

基于主成分分析法的人工免疫识别软件缺陷预测模型研究

朱朝阳¹ 陈相舟¹ 闫 龙² 张信明²

(中国电力科学研究院信息通信研究所 北京 100192)¹

(中国科学技术大学计算机科学与技术学院 合肥 230027)²

摘要 针对软件系统日益复杂以及软件缺陷难以预测的问题,提出了一种使用人工免疫识别系统的软件缺陷预测模型。模型的构建首先通过主成分分析法对软件缺陷预测数据集进行特征的识别和提取,进一步提高学习算法的性能;针对计算亲和度时欧氏距离不能满足非线性应用需求的情况,为了提高算法对非线性应用的适用性,使用基于高斯径向基核函数的亲和度计算来计算抗体和抗原之间的亲和度阈值。再基于亲和度计算,进行抗体训练、资源竞争以及记忆细胞的选择。最后,利用记忆细胞集进行分类。模拟实验表明,针对实验中的数据集,所提模型的预测准确度达到84%~90%,精确度达到85%~91%。

关键词 软件缺陷检测,人工免疫识别系统,主成分分析法,静态分析,有监督机器学习

中图分类号 TP311.53 **文献标识码** A

Research on Software Defect Prediction Based on AIRS Using PCA

ZHU Chao-yang¹ CHEN Xiang-zhou¹ YAN Long² ZHANG Xin-ming²

(Information & Communication Department, China Electrical Power Research Institute, Beijing 100192, China)¹

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)²

Abstract Aiming at the problem that the software system is becoming more and more complex and the software defect is difficult to detect, a software defect prediction model based on artificial immune recognition system was proposed. The model was constructed firstly by using the principal components analysis method to reduce the dimension of the original data set, and then the affinity between antibody and antigen was calculated based on the Gauss radial basis function (RBF). Based on affinity calculation, antibody training, resource competition and the selection of memory cells were conducted. Classification was performed using memory cell set at last. Simulation shows that our model prediction accuracy reaches 84%~90%, accuracy reaches 85%~91%.

Keywords Software defect detection, Artificial immune recognition system, Principal components analysis, Static analysis, Supervised machine learning

1 引言

在信息技术快速发展的今天,软件开发日益复杂,针对软件可靠性的预测对软件开发有着越来越重要的作用。特别是对于一些大型软件,由于其规模较大,更是难以管理和监控。软件是一种虚拟产品,不会像实体商品一样出现物理上的损耗,但可能出现无法预测的故障。软件缺陷分析和预测主要有3个目的^[1]:首先,为项目经理评估项目进度和计划缺陷检测活动提供依据;其次,评估产品质量,最终提高过程管理能力和评估过程绩效;最后,缺陷检测数据能够帮助确定故障易发模块、缺陷位置、缺陷数以及缺陷分布的信息可以提高测试效率和软件质量。软件缺陷预测对于提升软件质量及降低软件开发的成本有着十分重要的意义^[2]。

进行软件缺陷预测的方法有很多种,比较常用的是进行静态分析。在软件质量监控活动中,越早发现软件缺陷对于软件缺陷的修正越有意义,因为越早发现软件缺陷就能以更

小的代价来修正软件缺陷。没有在现阶段检测出的缺陷会被遗漏到下一阶段,缺陷修复的成本会成倍增加,甚至最终难以修复^[3]。

静态分析技术进行软件缺陷预测时,并不需要运行程序,在软件实现阶段就能较早地检测到软件缺陷,对软件质量管理能够起到十分重要的作用,因而被很多学者、大型软件公司所关注。软件缺陷预测早期是基于字符串匹配进行的,逐渐发展为基于词法分析和语义分析,后来逐渐使用统计学方法构建软件预测模型,但是统计学方法的准确度不能满足要求^[4]。直至近些年将数据挖掘应用到软件缺陷预测,静态分析技术的手段在逐渐增多,静态分析技术也在不停地进步^[5]。

在数据挖掘技术中,已经有很多构建软件缺陷预测模型的方法,比如遗传规划、决策树^[6]、神经网络^[7]、贝叶斯方法^[8]等。但这些方法应用于软件缺陷预测时还存在一些问题。决策树存在过度拟合的问题;神经网络有时会产生局部最优,并且有时拟合程度不够,此时就需要根据专家经验获得与软件

本文受国家自然科学基金(61672485, 61379130)资助。

朱朝阳(1974—),男,博士,高级工程师,主要研究方向为信息系统安全;陈相舟(1986—),男,硕士,工程师,主要研究方向为信息通信仿真与测试;闫 龙(1992—),男,博士生,主要研究方向为无线网络、车载自组织网络;张信明(1964—),男,博士,教授,博士生导师,主要研究方向为无线网络、智能电网、基于深度学习的大数据分析, E-mail: xinming@ustc.edu.cn(通信作者)。

缺陷相关的因子,计算效率低。

人工免疫识别系统是第一个专门为分类设计的人工免疫系统算法,它是一种免疫后发式监督学习算法^[9]。人工免疫识别系统是一个强大的分类器,本文使用基于人工免疫识别系统的有监督机器学习方法来构建软件预测模型。首先使用主成分分析法对数据集进行特征的识别和提取,进一步提高学习算法的性能;针对计算亲和度时欧氏距离不能满足非线性应用需求的情况,为了提高算法对非线性应用的适用性,使用核函数来提高算法的非线性能力,完成基于人工免疫识别系统的软件预测模型的构建。

2 算法数据集预处理

在进行软件缺陷预测研究时,如何获取数据集是需要解决的首要问题。如果研究只是基于特定公司提供的数据,那么所得到的结果明显缺少足够的适用性和公信力。在 2005 年以后,公共数据集的使用显著增加^[10]。本文使用 PROMISE 项目提供的公共数据集。实验数据来源于美国国家航空航天局(NASA)发布的一个数据包,数据内容主要是对软件源代码 McCabe 和 Halstead 复杂度的解析,这些数据是对与软件质量相关的代码特征进行的一种客观描述。

在现实世界中,对要解决的问题进行研究时,很多影响因子都需要被考虑到。这些影响因子可以称作变量。每个影响因子都体现了所分析问题的部分信息,而且影响因子间有相当程度的关联性,因而影响因子间体现的信息有很大程度的重叠。在进行多变量问题分析时,由于影响因子太多,导致问题剖析的复杂性大大增加,计算量也会太大,导致所研究的问题很难得到高效的解决,这时就需要利用主成分分析法来进行特征识别和提取^[11]。主成分分析法算法如下^[11-12]:

1) 对 m 维的变量 $X = (X_1, X_2, \dots, X_m)^T$, 计算其协方差矩阵 $\Sigma = XX^T$ 。

2) 计算 Σ 的特征值 λ_i ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$), 再计算特征值相应的特征向量,并基于特征向量计算 X 的主成分。

3) 根据下面的公式得到主成分贡献率以及累计贡献率,最后基于累计贡献率得到主成分向量。

$$\text{贡献率: } \frac{\lambda_k}{\sum_{i=1}^m \lambda_i}$$

$$\text{累计贡献率: } \frac{\sum_{i=1}^p \lambda_i}{\sum_{i=1}^m \lambda_i}$$

特征的识别和提取对数据集的学习很有帮助,这些功能体现在对数据集价值的分析以及未来的预测上。通过去除不重要的和冗余的数据,学习算法的性能可以得到很大的提高。如果数据未能显示出统计规律性,那么学习算法就不能得到很好的结果,学习就会失败,所以从训练集里进行特征的识别和提取显得尤为重要,它会让学习算法速度更快,而且更准确^[9]。

3 算法设计

人工免疫系统具有分布式、鲁棒性和动态稳定性等特点^[13],这些特性是人工免疫系统被广泛应用的基础,也促使研究人员从免疫系统中获取灵感而开发相应的人工免疫机制和模型。对应于免疫系统的应答过程,本文的软件缺陷预测

模型主要基于下面 3 种重要的仿生机理:免疫学习、免疫记忆和克隆选择^[14]。

(1) 免疫学习:免疫系统对抗原的应答分为两种情况。当初次应答时,这种情况下由于抗体是第一次接触抗原,一般应答时间相对较长,这也是一个进行免疫学习的过程;另一种是发生再次应答时,由于在初次应答时进行了免疫学习,免疫系统对抗原的处理速度更快,而且产生的抗体与抗原的亲和度更高^[14]。这个过程对应于软件缺陷预测中训练阶段抗体学习软件缺陷模块的规律。

(2) 免疫记忆:免疫记忆是免疫学习的结果。当初次应答时,免疫细胞需要时间进行免疫学习,并在免疫学习终止的时候保留抗原的信息。这样,当免疫系统再次发生应答时,因为初次应答后已有免疫记忆存在,所以其对曾遇到的同类抗原的处理速度将会更快。免疫学习和免疫记忆也就是软件缺陷预测中训练阶段抗体学习软件缺陷模块规律的过程。

(3) 免疫克隆:当免疫细胞成功识别抗原后,B 细胞被激活并进行克隆,只有成功对抗原进行识别的 B 细胞才能够进行克隆。B 细胞克隆会分化为两种,一种是抗体,一种是记忆细胞。抗体主要负责对检测到的抗原进行处理;记忆细胞主要是进行免疫记忆,以便在发生再次应答时提高应答速度。

在早期的研究中,人工免疫系统主要被用于优化等问题。人工免疫识别系统是第一个专门为分类设计的人工免疫系统算法,它是一种免疫后发式监督学习算法。人工免疫识别系统是一个强大的分类器,适用于软件缺陷预测过程中从数据集中识别有缺陷的模块。

在人工免疫识别系统中,抗体分子和抗原分子通常使用相同长度的向量来表示。抗体分子和抗原分子之间的亲和度常用欧氏距离来衡量。如果使用 ag 表示抗原,用 ab 表示抗体,则使用欧氏距离来衡量的抗体和抗原之间的亲和度可表示为:

$$\text{affinity} = 1 - \sqrt{\sum_{i=1}^n (ab_i - ag_i)^2} \quad (1)$$

由于运算简单并且易于理解,欧氏距离已经成为人工免疫识别系统中最常用的计算亲和度的表示方法。但欧氏距离有时并不能取得最优的性能,而且由于欧氏距离是适用于线性空间的,这就使得欧氏距离不能满足非线性应用的需求。为了提高算法对非线性应用的适用性,本文使用核函数来提高算法的非线性能力。在软件缺陷预测中,需要考虑的参数很多,数据集通常非常复杂。本文针对这种情况,首先在初始化时用主成分分析法对数据集进行特征分析和提取,再采用高斯径向基核函数来计算抗体分子和抗原分子之间的亲和度,以提高算法的非线性能力。高斯径向基核函数的计算公式可表示为:

$$K(Ab, Ag) = \exp\left(-\frac{\|Ab - Ag\|^2}{2\sigma^2}\right), \sigma > 0 \quad (2)$$

其中, $Ab = [ab_1, ab_2, ab_3, \dots]$, $Ag = [ag_1, ag_2, ag_3, \dots]$ 。基于高斯径向基核函数的核空间距离可表示为:

$$\text{dist} = (K(Ab, Ab) + K(Ag, Ag) - 2K(Ab, Ag)) * \delta \quad (3)$$

其中,参数 δ 用来控制 $\text{dist} \leq 1$ 。基于高斯径向基核函数的亲和度可表示为:

$$\text{affinity} = 1 - \text{dist}(Ab, Ag) \quad (4)$$

使用高斯径向基核函数的人工免疫识别系统可以分为 5 个步骤:初始化、抗体训练、资源竞争、记忆细胞选择和分类,

具体流程如图 1 所示。步骤如下:

(1)初始化。这一步骤主要分为两部分:对软件缺陷预测数据集进行主成分分析和亲和度阈值计算。首先是使用主成分分析法对原始数据集进行特征的识别和提取,然后进行基于高斯径向基核函数的亲和度计算,根据亲和度计算的结果设置亲和度阈值。亲和度阈值的计算公式为:

$$k = \frac{\sum_{i=1}^n \sum_{j=i+1}^n affinity(ag_i, ag_j)}{n(n+1)/2} \quad (5)$$

(2)抗体训练。首先需要从训练样本中随机选择数据创建记忆细胞集,然后再从剩余的训练样本中随机选取一个抗原 ag 。对于记忆细胞集中的每个记忆细胞,计算其与抗原 ag 的刺激度,刺激度高的被选作最优记忆细胞。

(3)资源竞争。对记忆细胞进行克隆,克隆时按指定概率变异,克隆体作为 B 细胞放入抗体集中。对于抗体集中的每个抗体,根据其与其抗原的刺激度分配资源,刺激度高的抗体分配的资源更多。由于资源是有限的,分不到资源的抗体会慢慢消亡。

(4)记忆细胞选择。若抗体集中抗体与抗原 ag 的平均刺激度满足要求,则选取与 ag 刺激度最高的抗体加入记忆细胞集,转到步骤(2),选择未训练的抗原进行训练,直到所有抗原训练完毕;否则,转到步骤(2)继续训练当前抗原。

(5)利用记忆细胞集进行分类。当训练过程完成后,把记忆细胞集作为分类器的核心,得到记忆细胞集所对应的原始数据集中的向量。分类采用 K 最近邻算法,由离测试样本最近的 K 个邻居投票决定样本的分类。

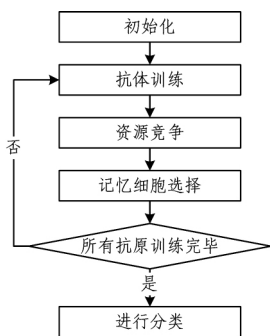


图 1 基于人工免疫识别系统的软件缺陷预测模型构建流程

4 实验

4.1 实验设置

为了验证本文所提模型的性能,基于 MATLAB 实现了本文所提出的算法 PCA-AIRS,并选择 ACO-SVM^[12]和传统的未使用主成分分析法进行数据预处理的人工免疫识别系统(AIRS)作为对比对象。实验数据来源于美国国家航空航天局(NASA)发布的一个数据包,数据内容主要是对软件源代码 McCabe 和 Halstead 复杂度的解析。Shepperd 等人^[15]发现之前使用的 NASA 数据集中含有噪音,因此对数据集进行了清理。使用清理后的数据集进行实验。由于缺陷在被测项目的分布大致满足二八原则,因此造成缺陷模块的数量要远小于无缺陷模块的数量。NASA 数据集中很多数据都存在类不平衡问题,这个显然不满足实际的场景。从清理后的 NASA 数据集中选取 4 个大致满足二八原则的数据集进行实验,如表 1 所列。

表 1 实验数据集

数据集	模块数	维数	无缺陷模块数	有缺陷模块数
JM1	7782	22	6110	1672
KC3	194	40	158	36
PC3	1077	38	943	134
PC5	1287	38	1110	177

本文采用准确率、精准度这两个指标来衡量模型的检测能力。这两个指标的定义可以从表 2 中推出。

表 2 检测指标定义

	模块无缺陷	模块有缺陷
预测无缺陷	A	B
预测有缺陷	C	D

准确率表示准确分类的模块数与总模块数的比值,可表示为:

$$acc = \frac{A+D}{A+B+C+D} \quad (6)$$

精准度表示预测有缺陷的模块中确实有缺陷的模块所占的比重,可表示为:

$$prec = \frac{D}{C+D} \quad (7)$$

4.2 实验结果

本文采用十折交叉验证法进行实验^[12],将选择的每个数据集分成 10 份,将其中 9 份作为训练数据,1 份作为测试数据进行实验。在轮流进行 10 次实验后,统计 10 次实验的结果,再计算 10 次结果的平均值作为对模型预测能力的评价。

实验结果如图 2、图 3 所示。

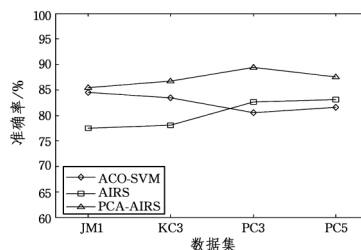


图 2 各个数据集的预测准确度

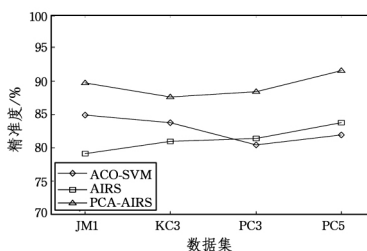


图 3 各个数据集的预测精准度

从图 2、图 3 可以看出,本文提出的协议在 4 个数据集的预测准确度、预测精准度上都优于 ACO-SVM 和 AIRS。这一方面是因为本文使用主成分分析法对数据集进行特征的识别和提取,进一步提高了学习算法的性能;另一方面是因为针对计算亲和度时欧氏距离不能满足非线性应用需求的情况,为了提高算法对非线性应用的适用性,使用核函数来提高算法的非线性能力。相比于 ACO-SVM 和 AIRS,对于这 4 个数据集,本文所提算法在预测准确度上平均提高了 5.8% 和 8.7%,在预测精准度上平均提高了 7.9% 和 9.9%。

结束语 本文针对软件缺陷难以检测的问题,提出了一

(下转第 518 页)

进入 APP 软件后能够运行查看菜品餐桌、点餐和点赞等功能,运营结果完全符合各项指标要求。

5.4 软件综测上线最终验收

软件综测上线测试的框图如图 5 所示,包含:1)关卡任务,通过 ECLIPSE 打包导出 APK;2)代码重构及 BUG 修复,代码重构时,给类文件上的文件进行注释^[13],方便后期的维护和二次开发;进行魔法数值(代码中直接出现的数值)的使用,通过该数值了解其含义;在代码的字符串内引用 @ string 中的资源及 color.xml 资源,采用重要操作抽出方法^[14-16];3)生成 APK 签名文件,对发布的 APK 文件进行唯一签名,保证每次发布的版本的一次性;4)Eclipse 打包导出 APK 文件并发布。

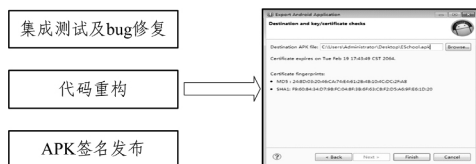


图 5 项目综测框图

结束语 物联网终端设备软件开发系统为信息化旅游及其他行业提供了快速、有效的 APP 软件开发模式。整个系统由客户端、服务器端、仿真应用环境、硬件及扩展能力构成。仿真结果表明本系统能够完成 APP 软件界面设计、程序开发和仿真测试,以云+端的虚拟桌面系统实现实验环境与真实旅游环境发展的同步,通过完成主页面的布局,使登录联网请求和登录数据化持久,使项目测试和 DEBUG 及代码重构,从而达到项目综测和上线的目的。本系统的应用为旅游信息化发展提供了一个有力的保障。

参 考 文 献

[1] 唐川,姜禾,张娟,等. 物联网关键技术发展态势分析[J]. 科学观察,2012,7(1):6-21.

[2] 程子阳. 移动互联网业务的发展趋势[J]. 移动通信,2012,(5):68-69.

[3] 高云娇,王晨. APP 应用对旅游行业的影响分析[J]. 产业与科技论坛,2014(12):115-116.

[4] FLING B. 移动应用的设计与开发[M]. 电子工业出版社,2010:201-202.

[5] BRICKELL E, CHEN L Q, LI J T. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings[J]. International Journal of Information Security, 2009, 8(5):315-330.

[6] 杨明洁,黄晓靖. 设计趋势报告[M]. 北京:北京理工大学出版社,2012:121-123.

[7] ZHANG H G, WANG F. A behavior-based remote trust attestation model[J]. Wuhan University Journal of Natural Sciences, 2006, 11(6):1819-1822.

[8] TENNETI R, JOHNSON D, GOLDENBERG L, et al. Towards a capabilities database to inform inclusive design: Experimental investigation of effective survey-based predictors of human-product interaction[J]. Applied Ergonomics, 2011, 43(443):713-726.

[9] 赵江洪. 人机工程学[M]. 北京:高等教育出版社,2006:156.

[10] 张世颖. 移动互联网用户生成内容动机分析与质量评价研究[D]. 吉林:吉林大学,2014.

[11] 刘子建,黄晟. 限制机制在用户体验设计中的应用[J]. 包装工程,2012,2(4):52-55.

[12] 朱生,牟星亮,单康康. 基于 Android 平台的应用程序开发研究[J]. 网络安全技术与应用,2013(10):46-47.

[13] 王钰. 基于云计算的软件测试[J]. 现代计算机,2013(23):47-50.

[14] 刘法旺,杨场,骆骏瑞. 智能移动终端应用软件综合测试服务平台[J]. 软件,2011,(12):56-57.

[15] 张娟. 软件测试中测试用例复用的研究[D]. 上海:上海大学,2012.

[16] 刘舒. Android 功能自动化测试工具的设计与实现[D]. 大连:大连理工大学,2013.

(上接第 485 页)

种基于人工免疫识别系统的软件缺陷预测模型。模型的构建使用基于高斯径向基核函数计算亲和度。基于亲和度计算,进行抗体训练、资源竞争以及记忆细胞的选择。最后,利用记忆细胞集进行分类。模拟实验表明,针对实验中的数据,本文的模型准确度、检测率和精准度都有良好的表现。

参 考 文 献

[1] CLARK B, ZUBROW D. How Good is the Software: A Review of Defect Prediction Techniques[C]// Sponsored by the US department of Defense, 2001.

[2] 王斌,吴太文,胡培培. 软件缺陷分类和分析研究[J]. 计算机科学,2013,40(9):16-20.

[3] 张华. 面向软件缺陷检测的静态分析技术[J]. 潍坊学院学报,2008,8(2):8-11.

[4] 唐磊,李春平,杨柳. 统计策略序列模式挖掘及其在软件缺陷预测中的应用[J]. 计算机科学,2013,40(5):164-167.

[5] 周丹丹,李先国. 基于静态检测工具的软件缺陷检测模型研究[J]. 计算机与现代化,2012(11):55-58.

[6] 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.

[7] 费青春,严沁,史莹莹. 基于 BP 神经网络软件测试缺陷预测技术研究及应用[J]. 测控技术,2016,35(1):102-105.

[8] MENZIES T, GREENWALD J, FRANK A. Data Mining Static Code Attributes to Learn Defect Predictors [J]. IEEE Transactions on Software Engineering, 2007, 33(1):2-13.

[9] ABAEI G, SELAMAT A. A survey on software fault detection based on different prediction approaches [J]. Vietnam Journal of Computer Science, 2013, 1(1):1-17.

[10] CATAL C, DIRI B. A systematic review of software fault prediction studies [J]. Expert Systems with Applications, 2009, 36(4):7346-7354.

[11] SMITH, LINDSAY I. A tutorial on principal components analysis[J]. Information Fusion, 2002, 51(3):219-226.

[12] 姜慧研,宗茂,刘相莹. 基于 ACO-SVM 的软件缺陷预测模型的研究[J]. 计算机学报,2011,34(6):1148-1154.

[13] 郭依林,李中华,毛宗源. 自适应人工免疫算法在数据挖掘中的应用[J]. 计算机应用,2006,26(8):1943-1946.

[14] 肖人彬,王磊. 人工免疫系统:原理、模型、分析及展望[J]. 计算机学报,2002,25(12):1281-1293.

[15] SHEPPERD M, SONG Q, SUN Z, et al. Data Quality: Some Comments on the NASA Software Defect Datasets [J]. IEEE Transactions on Software Engineering, 2013, 39(9):1208-1215.