯度 Boosting 回归在迭代时选择的是梯度下降的方向来保证最后的结果最好。

1.2.7 高斯过程回归

高斯过程回归(Gaussian Process Regression, GPR)[22]与贝叶斯岭回归类似,区别在于高斯过程回归中用核函数代替了贝叶斯岭回归中的基函数。高斯过程回归从函数空间角度出发,定义一个高斯过程来描述函数分布,直接在函数空间进行贝叶斯推理。

1.2.8 最近邻回归

最近邻回归算法(Nearest Neighbors Regression, NNR)[23]通过找出一个样本的 k 个最近邻,将这 k 个最近邻的回归值的平均值赋给该样本,就可以得到该样本的回归值。更有用的方法是将不同距离的邻居对该样本产生的影响给予不同的权值(weight),如权值与距离成正比。

1.2.9 随机梯度下降回归

随机梯度下降回归(Stochastic Gradient Descent Regression, SGDR)[24]是利用随机梯度下降的方法来最小化训练时回归方程中的误差的回归方法。

1.2.10 Huber 回归

相比线性回归算法采用最小二乘法求出其回归方程,异常点对回归模型的影响会非常大,传统的基于最小二乘的回归方法将不适用。Huber 回归(Huber Regression, HR)[25]并不会忽略异常点,而是给予它们一个很小的权重值,因此对异常点具有鲁棒性。

2 实验设置

2.1 数据集

实验数据集为从 PROMISE 库中的 6 个常用的项目,这 6 个项目的详细信息如表 1 所示。对项目中的每个软件模块共提取了 20 个特征,这 20 个特征的具体细节参考文献[3]。

表 1 实验数据集

Tab. 1 Experimental data set

项目	版本	模块数	缺陷总数	缺陷比例/%
Ant	Ant-1.3	125	33	16.0
	Ant-1.4	178	47	22.5
	Ant-1.5	293	35	10.9
	Ant-1.6	351	184	26.2
	Ant-1.7	745	338	22.3
Camel	Camel-1.0	339	14	3.4
	Camel-1.2	608	522	35.5
	Camel-1.4	872	335	16.6
	Camel-1.6	965	500	19.5
Jedit	Jedit-3.2	272	382	33.1
	Jedit-4.0	306	226	24.5
	Jedit-4.1	312	217	25.3
	Jedit-4.2	367	106	13.1
	Jedit-4.3	492	12	2.2
Synapse	Synapse-1.0	157	21	10.2
	Synapse-1.1	222	99	27.0
	Synapse-1.2	256	145	33.6
Xalan	Xalan-2.4	723	156	15.2
	Xalan-2.5	803	531	48.2
	Xalan-2.6	885	625	46.4
Log4j	Log4j-1.0	135	61	25.2
	Log4j-1.1	109	86	33.9

2.2 度量指标

在实验中,本文采用 FPA 评价指标度量软件缺陷个数预测模型的性能。假设 k 个软件模块以包含的缺陷个数的预测值的增序排列为 $f_1, f_2, \cdots, f_k$,这 k 个软件模块包含的缺陷个数的真实值为 $n_1, n_2, \cdots, n_k$。因此,最后的前 m 个软件模块 $(f_{k-m+1}, f_{k-m+2}, \cdots, f_k)$ 包含 $\sum_{i=k-m+1}^{k} n_i$ 个缺陷,这 m 个软件模块包含的缺陷对于全部 k 个软件模块包含的缺陷所占比例为 $\frac{1}{n}\sum_{i=k-m+1}^{k} n_i$,其中 $n = n_1 + n_2 + \cdots + n_k$。FPA 定义为 $\frac{1}{k}\sum_{m=1}^{k}\frac{1}{n}\sum_{i=k-m+1}^{k} n_i$。实际上,FPA 就是 m 取不同值时得到的缺陷百分比的平均值,因此 FPA 值越高,表示模型得到的排序越好。

2.3 实验流程

为了评估 1.2 节中介绍的 10 个回归算法的预测性能,实验采用 10 折交叉检验方法。进行实验时,对于表 1 中的项目,是将该项目的所有版本合并为一个数据集。然后将该数据集均分为 10 份,轮流将其中的 9 份作训练集,1 作做测试集,进行实验。每次实验在训练集上训练 1.2 节中介绍的 10 个回归算

法,然后在测试集上测试这 10 个回归算法的 FPA 值。最后返回 10 次的结果的 FPA 值的平均值。为了防止样本误差,本实验进行 20 次 10 折交叉检验,最后记录的实验结果为 20 次 10 折交叉检验的 FPA 均值。

2.4　研究问题

本文提出了以下两个研究问题,为软件缺陷个数预测模型中各种回归算法的选择提供了指导依据。

RQ1: 软件缺陷个数预测过程中,回归算法的选取是否影响预测效果?

RQ2: 采用哪一个回归算法得到的软件缺陷个数预测模型的预测效果更好?

3　实验结果分析

3.1　针对 RQ1

图 3 给出了 10 个回归算法在 6 个数据集上的预测结果的盒图,很容易看出预测结果分布有一定差别,其中 LR、BRR、ARDR、GBR 和 HR 这 5 种回归算法的效果较好 FPA 中位数分别为 0.764、0.763、0.763、0.761 和 0.744,而效果较差的 GPR 和 SGDR 其中位数仅为 0.449 和 0.501。BRR 取得了最大的 FPA 值为 0.813,GPR 取得了最低的 FPA 值为 0.385,BRR 在 6 个数据集上取得的最低的 FPA 值都比 GPR 在这 6 个数据集上取得的最高的 FPA 值都要高。因此从图 3 的盒图可以看出,对 RQ1 的回答是肯定的,即在软件缺陷个数预测过程中,回归算法的选取会影响预测效果。

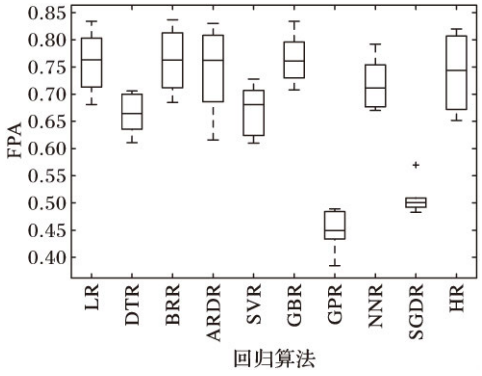

图 3　10 个回归算法下得到的 FPA 值对比

Fig. 3　Comparison of FPA values by 10 regression algorithms

为了进一步分析这 10 个回归算法对预测效果的影响的显著程度,本文也对这 10 个回归算法的预测结果采用 Wilcoxon 符号秩检验进行了假设检验。在 Wilcoxon 符号秩检验中,它把观测值和零假设的中心位置之差的绝对值的秩分别按照不同的符号相加作为其检验统计量。它检验成对产生观测数据的总体是否具有相同的均值。本文建立原假设 H0: 两种回归算法预测结果来自同一分布,即它们之间没有差别。因此在显著性水平为 0.05 的情况下,若检测的显著性水平大于 0.05,表示假设成立,接受 H0; 否则,假设不成立,拒绝 H0。由表 2 可见,LR 与除 BRR、ARDR、GBR、HR 外的算法显著性水平均小于 0.05,说明 LR 与除 BRR、ARDR、GBR、HR 外的算法均有较大差异。DTR 与 SVR 的显著性水平为 0.563,高于 0.05,说明 DTR 与 SVR 没有较大差异,但与除了 SVR 外的算法显著性水平均小于 0.05,说明 DTR 与除 SVR 外的算法有较大差异。BRR 与除 LR、ARDR、GBR 外的算法显著性水平均小于 0.05,说明 BRR 与除 LR、ARDR、GBR 外的算法具有较大差异。ARDR 与 DTR、SVR、GPR、SGDR 外的算法显著性水平均小于 0.05,说明 ARDR 与 DTR、SVR、GPR、SGDR 外的算法均有较大差异。SVR 与除 DTR、NNR、HR 外的算法显著性水平均小于 0.05,说明 SVR 与除 DTR、NNR、HR 外的算法具有较大差异。GBR 与除 LR、BRR、ARDR、HR 外的算法显著性水平均小于 0.05,说明 GBR 与除 LR、BRR、ARDR、HR 外的算法具有较大差异。NNR 与除 ARDR、SVR、HR 外的算法显著性水平均小于 0.05,说明 NNR 与除 ARDR、SVR、HR 外的算法具有较大差异。GPR 和 SGDR 与所有除自身以外的其他算法的显著性水平均低于 0.05,说明 GPR 和 SGDR 与其他算法都有较大差异。HR 与 LR、DTR、BRR、GPR、SGDR 的显著性水平均小于 0.05,说明 HR 与 LR、DTR、BRR、GPR、SGDR 都要较大差异。因此对第一个研究问题可以得出结论,回归算法的选取不仅对预测结果有影响,而且部分算法之间影响效果显著。

表 2　预测结果的 Wilcoxon 符号秩检验(显著性水平 0.05)

Tab. 2　Wilcoxon symbol rank test of prediction results (significance level 0.05)

算法	LR	DTR	BRR	ARDR	SVR	GBR	GPR	NNR	SGDR	HR
LR	—	0.031	0.438	0.219	0.031	0.668	0.031	0.031	0.031	0.063
DTR	0.031	—	0.031	0.031	0.563	0.031	0.031	0.031	0.031	0.031
BRR	0.438	0.031	—	0.063	0.031	0.688	0.031	0.031	0.031	0.031
ARDR	0.219	0.031	0.063	—	0.031	0.313	0.031	0.313	0.031	0.844
SVR	0.031	0.563	0.031	0.031	—	0.031	0.031	0.063	0.031	0.063
GBR	0.668	0.031	0.688	0.313	0.031	—	0.031	0.031	0.031	0.219
GPR	0.031	0.031	0.031	0.031	0.031	0.031	—	0.031	0.031	0.031
NNR	0.031	0.031	0.031	0.313	0.063	0.031	0.031	—	0.031	0.156
SGDR	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	—	0.031
HR	0.063	0.031	0.031	0.844	0.063	0.219	0.031	0.156	0.031	—

表 3　每个项目预测结果统计

Tab. 3　Statistics of each project's prediction result

数据集	LR	DTR	BRR	ARDR	SVR	GBR	GPR	NNR	SGDR	HR
Ant	0.801	0.660	0.805	0.799	0.686	0.796	0.385	0.754	0.570	0.789
Camel	0.726	0.669	0.721	0.726	0.707	0.757	0.484	0.679	0.509	0.672
Jedit	0.834	0.706	0.837	0.830	0.728	0.839	0.456	0.792	0.492	0.820
Synapse	0.713	0.636	0.712	0.686	0.624	0.730	0.442	0.670	0.506	0.698
Xalan	0.681	0.611	0.685	0.616	0.610	0.708	0.489	0.677	0.495	0.652
Log4j	0.803	0.700	0.813	0.808	0.676	0.765	0.434	0.744	0.483	0.807
AVG	0.759	0.664	0.762	0.744	0.671	0.766	0.448	0.719	0.509	0.740

3.2　针对 RQ2

从表 2 还可发现,LR、BRR、ARDR 和 GBR 两两之间没有显著性差异,而 HR 和 LR、ARDR、GBR、NNR 之间也没有显著性差异。这是从 6 个数据集整体分析得到的结果,但在单个数据集上 10 种算法对预测结果的影响是否依然如此,还需作进一步分析。因此本文统计了在每个数据集上 10 种回归算法的预测结果。表 3 表明,在数据集 Ant 和 Log4j 中 BRR 占优且在 Log4j 这个数据集上取得 FPA 的最大值为 0.813,而在数据集

Camel、Jedit、Synapse 和 Xalan 上 GBR 占优,取得的 FPA 的最大值为 Jedit 的 0.839。在这 6 个数据集上,GBR 取得最优的平均值为 0.766,BRR 取得第二好的平均值为0.762,但这两个平均值相差不大。因此对第二个研究问题可以得出结论,采用梯度 Boosting 回归算法和贝叶斯岭回归算法得到的软件缺陷个数预测模型的预测效果更好。

4 讨论

针对 2.4 节中提出的两个研究性问题,本文通过实验回答了这两个研究性问题,但实验过程中也潜在一些有效性威胁,具体如下:

1) 实验选用的是 PROMISE 平台提供的 6 个数据集,虽然数据提供者 Jureczko[26] 曾表示这些数据在数据搜集过程中可能存在不足,但这 6 个数据集已经广泛地应用于软件缺陷个数预测的应用研究[8-13] 中。因此,但本文坚信本文的实验结果具有一定的可信性和可重复性。

2) 本文研究的 10 个回归算法均为较常见的回归算法,这些算法全部基于 Sklearn 包实现,算法参数使用 Sklearn 包中提供的默认参数,即本文没有对回归算法进行任何优化。

3) 本文采用了平均缺陷百分比这个评价指标来评价软件缺陷个数预测模型的性能好坏,其他的一些指标如代价有效性图[27] 也可以进行考虑。

5 结语

本文围绕软件缺陷个数预测展开研究,针对建立软件缺陷个数预测模型过程中回归算法的选择问题,分析了 10 个常见的回归算法对软件缺陷个数预测模型预测结果的影响以及各个回归算法之间的差异。研究结果表明:使用不同的回归算法建立的软件缺陷个数预测模型具有不同的预测效果,其中梯度 Boosting 回归算法和贝叶斯岭回归算法预测效果更好。

在后续的工作中,将进一步在更多的数据集上进行分析,验证本文得出的实验结果的一般性;此外,将理论上分析梯度 Boosting 回归算法和贝叶斯岭回归算法建立的缺陷数目预测模型预测效果较好的原因。

参考文献 (References)

[1] RAHMAN R, POSNETT D, DEVANBU P. Recalling the "imprecision" of cross-project defect prediction [C]// FSE '12: Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering. New York: ACM, 2012: Article No. 61.

[2] SHEPPERD M, BOWES D, HALL T. Researcher bias: the use of machine learning in software defect prediction [J]. IEEE Transactions on Software Engineering, 2014, 40(6): 603 - 616.

[3] 王星,何鹏,陈丹,等. 跨项目缺陷预测中训练数据选择方法[J]. 计算机应用,2016,36(11):3165 - 3169. (WANG X, HE P, CHEN D, et al. Selection of training data for cross-project defect prediction [J]. Journal of Computer Applications, 2016, 36(11): 3165 - 3169.)

[4] 陈翔,顾庆,刘望舒,等. 静态软件缺陷预测方法研究[J]. 软件学报,2016,27(1):1 - 25. (CHEN X, GU Q, LIU W S, et al. Survey of static software defect prediction [J]. Journal of Software, 2016,27(1):1 - 25.)

[5] YANG X, TANG K, YAO X. A learning-to-rank approach to software defect prediction [J]. IEEE Transactions on Reliability, 2015, 64(1): 234 - 246.

[6] FENTON N E, NEIL M. A critique of software defect prediction models [J]. IEEE Transactions on Software Engineering, 1999, 25(5): 675 - 689.

[7] MALHOTRA R. A systematic review of machine learning techniques for software fault prediction [J]. Applied Soft Computing, 2015, 27: 504 - 518.

[8] RATHORE S S, KUMAR S. A decision tree regression based approach for the number of software faults prediction [J]. ACM Sigsoft Software Engineering Notes, 2016, 41(1): 1 - 6.

[9] WANG J, ZHANG H. Predicting defect numbers based on defect state transition models [C]// ESEM '12: Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. New York: ACM, 2012: 191 - 200.

[10] AFZAL W, TORKAR R, FELDT R. Prediction of fault count data using genetic programming [C]// INMIC 2008: Proceedings of the 12th IEEE International Multitopic Conference. Piscataway, NJ: IEEE, 2008: 349 - 356.

[11] RATHORE S S, KUMAR S. Predicting number of faults in software system using genetic programming [J]. Procedia Computer Science, 2015, 62: 303 - 311.

[12] GAO K, KHOSHGOFTAAR T M. A comprehensive empirical study of count models for software fault prediction [J]. IEEE Transactions on Reliability, 2007, 56(2): 223 - 236.

[13] CHEN M, MA Y. An empirical study on predicting defect numbers [C]// Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering. Piscataway, NJ: IEEE, 2015: 397 - 402.

[14] RATHORE S S, KUMAR S. An empirical study of some software fault prediction techniques for the number of faults prediction [J]. Soft Computing, 2016, 21(24): 7417 - 7434.

[15] WEYUKER E J, OSTRAND T J, BELL R M. Comparing the effectiveness of several modeling methods for fault prediction [J]. Empirical Software Engineering, 2010, 15(3): 277 - 295.

[16] ASAI H T S U K. Linear regression analysis with fuzzy model [J]. IEEE Transaction on System, Man and Cybernetics, 1982, 12(6): 903 - 907.

[17] XU M, WATANACHATURAPORN P, VARSHNEY P K, et al. Decision tree regression for soft classification of remote sensing data [J]. Remote Sensing of Environment, 2005, 97(3): 322 - 336.

[18] HOERL A E, KENNARD R W. Ridge regression: biased estimation for nonorthogonal problems [J]. Technometrics, 1970, 12(1): 55 - 67.

[19] JACOBS J P. Bayesian support vector regression with automatic relevance determination kernel for modeling of antenna input characteristics [J]. IEEE Transactions on Antennas and Propagation, 2012, 60(4): 2114 - 2118.

[20] BASAK D, PAL S, PATRANABIS D C. Support vector regression [J]. Neural Information Processing — Letters and Reviews, 2007, 11(10): 203 - 224.

[21] ELITH J, LEATHWICK J R, HASTIE T. A working guide to boosted regression trees [J]. Journal of Animal Ecology, 2008, 77(4): 802 - 813.

(下转第 858 页)

[3] YANG J, WRIGHT J, HUANG T S, et al. Image super-resolution via spare representation [J]. IEEE Transactions on Image Processing, 2010, 19(11): 2861 - 2873.

[4] TIMOFTE R, de SMET V, van GOOL L. Anchored neighborhood regression for fast example-based super-resolution [C]// ICCV '13: Proceedings of the 2013 IEEE International Conference on Computer Vision. Washington, DC: IEEE Computer Society, 2013: 1920 - 1927.

[5] TIMOFTE R, de SMET V, GOOL L. A+: adjusted anchored neighborhood regression for fast super resolution [C]// ACCV 2014: Proceedings of the 2014 Asian Conference on Computer Vision, LNCS 9006. Berlin: Springer-Verlag, 2014: 111 - 126.

[6] YANG C Y, YANG M H. Fast direct super-resolution by simple functions [C]// ICCV '13: Proceedings of the 2013 IEEE International Conference on Computer Vision. Washington, DC: IEEE Computer Society, 2013: 561 - 568.

[7] DAI D, TIMOFTE R, van GOOL L. Jointly optimized regressors for image super-resolution [J]. Computer Graphics Forum, 2015, 34(2): 95 - 104.

[8] FREEDMAN G, FATTAL R. Image and video upscaling from local self-examples [J]. ACM Translations on Graphics, 2011, 30(2): Article No. 12.

[9] PROTTER M, ELAD M, TAKEDA H, et al. Generalizing the non-local-means to super-resolution reconstruction [J]. IEEE Transactions on Image Processing, 2009, 18(1): 36 - 51.

[10] MAIRAL J, BACH F, PONCE J, et al. Non-local sparse models for image restoration [C]// ICCV '09: Proceedings of the 2009 IEEE 12th International Conference on Computer Vision. Washington, DC: IEEE Computer Society, 2009: 2272 - 2279.

[11] DONG C, LOY C C, HE K, et al. Learning a deep convolutional network for image super resolution [C]// ECCV 2014: Proceedings of the 2014 European Conference on Computer Vision, LNCS 8692. Berlin: Springer-Verlag, 2014: 184 - 199.

[12] GKASNER D, BAGON S, IRANI M. Super-resolution from a single image [C]// ICCV '09: Proceedings of the 2009 IEEE 12th International Conference on Computer Vision. Washington, DC: IEEE Computer Society, 2009: 349 - 356.

[13] CANDOCIA F M, PRINCIPE J C. Super-resolution of images based on local correlations [J]. IEEE Transactions on Neural Networks, 1999, 10(2): 372 - 380.

[14] DONG W, ZHANG L, SHI G, et al. Nonlocally centralized sparse representation for image restoration [J]. IEEE Transactions on Image Processing, 2013, 22(4): 1620 - 1630.

[15] YOU X, XUE W, LEI J, et al. Single image super-resolution with non-local balanced low-rank matrix restoration [C]// ICPR 2016: Proceedings of the 2016 23rd International Conference on Pattern Recognition. Washington, DC: IEEE Computer Society, 2016: 1255 - 1260.

[16] XU J, ZHANG L, ZUO W, et al. Patch group based nonlocal self-similarity prior learning for image denoising [C]// ICCV '15: Proceedings of the 2015 IEEE International Conference on Computer Vision. Washington, DC: IEEE Computer Society, 2015: 244 - 252.

[17] TEKALP A M, OZKAN M K, SEZAN M I. High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration [C]// ICASSP '92: Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing. Washington, DC: IEEE Computer Society, 1992: 169 - 172.

[18] LU Y, IMANURA M. Pyramid-based super-resolution of the undersampled and subpixel shifted image sequence [J]. International Journal on Systems Technology, 2002, 12(6): 254 - 263.

[19] AHARON M, ELAD M, BRUCKSTEIN A. *K*-SVD: an algorithm for designing overcomplete dictionaries for sparse representation [J]. IEEE Transactions on Signal Processing, 2006, 54(11): 4311 - 4322.

This work is partially supported by the Guangdong Natural Science Foundation Doctoral Program (2014A030310169), the Natural Science Foundation Project of Guangdong Province (2016A030313703, 2016A030313713), the Science and Technology Project of Guangdong Province (2016B030305002), the Science and Technology Project of Guangdong Department of Transportation (Sci. and Tech. -2016-02-030).

XIANG Wen, born in 1993, M. S. candidate. His research interests include image processing, pattern recognition.

ZHANG Ling, born in 1968, Ph. D., professor. Her research interests include data mining, computer vision, wireless sensor.

CHEN Yunhua, born in 1977, Ph. D., lecturer. Her research interests include computer vision, pattern recognition, deep learning.

JI qiumin, born in 1992, M. S. candidate. Her research interests include image processing, pattern recognition.

(上接第 828 页)

[22] QUIÑONERO-CANDELA J, RASMUSSEN C E. A unifying view of sparse approximate Gaussian process regression [J]. Journal of Machine Learning Research, 2005, 6: 1939 - 1959.

[23] ALTMAN N S. An introduction to kernel and nearest-neighbor nonparametric regression [J]. The American Statistician, 1992, 46(3): 175 - 185.

[24] CARPENTER B. Lazy sparse stochastic gradient descent for regularized multinomial logistic regression [R]. [S.l.]: Alias-i, Inc., 2008: 1 - 20.

[25] HUBER P J. Robust regression: asymptotics, conjectures and Monte Carlo [J]. The Annals of Statistics, 1973, 1(5): 799 - 821.

[26] JURECZKO M, MADEYSKI L. Towards identifying software project clusters with regard to defect prediction [C]// PROMISE '10: Proceedings of the 6th International Conference on Predictive Models in Software Engineering. New York: ACM, 2010: Article No. 9.

[27] JIANG T, TAN L, KIM S. Personalized defect prediction automated software engineering [C]// ASE 2013: Proceedings of the IEEE/ACM 28th International Conference on Automated Software Engineering. Piscataway, NJ: IEEE, 2013: 279 - 289.

FU Zhongwang, born in 1993, M. S. candidate. His research interests include data mining, software engineering.

XIAO Rong, born in 1980, Ph. D. candidate, lecturer. Her research interests include software engineering.

YU Xiao, born in 1994, Ph. D. candidate. His research interests include software engineering, deep learning.

GU Yi, born in 1996, undergraduate. His research interests include machine learning.