

文章编号: 2095-2163(2021)07-0066-08

中图分类号: TP311

文献标志码: A

基于 mRMR 算法的 learning-to-rank 错误定位

李天舒, 舒挺, ANGWECH KEVIN

(浙江理工大学 信息学院, 杭州 310018)

摘要: 基于 learning-to-rank 技术构建频谱错误定位模型, 从而实现高效的程序错误定位是当前的研究热点。然而, 针对不同的程序和错误类型, 如何生成有效的程序频谱特征集来训练错误定位模型, 成为了极具挑战的问题。针对该问题, 应用 mRMR 算法生成程序频谱特征集, 提出一种 learning-to-rank 的错误定位新方法。该方法应用基因编程自动生成备选可疑度公式集, 并利用 mRMR 算法从中选取一组公式子集, 该子集中的可疑度公式具有与程序错误高相关且彼此之间低相关的特性。利用此可疑度公式子集结合程序频谱计算特征值输入机器学习算法, 从而构造错误定位模型。实验结果表明, 新方法不仅能够提高基于 learning-to-rank 错误定位的效率, 也优于 Naish1、Tarantula 等传统 SBFL 方法。

关键词: 错误定位; mRMR 算法; 基因编程; 排序学习

learning-to-rank fault localization based on mRMR

LI Tianshu, SHU Ting, ANGWECH KEVIN

(School of Informatics Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

[Abstract] It is a current research hotspot to build a efficient fault localization model based on learning-to-rank technique. However, for different programs and fault types, how to generate an effective spectrum feature set to build the fault localization model has become a very challenging problem. In response to this problem, the mRMR is used to generate spectrum feature set, and a new learning-to-rank fault localization method is proposed. LTRmR uses genetic programming to automatically generate a set of alternate suspiciousness formulas, the suspiciousness formulas in this subset have the characteristics of high correlation with program faults and low correlation with each other. Suspiciousness formula subset combined with the spectrum is used to calculate the feature value input into the machine learning algorithm, thereby constructing a fault localization model. Experimental results show that the new method can not only improve the efficiency of learning-to-rank fault localization, but also outperform SBFL methods such as Naish1 and Tarantula.

[Key words] fault localization; mRMR; genetic programming; learning-to-rank

0 引言

现如今软件结构日趋复杂, 导致软件出错的概率也越来越高。近年来, 由于软件出错而造成的重大事故也时有发生。2018年3月, 苹果公司高级工程师 Walter Huang 在一起因特斯拉自动驾驶系统出错而造成的事故中不幸离世, 引发全美关注^[1]。因此, 如何保证软件的高可靠性已成为吸引各方瞩目的一个重要问题。通常, 软件调试工作都是人工设置断点进而分析错误位置, 需要花费大量时间和精力。而错误定位是软件调试中的关键步骤之一, 研究表明, 错误定位成为近年来国内外学术界研究的热点问题。

目前, 学者们提出了许多自动化的方法用于错

误定位研究, 致力于提高软件错误定位的效率。基于程序切片的错误定位^[2]、基于程序频谱的错误定位^[3]和基于模型检测的错误定位^[4-5]都是主流的错误定位技术, 其中, 基于频谱的错误定位方法 (Spectrum-Based Fault Localization, SBFL) 是目前最有效的定位方法之一。简而言之, SBFL 是通过对比程序在不同颗粒度 (语句、谓词、分支、基本块) 下成功执行和失败执行的覆盖信息来确定可疑代码的位置。首先执行测试用例并收集测试用例执行结果和覆盖信息, 然后使用可疑度公式计算每条语句的可疑度值, 可疑度值越高, 代表这条语句是错误语句的可能性就越高。其中, 可疑度公式是 SBFL 技术的核心。然而, 单个可疑度公式只适合定位某种类型的程序结构和错误, 并不具备普适性。

基金项目: 浙江省自然科学基金 (LY17F020033); 国家自然科学基金 (61101111, 61572441)。

作者简介: 李天舒 (1994-), 男, 硕士研究生, 主要研究方向: 软件测试; 舒挺 (1979-), 男, 博士, 副教授, 主要研究方向: 计算机网络协议、软件测试、移动电子商务系统开发等; ANGWECH KEVIN (1989-), 女, 硕士研究生, 主要研究方向: 软件测试。

通讯作者: 舒挺 Email: shuting@zstu.edu.cn

收稿日期: 2021-03-27

learning-to-rank 是一种解决排序问题的技术^[6],近年来在自然语言、信息检索等领域应用较为广泛。利用 learning-to-rank 技术构建错误定位模型进行错误定位可进一步提高错误定位的精度。Kim 等人^[7]提出 PRINCE 错误定位框架,通过对待测程序提取程序静态特征(文件、函数和语句的依赖性和复杂性)和程序动态特征(可疑度公式),结合 learn-to-rank 技术并应用支持向量机算法构造错误定位模型,该方法经实验验证可显著提高错误定位效率。研究表明,擅长定位同类型程序或错误的技术之间具有高相关性^[8]。目前大多数的研究在构建错误定位模型特征集时只是将相关信息、相关技术简单集成,并没有考虑这些程序信息或错误定位技术之间的相关性。然而,将彼此低相关性的错误定位技术集成起来往往能够为错误定位模型提供不同方面的信息,从而提高错误定位的效率。

本文以错误定位模型的特征选取为出发点,开展深入研究,提出一种基于最大相关最小冗余算法(Max-Relevance and Min-Redundancy, mRMR)的 learning-to-rank 错误定位方法(Learning To Rank Fault Localization Based on mRMR, LTRmR)。首先将程序频谱作为基因编程算法的输入生成可疑度公式,再通过 mRMR 算法选取若干能够准确识别程序错误且彼此之间低相关的可疑度公式构建模型特征集,最后经过训练获得错误定位模型。由于 LTRmR 融合了基因编程和 learning-to-rank 技术的优势,错误定位的效率与性能将得到提升。

1 研究背景

1.1 SBFL 方法

SBFL 依赖于测试用例覆盖信息计算语句可疑度分数,根据可疑语句排序列表从上到下排查程序错误。SBFL 由 2 个重要成分构成:程序频谱和可疑度公式。程序频谱即反映程序运行轨迹不同方面信息的四元组数据 $(a_{ef}, a_{ep}, a_{nf}, a_{np})$ 。其中, a_{ef} 和 a_{ep} 分别表示在失败和成功的测试中语句执行的次数; a_{nf} 和 a_{np} 分别表示在失败和成功的测试中语句未执行的次数。目前已被提出的可疑度公式有很多,如 Ochiai、Tarantula、Jaccard 等,其中大多数公式皆由经验直觉或引自其他领域设计而成。经实例验证,很难通过传统 SBFL 技术进一步提高错误定位效率。XIE 等人^[9]提出新的理论框架并分析了 30 个可疑度公式的定位效率,但只证明了其中 5 个公式在理

论框架基础上具有最优性。研究人员发现,利用基因编程所设计的可疑度公式能够提高其对不同类型错误的适应性,Yoo^[10]使用基因编程设计可疑度公式并证明其效率始终优于传统公式。由此可见,传统公式不仅设计起来费时费力,而且具有很大的局限性,效率很难提高。

1.2 learning-to-rank 错误定位

learn-to-rank 也称为排序学习,是一种有监督的将机器学习技术应用在排序问题的学习方法,通过先验并且带有标注的样本训练模型解决排序问题。在 Zou 等人^[8]的研究中,通过集成不同错误定位技术进行错误定位,包括 SBFL 技术、基于变异的错误定位技术、动态切片技术等。除上述研究外,Xuan 等人^[11]将 25 个可疑度公式组合并应用 learn-to-rank 技术完成错误定位,实验结果证明该方法能够更快地定位程序错误。但是,上述研究只考虑选取何种特征构建错误定位模型,却忽略了特征间相关性对模型性能的影响。事实上,如果在特征集中存在冗余特征,会对错误定位模型的定准性产生负面效果。

2 LTRmR 方法

LTRmR 设计流程如图 1 所示。由图 1 可知, LTRmR 构建错误定位模型共分 3 个阶段:基于基因编程的特征生成、基于 mRMR 算法的特征集构造、错误定位模型训练。在特征生成阶段,使用基因编程技术实现可疑度公式自动化生成。第二阶段基于 mRMR 算法选取可疑度公式构造特征集,最后根据频谱输入和特征集训练错误定位模型,用于被测程序可疑语句的可疑度值计算和排序。LTRmR 输入为程序频谱和语句标签,错误语句位置已知,在语句标签中,错误语句标记为 1,正确语句标记为 0;输出为按可疑度值排序的程序语句列表,语句排名越高,该条语句导致程序出错可能性越大。

2.1 基于基因编程的特征生成

本文使用基因编程技术设计可疑度公式(GP-evolved),其目的是使用可疑度公式组合学习错误定位模型。算法通过交叉和变异等操作逐代演化出越来越优秀的种群,新一代的种群通常比上一代种群更加适应于环境。程序停止的条件是某单独个体达到预设的适应度值或者进化的代数达到预设参数值。在内存中,种群个体以树形结构表示,树的终端节点是操作数,非终端节点是运算符,如图 2 所示。

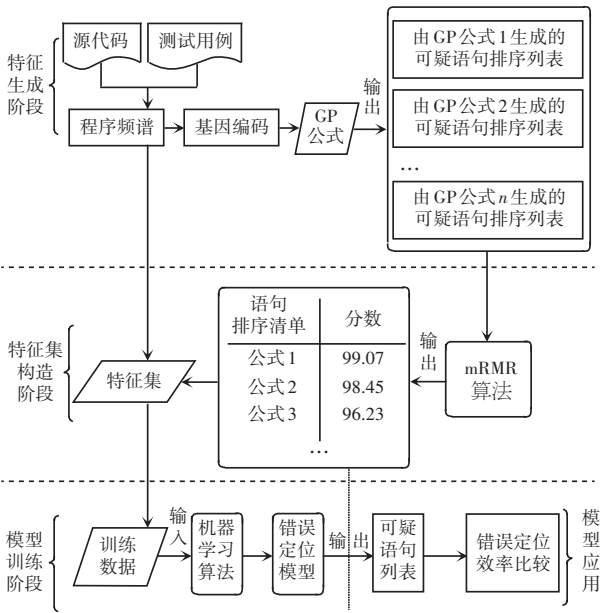


图1 LTRmR 流程图

Fig. 1 The workflow of LTRmR

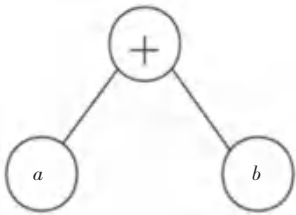


图2 基因表达式树形结构

Fig. 2 Genetic expression tree

基因编程算法参数和 GP 运算符参见表 1 和表 2^[10]。

表 1 基因编程算法程序参数

Tab. 1 List of experimental parameters

初始种群人口	最大树深	进化代数	交叉率	变异率
40	4	100	1.0	0.8

表 2 GP 算子

Tab. 2 List of GP operator

算子节点	定义
$gp_add(a, b)$	$a + b$
$gp_sub(a, b)$	$a - b$
$gp_mul(a, b)$	$a \times b$
$gp_div(a, b)$	$\begin{cases} 1 & \text{if } b = 0 \\ \frac{a}{b} & \text{otherwise} \end{cases}$
$gp_sqrt(a)$	$\sqrt{ a }$

适应度函数直接影响到基因编程算法的收敛速度以及能否找到最优解,以及最终结果与初始问题

期望输出之间的匹配程度。因此,适应度函数至关重要。本文适应度函数可写作如下形式:

$$E(t, p, b) = \frac{\text{Ranking of } b \text{ according to } t}{\text{Number of statement in } p} \times 100 \quad (1)$$

其中, t 是可疑度公式; b 是错误语句; p 是包含错误语句 b 的程序。

2.2 基于 mRMR 算法的特征集构造

作为错误定位模型构建过程中最核心的部分,如何选取可疑度公式构造模型特征集决定了能否提高模型对程序错误的预测准确性。本文通过集成学习可疑度公式构建错误定位模型,为了获得更好的模型错误定位能力,使用 mRMR 算法^[12]作为可疑度公式选择算法。mRMR 算法旨在原始特征集中找到与目标输出相关性最高,而特征彼此之间相关性最低的一组特征子集。mRMR 算法选取可疑度公式过程具体如下:

(1) 根据不同可疑度公式计算程序语句可疑度值。

(2) 将可疑度值作为算法输入,计算可疑度公式与语句标签之间的互信息值 $I(x_i; c)$,并基于式 (2) 计算互信息值平均值 $D(S, c)$:

$$D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i; c) \quad (2)$$

其中, x_i 为特征集 S 中第 i 个可疑度公式。

(3) 计算公式间最小冗余值 R 。其对应数学公式为:

$$R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i; x_j) \quad (3)$$

(4) 选取互信息值大于等于 D 值的 m 个可疑度公式,即选出与标签 c 相关性最高的 m 个公式。

(5) 剔除其中互信息值大于 R 值的公式,此步骤保证特征集中可疑度公式彼此之间相关性最低。

(6) 剔除后公式数量不足 m ,基于式(4)使用增量搜索的方法在剩余公式中继续挑选,直到特征集中的可疑度公式数量满足 m 值。其对应数学公式为:

$$\max_{x_j \in X - S_{m-1}} \left(I(x_j; c) - \frac{1}{m-1} \sum_{x_i \in X - S_{m-1}} I(x_i, x_j) \right) \quad (4)$$

其中, m 值为特征集数量预设值。

2.3 错误定位模型训练

LTRmR 是一个从选取的可疑度公式中学习错误定位模型的完整框架。在训练阶段,使用一组程序频谱和语句标签作为输入数据,由特征集构造特征值,输入机器学习算法学习错误定位模型。

本文使用随机森林和支持向量机算法训练模型,算法基于 0.21.3 版本 scikit-learn 机器学习工具包^[13]实现,并使用交叉验证方法评估错误定位模型性能。对于 Defects4J 和 SIR 数据集不同的训练数据,模型分别使用留一验证法和 k 折交叉验证法完成训练任务。其中, k 值设置为 10。错误定位模型构造完成后,其功能相当于可疑度公式,读取待测程序频谱数据,经过计算可疑度值将程序语句降序排列,生成可疑语句排序列表,从上到下依次排查错误。

3 实验

3.1 实验研究问题

本文实验基于以下研究问题进行设计。

(1) RQ1: LTRmR 能否提高 learning-to-rank 错误定位精度?

此研究问题是评估本文创新工作的重要参考,通过 mRMR 算法构造一组与语句标签相关性最大,且彼此低相关的特征公式设置对比实验,验证 LTRmR 方法有效性。

(2) RQ2: LTRmR 和传统 SBFL 公式应用 learning-to-rank 技术相比表现如何?

此研究问题进一步评估 LTRmR 的有效性。考虑 2 种不同的特征集:

- ①由传统 SBFL 公式所组成的特征集。
- ②由 GP-evolved 公式所组成的特征集。

另一方面,(1)中公式未做任何处理,仅简单组合而成。(2)中公式应用 mRMR 算法选取,将公式间相关性作为组合考虑因素。

(3) RQ3: LTRmR 和传统 SBFL 公式相比定位效果如何?

此研究问题评估了 LTRmR 与传统 SBFL 公式的错误定位表现,同时评估 learning-to-rank 技术对传统错误定位方法效率提升所提供的帮助。

3.2 实验对象

Defects4J^[14]是一个基于 Java 的大型开源故障数据集,其故障均来自于现实中开源代码中的实际故障。每个故障都有一个测试套件,并保证该测试套件中至少有一个测试用例可以触发该故障。实验使用其中包括 5 个 Java 程序的 357 个故障,具体信息见表 3。

SIR 数据集是软件错误定位领域中较为流行的开源数据集存储库之一,实验使用 4 个来自 SIR 数据集的程序,共 92 个人工故障,见表 4。

表 3 Defects4J 数据集故障信息

程序	故障数量/个	代码量/k 行
Apache CommonsMath	106	103.9
Apache CommonsLang	65	49.9
Joda-Time	27	105.2
JFreeChart	26	132.2
GoogleClosure compiler	133	216.2

表 4 SIR 数据集故障信息

程序	故障数量/个	测试用例数量/个
flex	47	567
grep	11	199
gzip	18	214
sed	16	360

3.3 评价指标

(1) Accuracy 标准。 $acc@n$ 度量标准计算在语句可疑度排名列表中前 n 个位置内成功定位的错误数量。本文选择 1、3、5、10 作为 $acc@n$ 中的 n 值。

(2) EXAM 指标。EXAM 指标是被广泛使用的用来评估错误定位效率的指标^[15]。其值为在可疑语句排序列表中找到第一个错误之前所检查的语句数占程序语句总数的百分比。显然,该指标的值越小表示定位效果越好。

3.4 实验设置

本文所有实验均在 Intel Core i7-4700MQ CPU, 2.40 GHz, 内存为 4 GB, Microsoft Windows 10 操作系统的计算机上完成,其中基因编程算法基于 pyevolve 框架实现。为验证 LTRmR 有效性,设置 2 组对比实验。第一组对比实验使用 15 个传统 SBFL 公式作为特征集训练错误定位模型,具体公式参见表 5。第二组对比实验使用 GP-evolved 公式构成特征集,其可疑度公式选取策略为随机选择,为避免随机性影响实验结果的准确性,在第二组实验中共设置 30 个可疑度公式特征集,每个特征集从 3 000 个备选可疑度公式中随机选取 15 个公式。在评估实验结果时,将 30 组结果的平均值作为第二组实验最终结果。

实验将评估 LTRmR 与 2 组对比实验的错误定位效率,并将实验结果进行对比。图表中分别使用“max”和“random”代表第一、二组实验,另分别使用“r”和“s”代表随机森林和支持向量机两种建模算法。例如,“LTRmR_s”指使用支持向量机算法建模的 LTRmR。

表5 SBFL可疑度公式

Tab. 5 SBFL techniques

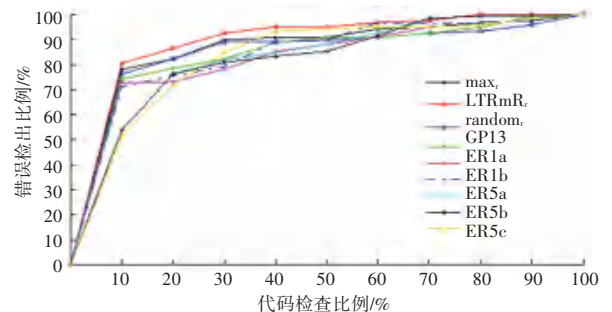
公式	公式表达式	公式	公式表达式
Naish1 (ER1a)	$\begin{cases} -1, & \text{if } a_{ef} < F \\ P - a_{ep}, & \text{if } a_{ef} = F \end{cases}$	GP19	$a_{ef} \sqrt{ a_{ep} - a_{ef} + a_{nf} - a_{np} }$
Naish2 (ER1b)	$a_{ef} - \frac{a_{ef}}{a_{ef} + a_{nf} + 1}$	GP13	$a_{ef} \left(1 + \frac{1}{2a_{ep} + a_{ef}}\right)$
Wong1 (ER5a)	a_{ef}	Kuleczynski	$\frac{a_{ef}}{a_{nf} + a_{np}}$
Russel & Rao (ER5b)	$\frac{a_{ef}}{a_{ef} + a_{nf} + a_{ep} + a_{np}}$	Kulczynski2	$\frac{1}{2} \left(\frac{a_{ef}}{a_{ef} + a_{nf}} + \frac{a_{ef}}{a_{ef} + a_{ep}} \right)$
Binary (ER5c)	$\begin{cases} 0, & \text{if } a_{ef} < F \\ 1, & \text{if } a_{ef} = F \end{cases}$	Jaccard	$\frac{a_{ef}}{a_{ef} + a_{nf} + a_{ep}}$
GP02	$2(a_{ef} + \sqrt{a_{np}}) + \sqrt{a_{ep}}$	Ample	$\left \frac{a_{ef}}{a_{ef} + a_{nf}} - \frac{a_{ep}}{a_{ep} + a_{np}} \right $
GP03	$\sqrt{ a_{ef}^2 - \sqrt{a_{ep}} }$	Ochiai	$\frac{a_{ef}}{\sqrt{(a_{ef} + a_{nf}) \cdot (a_{ef} + a_{ep})}}$
Tarantula	$\frac{a_{ef}}{a_{ef} + a_{nf}}$ $\frac{a_{ep}}{a_{ep} + a_{np}} + \frac{a_{ef}}{a_{ef} + a_{nf}}$		

4 实验结果分析

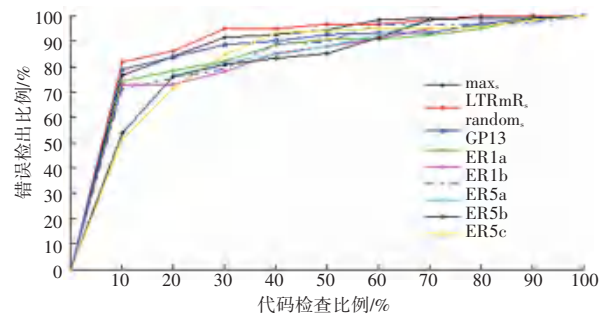
4.1 RQ1

图3和图4显示SBFL公式与LTRmR在5个Defects4J多错误程序和4个SIR多错误程序的错误定位效果。 x 轴表示已检查代码比例, y 轴表示错误的检出比例。可以发现,LTRmR曲线总体位于random曲线上方,证明LTRmR对learning-to-rank错误定位的效率有明显提高。研究时通过随机选取可疑度公式集成学习错误定位模型代表目前大多数learning-to-rank错误定位方法,通过随机集成特征所构造的特征集并未考虑特征间相关性。容易发现,经过LTRmR对特征集进行去冗余处理后,其效率明显增高。 $acc@n$ 评价标准实验结果见表6。从表6实验数据得知,LTRmR对于random在top-10位置所检出的错误数量平均提高20.3%和7.6%,分别基于Defects5J和SIR数据集。由此可见,应用mRMR算法处理错误定位模型的特征集,对于提高错误定位模型性能有显著效果。

为了进一步说明问题,计算了在可疑语句排名列表中第一个错误在被测程序的所有版本中的平均排名。例如,在Defects4J数据集的Closure程序中,



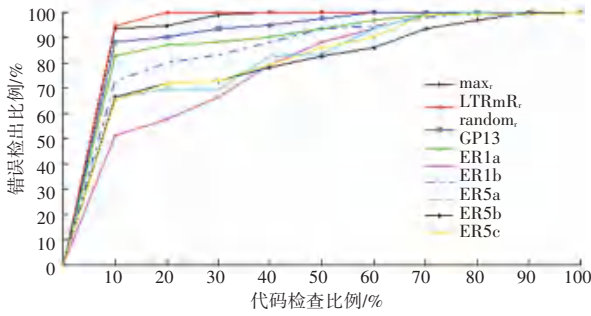
(a) 基于随机森林算法建模



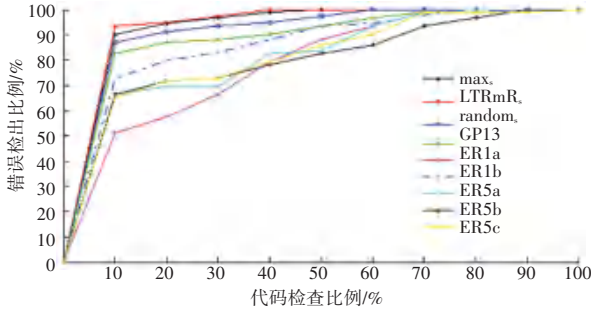
(b) 基于支持向量机算法建模

图3 基于Defects4J数据集LTRmR与其它SBFL技术定位效率比较

Fig. 3 The localization efficiency between LTRmR and other SBFL techniques on Defects4J



(a) 基于随机森林算法建模



(b) 基于支持向量机算法建模

图 4 基于 SIR 数据集 LTRmR 与其它 SBFL 技术定位效率比较

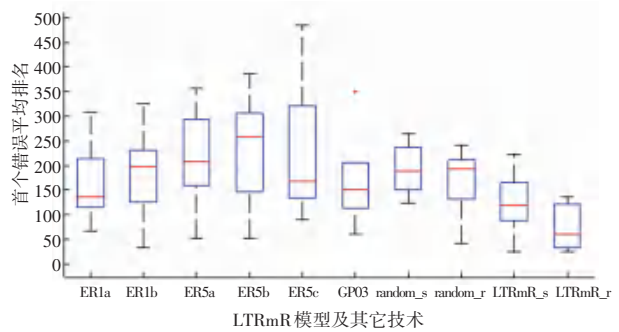
Fig. 4 The localization efficiency between LTRmR and other SBFL techniques on SIR

共有 133 个版本。首先统计 GP13 公式在每个版本中所找到的第一个错误的排名,再计算所有版本排名的平均值,通过计算,GP13 在所有版本中定位首个错误的平均排名是 357.99。实验中统计了来自 Defects4J 的 5 个程序和 SIR 的 4 个程序的第一个错误由不同错误定位技术所得到的平均排名,然后将所得结果与 LTRmR 比较,建立箱形图,如图 5 所示。每个箱型图上方和下方的横线分别表示第一个错误在所有版本中的最高排名和最低排名,箱子的顶部和底部分别表示上 4 分位数和下 4 分位数,箱子内的横线表示中位数,红点表示异常值。从图 5 可以观察得到,LTRmR 的稳定性优于其他错误定位技术,且中位数最低,这表明,在跨项目的多错误版本中 LTRmR 总能更快地定位到错误。

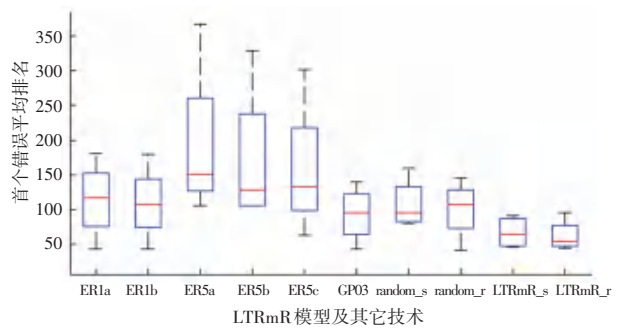
4.2 RQ2

图 3 和图 4 中最上方 3 条线分别表示 LTRmR 与 2 组对比实验的 EXAM 指标表现。从这些图中可以发现,LTRmR 曲线处于所有曲线之上,表明在相同代码检查比例下,LTRmR 能够找到更多的错误。同样地,检出相同比例的程序错误,LTRmR 只需要检查更少的代码。研究通过 3 种不同的组合策略验证了 LTRmR 的有效性,“max”中公式均为领域中应用广泛的 SBFL 公式,相比于特征成员更平民

化的 LTRmR,效率明显略低。相比技术本身,决定错误定位模型定位效率很重要的因素便是特征公式之间的相关性,在保持与目标输出高度相关性的前提下,可疑度公式之间相关性越低,越能为错误定位提供不同方面的信息,起到增益效果。单纯地将公式组合在一起虽然能够提高定位效率,但是只能验证 learning to rank 技术的有效性,并没有考虑不同技术之间的内在关系。从图 4 能够发现,在特征集不变的情况下,使用随机森林算法所构建的错误定位模型往往能够更快地定位到错误,图 4 (a) 中“LTRmR_r”找出全部错误所需检查的代码比例大约为 20%,而图 4 (b) 中“LTRmR_s”找到全部错误则需要检查大约 40%的代码。另一方面,使用单个 SBFL 公式进行错误定位,最快能找到全部错误的公式是 GP13,GP13 在代码检查率达到 10%时能够成功定位超过 80%的错误。



(a) 基于 Defects4J 数据集



(b) 基于 SIR 数据集

图 5 LTRmR 与其它 SBFL 技术定位首个错误平均排名比较

Fig. 5 The average ranking of first detected fault between LTRmR and other SBFL techniques

表 6 详细阐述了 LTRmR 和其他 SBFL 方法在 $acc@n$ 指标上的表现。从表 6 可以发现,在 top-10 前,“LTRmR_r”和“LTRmR_s”分别成功定位 Defects4J 数据集中 257 个和 263 个错误以及 SIR 数据集中的 46 个和 45 个错误。容易发现,在表 5 所示公式中,只有 Kulczynski、Kulczynski2、Ochiai 以及

GP13 能在 top-10 位置前定位的错误比例高于 50%,并分别定位 189、192、183 和 183 个错误。结

果表明,LTRmR 比传统方法更具有性能优势,效率更高。

表 6 $acc@n$ 评价标准实验结果

Tab. 6 The $acc@n$ of experimental results on Defects4J/SIR

	acc@ 1	acc@ 3	acc@ 5	acc@ 10		acc@ 1	acc@ 3	acc@ 5	acc@ 10
Naish1 (ER1a)	1/12	3/19	7/26	13/35	Tarantula	60/5	118/5	135/8	177/9
Naish2 (ER1b)	57/11	105/19	126/26	153/35	Kuleszynski1	61/1	116/5	135/9	189/15
Wong1 (ER5a)	38/7	73/15	97/22	131/34	Kuleszynski2	64/11	120/19	145/27	192/35
Russel&Rao (ER5b)	38/7	73/14	97/25	129/33	Ample	64/1	120/6	140/10	191/15
Binary (ER5c)	60/8	115/17	138/23	159/37	max_r	110/16	159/23	190/34	221/42
GP02	23/3	46/6	66/11	112/19	max_s	131/16	161/19	207/32	230/40
GP03	11/8	24/16	32/22	41/30	random_r	103/15	113/21	139/31	176/40
GP13	61/14	117/24	141/28	183/36	random_s	109/16	123/28	156/30	194/39
GP19	62/11	118/20	137/26	179/35	mRMR_r	135/24	182/30	219/38	257/46
Jaccard	61/11	115/18	138/26	175/35	mRMR_s	135/17	165/25	218/33	263/45
Ochiai	61/11	116/18	138/26	183/35					

注:表中结果为基于 Defects4J 数据集实验结果/基于 SIR 数据集实验结果

4.3 RQ3

根据图 2 和图 3 的结果可以观察得出,LTRmR 曲线,即 LTRmR_r 和 LTRmR_s 总体上位于其他曲线上方,表明 LTRmR 在相同的代码检查量下,可以定位到更多错误。即使在代码检查比例大于 60%之后,会偶尔出现 LTRmR 曲线低于其他公式曲线,但仍然可以肯定 LTRmR 的有效性。一个很重要的原因是,在真实开发环境中,只有在代码检查百分比很低的时候定位到错误,错误定位工作才是有意义的。

由表 6 可知,GP13 和 Naish2 在前 10 检查位置中分别成功定位 Defects4J 共计 357 个错误中的 183 和 153 个错误。对于 SIR 基准数据集的错误,GP13 和 Naish2 分别可以定位 36 和 35 个错误。通过比较以上数据可以得到,GP13 比其他 SBFL 公式定位效率更高。LTRmR 在 top-10 前最多能够定位 263 个故障,较 GP13 相比,LTRmR 的错误检出率提高了 43.7%。通过表中数据发现,LTRmR 比单独使用 SBFL 公式错误定位的准确性高,错误定位效率也有显著提高。

5 结束语

本文提出一种结合基因编程和 mRMR 算法的频谱特征生成和选择方法 LTRmR,研究了在 357 个实际错误和 92 个人工错误上的定位性能,并通过实验对比评估了 15 个 SBFL 公式的错误定位效率。LTRmR 首先通过基因编程生成可疑度公式,然后使用 mRMR 算法选取公式并生成算法模型特征集,最终构建排序模型进行错误定位。经实验结果验证,在

集成多种 SBFL 技术进行错误定位时,各特征成员之间的相关性高低对最终的定位效果有显著影响。从实验结果发现,LTRmR 不仅能够提高 learning-to-rank 错误定位方法的效率,其性能也优于传统 SBFL 公式,错误定位能力和效率有一定提升。

在未来的工作中,研究将对 LTRmR 进行有效改进,使其满足不同颗粒度级别错误定位技术需求。

参考文献

- [1] LIN Rui, MA Liang, ZHANG Wei. An interview study exploring Tesla drivers' behavioural adaptation [J]. Applied ergonomics, 2018, 72: 37-47.
- [2] ZHANG Zhuo, LEI Yan, TAN Qingping, et al. Deep learning-based fault localization with contextual information [J]. IEICE Transactions on Information and Systems, 2017, 100(12): 3027-3031.
- [3] ZHANG Mengshi, LI Xia, ZHANG Lingming, et al. Boosting spectrum-based fault localization using pagerank [C]// Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis. California, United States: ACM, 2017: 261-272.
- [4] 舒挺, 黄明献, 丁佐华, 等. 基于条件概率模型的缺陷定位方法 [J]. 软件学报, 2018, 29(6): 1756-1769.
- [5] 姚弋, 谢晓鸣, 刘书涵, 等. 考虑缺陷关联模型的软件优化测试策略 [J]. 浙江理工大学学报(自然科学版), 2016, 35(1): 78-83.
- [6] LE T D B, LO D, Le GOUES C, et al. A learning-to-rank based fault localization approach using likely invariants [C]// Proceedings of the 25th International Symposium on Software Testing and Analysis. Saarbrücken, Germany: ACM, 2016: 177-188.

(下转第 79 页)