

基于拟似然估计方法的软件失效预测模型

张晓风 张德平

(南京航空航天大学计算机科学与技术学院 南京 210016)

摘要 软件缺陷预测是软件可靠性研究的一个重要方向。由于影响软件失效的因素有很多,相互之间关联关系复杂,在分析建模中常用联合分布函数来描述,而实际应用中难以确定,直接影响软件失效预测。基于拟似然估计提出一种软件失效预测方法,通过主成分分析筛选影响软件失效的主要影响因素,建立多因素软件失效预测模型,利用这些影响因素的数字特征(均值函数和方差函数)以及采用拟似然估计方法估计出模型参数,进而对软件失效进行预测分析。基于两个真实数据集 Eclipse JDT 和 Eclipse PDE,与经典 Logistic 回归和 Probit 回归预测模型进行实验对比分析,结果表明采用拟似然估计对软件缺陷预测具有可行性,且预测精度均优于这两种经典回归预测模型。

关键词 软件失效预测,主成分分析,Logistic 回归,拟似然估计

中图分类号 TP311 文献标识码 A

Software Failure Prediction Model Based on Quasi-likelihood Method

ZHANG Xiao-feng ZHANG De-ping

(College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)

Abstract Software defect prediction is an important direction of software reliability research. Because there are many factors influencing the software failure, and the relationship among them is complicated, the joint distribution function is commonly used to describe the analysis model, which is difficult to be determined in the practical application. This problem may impact software defect prediction directly. In this paper, we proposed a Quasi-likelihood method (PCA-QLM), which uses PCA to select the main metrics firstly, and then build defect prediction model. In our model, we can use the mean function and variance function of dependent variable to get the estimated parameter and then predict defects. In this paper, we draw a conclusion that PCA-QLM can apply to the software failure prediction and its performance is better than other models by comparing with probit regression forecasting model and logistic regression forecasting model based on two real datasets Eclipse JDT and Eclipse PDE.

Keywords Software failure prediction, PCA, Logistic regression, Quasi-likelihood method

1 引言

随着信息技术的迅速发展,计算机软件的应用越来越广泛。高效、安全的软件系统高度依赖于软件的可靠性,而影响软件可靠性的软件缺陷已成为导致系统错误、失败、崩溃、甚至是灾难的根本原因。因此,对软件缺陷预测的研究成为软件工程领域非常重要的课题。对有缺陷倾向的模块进行准确的预测有助于减少测试工作量、降低成本、提高软件质量。因此人们开始收集和利用与软件缺陷有关的数据,并构造软件缺陷预测模型,用于预测软件的缺陷信息,及时对软件存在的问题进行修复^[1,2]。

目前,大量的软件缺陷预测模型被研究出来。Zhang F 等^[3]针对在不同的项目中不同的环境因素导致缺陷预测模型的分布函数也不相同这一问题,提出上下文感知的等级转换缺陷预测模型。Yeresime Suresh 等^[4]使用 CK 度量元,检验线性回归、Logistic 回归和人工神经网络等模型在软件缺陷

预测中的应用。Basili^[5]、Briand^[6]以及 Olague^[7]将多元逻辑回归运用于软件缺陷预测中。其中 Basili 等人使用面向对象的 CK 度量元作为逻辑回归模型的输入,并基于实验结果分析讨论了该类度量元的优缺点;Briand 等人研究的主要目标是经验探索面向对象的耦合、内聚、继承度量元与故障检测率之间的关系,使用的预测模型为逻辑回归模型,并提出一种可复用的、完整的分析步骤,有助于分析来自不同数据集的实验结果;Olague 等人研究 3 类度量元的缺陷预测能力,分别为 CK 度量元、MOOD 度量元以及 QMOOD 度量元,实验中采用多元逻辑回归模型,并通过实验得出结论:CK 和 QMOOD 度量元在缺陷预测中的效率相近且较高,而 MOOD 效率则相对较低;Kanmani 等^[8]将广义回归和神经网络相结合,提出针对软件故障检测率的广义回归神经网络模型(GRNN)。Nagappan 等^[9]研究基于多元线性回归的软件缺陷预测模型,模型输入为 STREW-J 度量元,通过实验得出对于低质量的程序,基于 STREW 度量元的逻辑回归模型也是

本文受国防重点项目资金(JCKY2016206B001),国防一般项目(JCKY2014206C002)资助。

张晓风(1994—),女,硕士生,主要研究领域为软件测试与软件可靠性建模,E-mail:1522423422@qq.com;张德平(1973—),男,博士,主要研究领域为软件测试与软件可靠性建模,E-mail:depingzhang@nuaa.edu.cn。

可行的这一结论。

在这些模型中,很多对现实都有着严格的约束,以确保模型的可行性。但是,在现实中,这些约束通常很难满足。例如,很多模型是基于事先假定自变量的联合分布函数才可行的,如多元线性回归模型假定自变量的联合分布为线性回归函数,逻辑回归模型则假定自变量的联合分布为 Logistic 函数,要更准确得到自变量的联合分布函数很困难。针对这一问题,本文提出基于拟似然估计的软件故障预测模型(PCA-QLM),首先利用主成分分析技术筛选特征属性集,得出主成分数据集,再利用得到的主成分数据,通过拟似然估计法得出模型中的参数值,进而预测软件失效率。

2 软件失效影响因素分析

2.1 软件失效影响因素

影响软件失效的因素有很多,将这些影响因素量化就形成了度量元。度量元是描述软件产品或者软件开发过程的指标或参数,主要有面向过程度量元和面向对象度量元两大类,其中面向对象系统强调的是对等实体之间的关系而不是传统控制流所决定的层次分解关系,传统的度量有一定的局限性,而且对某些面向对象特性无法度量,因此需要新的度量元来反映。目前常用的面向对象度量元有 CK 度量、MOOD 度量、QMOOD 度量等。本文采用的数据集是基于 D' Ambros^[10]推广的面向对象的 Eclipse 数据集,只是数据采集的时间以及度量元个数与 D' Ambros 数据集有所不同。在 D' Ambros 的数据集中有 17 个度量元,包括 CK 度量元以及面向对象的其他度量元。为了初步降低维度,采用基于聚类的属性过滤过程对上述 17 个属性进行筛选,实现的具体过程是先对所有的属性空间进行聚类,然后选择每个聚类中有代表性的属性,一般是选择距离和最小的属性。筛选出的度量元需覆盖源码的不同属性,如复杂度(WMC)、耦合度(FAN-IN, FAN-OUT)、内聚度(LCOM)、规模(NO, LOC, NOM),筛选结果如表 1 所列。

表 1 数据集中的度量元

度量元	描述	度量类别
WMC	每类加权方法数	复杂度
LCOM	方法内聚缺乏度	内聚度
FAN-IN	扇入数	耦合度
FAN-OUT	扇出数	耦合度
NOA	属性数	规模
LOC	代码行数	规模
NOM	方法数	规模

2.2 主成分分析

主成分分析(Principal Component Analysis, PCA)又称主分量分析,是由 Pearson 于 1901 年首先引入,后来由 Hotelling 于 1933 年进行了发展。PCA 是一种掌握事物主要矛盾的统计分析方法,它通过降维技术把多个变量化为少数几个主成分,实现从多元因素中解析出主要因素,简化问题。这些主成分含有原始变量的大部分信息,通常表示为原始变量的线性组合。为使主成分不含重叠的信息,PCA 要求主成分之间互不相关。在软件缺陷预测中,影响软件失效的因子有很多,这些因子之间存在着明显的相关性关系,即包含的信息有所重叠,使用 PCA 技术可以得到对软件失效贡献最高的几个因子,从而实现降维的目的,简化计算。PCA 的实质是对角

化协方差矩阵,通过对协方差矩阵的对角化得到每个主成分的特征值(即对角化后的协方差矩阵中主对角线上的值),进而得到每个主成分的贡献率。其中特征值的大小代表了矩阵正交化之后所对应特征向量对于整个矩阵的贡献程度,贡献率是指第 i 个主成分 z_i 的方差所占的比例:

$$\lambda_i / \sum_{j=1}^p \lambda_j (i=1, 2, \dots, p) \quad (1)$$

前 $m(m \leq p)$ 个主成分的贡献率之和即为累积贡献率,由于 PCA 的主要功能就是降低维度,因此可以在原始信息损失不多的情况下,用少数的主成分来代替原始的高维数据,而防止信息丢失过多的依据就是累积贡献率,最终选取的前 m 个主成分的累积贡献率一般不低于某一水平(一般取 85%)。

2.3 拟似然估计

拟似然估计的方法最先由 Wedderburn^[11]于 1974 年提出出来,用于解决广义线性模型中因变量只能服从指数分布族这一局限性问题。拟似然估计并不需要确定因变量 Y 的具体函数分布形式,只需正确设定因变量的均值函数和方差函数即可,甚至在方差函数不确定的情况下仍然适用。拟似然估计方法的基本思想为^[12-14]:

对因变量的均值函数和方差函数做出明确的表述:

$$E(Y^i | X^i) = \mu(\beta) = h(X^i, \beta) \quad (2)$$

$$\text{var}(Y^i | X^i) = \sigma^2(\mu) = \nu(\mu) \quad (3)$$

其中,函数 $\mu(\cdot)$ 为均值函数, $h(\cdot)$ 为已知的充分光滑的函数,函数 $\nu(\cdot)$ 为方差函数, X 为 $m \times n$ 矩阵, Y 为一维因变量, β 为 q 维未知参数向量,即所需要估计的参数向量,则可以给出拟似然函数为:

$$Q(\beta) = Q(\beta, Y^{(n)}) = \sum_{i=1}^n \int_{y_i}^{\mu_i(\beta)} \frac{y_i - t}{\nu(t)} dt \quad (4)$$

其中, $Y^{(n)} = (y_1, y_2, \dots, y_n)$, $\mu_i(\beta) = h(x^i, \beta)$ 。从而表示出得分函数为:

$$U(\beta) = \frac{\partial Q(\beta)}{\partial \beta} = \sum_{i=1}^n \frac{\partial h(x^i, \beta)}{\partial \beta} \times \frac{y_i - \mu_i(\beta)}{\nu(\mu_i(\beta))} \quad (5)$$

令 $U(\beta) = 0$, 所得的解即为参数的拟似然估计值。

通常因变量的方差函数不易确定,因此可以采用替代函数替代方差函数。Liang 和 Zeger^[15]提出,只要均值函数假定正确,就可以预先假定响应变量的“工作分布”,从而用“工作方差”替代真实方差。本文采用的则是局部加权法^[16],用核函数进行局部加权。具体的做法是:设 $\varepsilon_i = (y_i - \mu_i(\beta))$, 则方差函数的替代函数为:

$$\nu(\mu) = \sum_{i=1}^n w_{ni}(\mu) \times \varepsilon_i^2 \quad (6)$$

其中, $A_{nj} = \frac{1}{nb_n} \sum_{i=1}^n K(\frac{\mu_i - \mu}{b_n}) (\mu_i - \mu)^j$, $\Delta n(\mu) = A_{n0} \times A_{n2} - A_{n1}^2$ 。

3 基于拟似然估计的软件失效模型

假定 Y^i 为第 i 模块的缺陷数量,记 X^i 为第 i 模块的 d 维度量元。软件失效预测的目的是在各类软件失效影响因素 X^i 给定的条件下,预测第 i 模块的缺陷数量:

$$Y^i = \Psi(X^i) \quad (7)$$

其中, Ψ 为某一累积分布函数,一般选择标准正态分布或 Logistic 分布。根据 Ψ 取值的不同可得到不同的预测分析模型,如 Ψ 取正态分布函数时为 Probit 回归预测模型,若 Ψ 为 Logistic 分布时为 Logit 回归预测模型。

然而,要更精确确定累积分布函数 Ψ 非常困难,这将无法利用最大似然估计方法来确定软件失效预测模型的参数。在实际应用中,尽管累积分布函数 Ψ 无法精确确定,但其数字特征通常容易确定,如均值函数、方差函数。拟似然估计的方法只需分析因变量的均值函数和方差函数,即一阶中心距和二阶中心距的函数,而不需要知道因变量具体服从什么分布,就可在适当的条件下得到参数的相合估计,称为拟似然估计值^[16-20]。

本文提出的基于拟似然估计的软件失效预测模型(PCA-QLM)首先利用主成分分析技术筛选出最优特征属性子集,将筛选的主成分作为拟似然估计的自变量,通过拟合得到因变量的均值函数分布并表示出因变量的方差函数,带入拟似然估计的得分函数中,得出均值函数中参数的拟似然估计值,最终计算出软件失效率。具体的算法流程为:

输入:数据集 $A(x_1, x_2, \dots, x_p, y)$, p 为原始数据维度, y 为因变量,表示软件模块失效的概率。

输出:测试集中每个样本 $(z_{11}, z_{12}, \dots, z_{1q})$ 的预测值 \hat{y}_i 和整体预测效果评价指标值, q 为降维后的维度。

1. 数据预处理:对数据进行归一化处理以及去除异常值;
2. PCA 对数据集进行降维,得到新的数据集 B , 样本为 $(z_1, z_2, \dots, z_q, y)$, q 为降维后的维度;
3. 将数据集 B 分割为训练集和测试集 B_1, B_2 , 其中 B_1 表示训练集, B_2 表示测试集;
4. 利用软件 `1stopt` 寻找训练集 B_1 中因变量 y 的期望值和自变量 z_i 之间最优函数关系(AE 值最小),从而得到因变量 y 的期望函数,即 $E(Y^i | X^i) = \mu(\beta) = h(X^i, \beta)$;
5. 利用得到的均值函数表示出因变量的方差函数;
6. 将因变量的均值函数和方差函数带入到拟似然函数的得分函数中;
7. 令得分函数为 0, 求出方程的根,即为均值函数中参数的拟似然估计值;
8. 将参数的估计值带回期望函数中,并将均值函数运用于测试集中,从而得出每个样本因变量的期望值,并对预测效果进行评价。

4 实例分析

为了更好地验证 PCA-QLM 模型的有效性,本文将与 Logistic 模型(或 Logit 模型)以及 Probit 回归模型做性能对比。

实验中使用的预测性能评价指标如下:

(1) 均值绝对误差(Mean Absolute Error, MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_i'| \quad (8)$$

(2) 均值误差(Average Error, AE)

$$AE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y_i'}{y_i} \right| \quad (9)$$

(3) 均方百分比误差(Mean Square Percentage Error, MSPE)

$$MSPE = \frac{1}{n} \sqrt{\sum_{i=1}^n \left(\frac{y_i - y_i'}{y_i} \right)^2} \quad (10)$$

本文在实验之前首先对观察值进行预处理,将各测量指标进行归一化处理,并去除一些异常点。图 1、图 2 给出了两个数据集经过预处理后各个属性在每个样本点中的取值,可以看出经过归一化后,所有属性的取值都限定在了 $(0, 1)$ 之间。

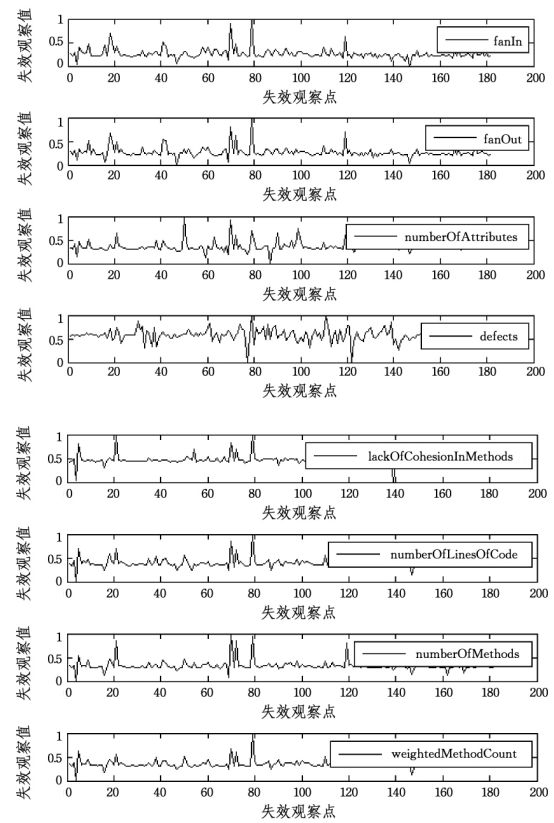


图 1 归一化后的 Eclipse JDT 数据集

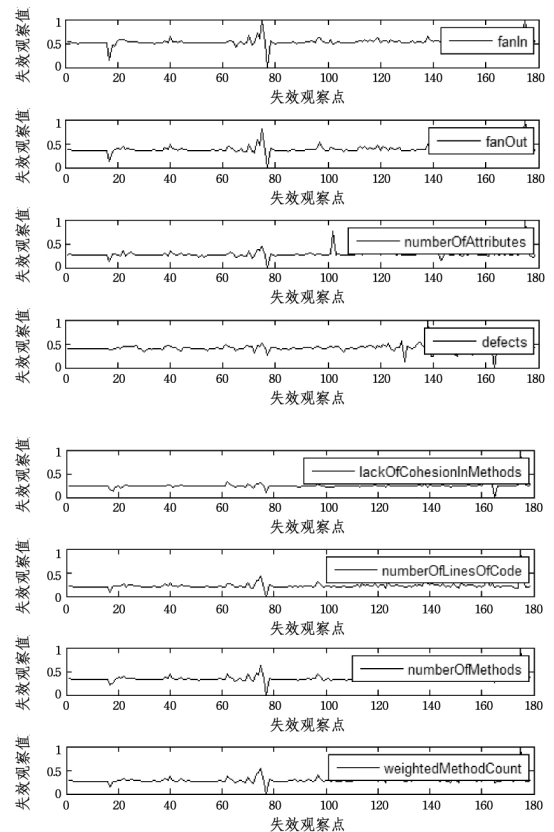


图 2 归一化后的 Eclipse PDE 数据集

数据集中的 Defects 属性作为因变量,其他 7 个属性作为自变量,对自变量矩阵使用主成分分析技术,筛选贡献率高的属性。表 2、表 3 给出了两个数据集的主成分分析结果,表中的“特征值”代表了矩阵正交化之后所对应特征向量对于整个矩阵的贡献程度;“贡献率”表示单个主成分综合原始变量信

息的能力,即解释原始变量的能力;“累积贡献率”表示的是前 m 个主成分解释原始变量的能力。

表 2 Eclipse JDT 数据集主成分分析的特征值及贡献率

主成分	特征值	贡献率	累积贡献率
1	5.5766	79.66	79.66
2	0.55735	7.9621	87.628
3	0.47133	6.7333	94.361
4	0.20181	2.883	97.244
5	0.11607	1.6582	98.903
6	0.051982	0.7426	99.645
7	0.024843	0.3549	100

表 3 Eclipse PDE 数据集主成分分析的特征值及贡献率

主成分	特征值	贡献率	累积贡献率
1	6.0466	86.38	86.38
2	0.4891	6.9872	93.367
3	0.29186	4.1694	97.536
4	0.1023	1.4615	98.998
5	0.029463	0.4209	99.822
6	0.028214	0.40306	99.822
7	0.012473	0.17819	100

从累积贡献率的值可以确定所选主成分的维度,在本实验中累积贡献率阈值取 85%,且维度 ≥ 2 ,因此两个数据集均取主成分 1 和主成分 2。

主成分分析的结果作为拟似然估计的自变量, $defects$ 属性值作为因变量进行拟似然估计,其中方差函数的替代函数采用核函数局部加权法所得的函数,此处的均值函数为恒等函数 $\mu_i(\beta) = h(z_i\beta) = z_i\beta$,其中 β 为 1×3 的未知向量, z_i 为二维的主成分矩阵。数据集 Eclipse JDT 的未知参数的拟似然估计值为:

$$\beta_0 = 0.018398, \beta_1 = -0.050981, \beta_2 = 0.532627$$

Eclipse PDE 数据集的未知参数拟似然估计值为:

$$\beta_0 = 0.01202, \beta_1 = 0.08229, \beta_2 = 0.40343$$

Eclipse JDT 数据集的前 100 组样本值为训练模型,后面的 82 组数据为检验模型。所得的实验结果与 Logistic 模型以及 Probit 模型作对比,所得结果如表 4 所列。

表 4 Eclipse JDT 数据集的实验结果对比

模型	AE	MSPE	MAE
PCA-QLM	0.17737	0.026576	0.095737
Logistic	0.17998	0.025745	0.10229
Probit	0.76887	0.08779	0.43328

Eclipse PDE 数据集的前 100 组样本值为训练模型,后面的 79 组数据为检验模型。所得的实验结果与 Logistic 模型以及 Probit 模型作对比,所得结果如表 5 所列。

表 5 Eclipse PDE 数据集的实验结果对比

模型	AE	MSPE	MAE
PCA-QLM	0.017115	0.0033731	0.0072402
Logistic	0.36223	0.069006	0.11658
Probit	0.54698	0.070242	0.209270

从对比结果可以看出,PCA-QLM 对软件失效预测是可行的,且综合性能相对于其他两种模型最优。在 Eclipse JDT 数据集中,PCA-QLM 的 MSPE 为 0.026576,与 Logistic 模型的 MSPE 仅相差 0.0008,说明 PCA-QLM 在预测的稳定性上与 Logistic 模型非常接近,且其他性能均优于另外两个模型。而在 Eclipse JDT 数据集中,PCA-QLM 的性能优势更加明显,MSPE 的值达到了 0.0033731,相对于另两种模型,在缺陷预测方面的稳定性优势更加明显。

陷预测方面的稳定性优势更加明显。

结束语 本文针对目前已有的缺陷预测模型中自变量联合分布难以确定的问题,提出了基于拟似然估计的软件缺陷预测模型(PCA-QLM)。该模型首先利用主成分分析技术进行特征子集的筛选,通过降维技术把多个变量化为少数几个主成分,从而一定程度上消除了各输入因子对结果的作用程度不同所造成的影响,在预测的精度和稳定性上得到进一步提高。用所得的主成分对拟似然方程中的参数进行估计,得到参数的拟似然估计值,进而估计软件失效的概率。PCA-QLM 模型不需要确定影响因子的联合分布,只需要确定其均值函数和方差函数即可,从而解决了传统预测模型中只能假定自变量的联合分布的局限性问题。

本文在两个真实的软件失效数据集的基础上,通过与经典的 Logistic 模型以及 Probit 模型作对比,从 AE、MSPE、MAE 3 个性能评价指标值可以看出基于拟似然的软件失效预测对软件缺陷预测是完全可行的,且性能优于这两种经典模型,预测的稳定性也很高。

参考文献

- [1] Catal C. Performance Evaluation Metrics for Software Fault Prediction Studies[J]. Acta Polytechnica Hungarica, 2012, 9(4):193-206
- [2] 杨晓杏. 基于度量元的软件缺陷预测技术[D]. 合肥:中国科学技术大学,2014:1-11
- [3] Zhang F,Mockus A,Keivanloo I, et al. Towards building a universal defect prediction model[C] // Working Conference on Mining Software Repositories, 2014:182-191
- [4] Suresh Y, Kumar L, Rath S K. Statistical and Machine Learning Methods for Software Fault Prediction Using CK Metric Suite: A Comparative Analysis[J]. International Scholarly Research Notices, 2014:1-15
- [5] Basili B V R, Briand L, Melo W L. A Validation of Object-Oriented Design Metrics As Quality Indicators[J]. IEEE Transactions on Software Engineering, 1996, 22(10):751-756
- [6] Briand L C, Wüst J, Daly J W, et al. Exploring the relationships between design measures and software quality in object-oriented systems[J]. Journal of Systems & Software, 2000, 51(3):245-273
- [7] Olague H M, Etzkorn L H, Gholston S, et al. Empirical Validation of Three Software Metrics Suites to Predict Fault-Prone-ness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Processes[J]. IEEE Transactions on Software Engineering, 2007, 33(6):402-419
- [8] Kanmani S, Uthariaraj V R, Sankaranarayanan V, et al. Object oriented software quality prediction using general regression neural networks[J]. Acm Sigsoft Software Engineering Notes, 2004, 29(5):1-6
- [9] Nagappan N, Williams L, Vouk M, et al. Early estimation of software quality using in-process testing metrics: A controlled case study[J]. Acm Sigsoft Software Engineering Notes, 2005, 30(4):1-7
- [10] D'Ambros M, Lanza M, Robbes R. An extensive comparison of bug prediction approaches[C] // IEEE Working Conference on Mining Software Repositories, 2010:31-41

(下转第 494 页)

表 3 插入程序中的错误信息

错误序号	文件名	文件类型	错误描述	错误类型	所在行号
1	mainproc. c	源文件	函数调用语句 Emu_Para_Verify()被注释	结构错误	78
2	init. c	源文件	变量 GPIO4 的赋值被注释	数据错误	13
3	init. c	源文件	变量 CKCFGWP 的赋值被注释	数据错误	113
4	init. c	源文件	else 分支无法到达(死代码)	结构错误	160
5	verify. c	源文件	函数调用语句 Veri_Etmr1_1o()被注释	结构错误	356
6	iokey. c	源文件	变量 DispErrorFlag 重复赋值	数据错误	81
7	iokey. c	源文件	循环变量 i 未初始化	数据错误	260
8	ex_yar. h	头文件	声明变量 ADCFG1 但未进行初始化	数据错误	35
9	define4. h	头文件	数组 Ascii[]声明但未初始化	数据错误	19
10	ex_func. h	头文件	删除一条函数声明	函数声明	58

表 4 C51 程序的测试结果

结构错误	序号	数据错误	序号	故障总数	注入错误总数	正确率 (%)
3	1,4,5	6	2,3,6,7,8,9	9	10	90

表 5 实验测试结果

实验号	测得数据				注入故障数	正确率 (%)
	故障总数	结构错误	数据错误	正确故障数		
1	9	3	6	9	10	90
2	4	2	2	3	3	100
3	2	0	2	1	2	50
4	6	2	4	6	7	86
5	7	2	5	7	8	87.5
6	5	2	3	5	5	100
7	9	3	6	8	10	80
...
28	2	0	2	2	2	100
29	6	3	3	6	6	100
30	3	2	1	3	3	100
总计	149	57	92	138	151	91

表 6 测试效果对比

测试工具	正确率 (%)	依赖的编译器	运行环境	是否画控制流图	是否检查语法错误	是否依赖环境
本方案	91	内部集成(简化的)	JVM	是	否	否
PC-lint	95	调用外部的编译器	Windows	否	是	是
Splint	90	内部集成	Windows/ UNIX	否	是	是

实验结果显示,分析测得的故障总数为 149,实际存在的故障总数为 151,正确测得的故障总数为 138,正确率为 91%。对比静态检测软件 PC-lint 的 95%和 Splint 的 90%,本方法的正确率介于两者之间,但是本方法具有不依赖于嵌入式软件开发环境的优势。由此可见,本文提出的基于静态结构模型的嵌入式软件分析方法在得出被测程序结构的同

时,能有效检测出程序中大部分的故障,简化了测试操作,为程序检测人员定位出了程序中潜在的代码隐患。

结束语 通过对嵌入式软件的静态结构模型的研究,在保证静态分析正确率的基础上,本文提出的基于文法简化和语句深度的静态结构模型嵌入式软件分析方法简化了分析过程,降低了分析工具对具体开发环境的依赖,提高了嵌入式软件分析的效率。基于该静态结构模型的嵌入式软件分析方法,能够提高嵌入式软件分析的自动化程度,减少嵌入式设备投入使用后因潜在隐患爆发而造成的损失。

参考文献

[1] 兰雨晴. 软件测试[J]. 计算机系统应用, 2003, 24(12): 66-68

[2] 刘佳欣. 嵌入式软件静态检测及自动化路径测试工具的研究与设计[D]. 广州: 华南理工大学, 2012

[3] 邓世伟. 嵌入式软件的测试方法和工具[J]. 单片机与嵌入式系统应用, 2001, 2(4): 140-142

[4] 赵立平. 控制流分析[J]. 计算机工程与应用, 1979(2)

[5] 周希. 基于静态分析的程序控制流图生成工具的设计与实现[D]. 广州: 中山大学, 2013

[6] 张广梅. 软件测试与可靠性评估[D]. 北京: 中国科学院计算技术研究所, 2006

[7] American National Standards Institute. ANSI C [EB/OL]. [2016-3-10]. https://zh.wikipedia.org/wiki/ANSI_C

[8] Kiczales G, lamping J, Mendhekar A. An Overview of Aspect [C] // J. Proc. 13th European Conference on Object-Oriented Programming, LNCS, Vol. 1241, Springer-Verlag, 2000: 220-242

[9] 张广梅. 数据流相关软件故障的静态检测[J]. 计算机辅助设计与图形学学报, 2005, 17(11): 2477-2482

[10] 王胜文. 采用数据流图的故障模型生成算法与应用[J]. 哈尔滨工业大学学报, 2009, 1(41): 118-121

(上接第 489 页)

[11] Wedderburn R W M. Quasi-likelihood function, generalized linear models and the Gauss-Newton method [J]. Biometrika, 1974, 61(3): 439-447

[12] Xia T, Jiang X, Wang X. Strong consistency of the maximum quasi-likelihood estimator in quasi-likelihood nonlinear models with stochastic regression[J]. Statistics & Probability Letters, 2015, 103(4): 391-400

[13] Su F, Chan K S. Quasi-likelihood estimation of a threshold diffusion process[J]. Journal of Econometrics, 2015, 189(2): 473-484

[14] Hu H C, Song L. Quasi-Maximum Likelihood Estimators in Generalized Linear Models with Autoregressive Processes[J]. 数学学报(英文版), 2014, 30(12): 2085-2102

[15] Liang K Y, Zeger S L. Logituidinal data analysis using generalized linear models[J]. Biometrika, 1986, 73(1): 13-22

[16] 陈夏, 陈希孺. 广义线性模型极大似然估计的强相合性和渐进正态性[J]. 应用概率统计, 2005, 21(3): 251-263

[17] 张三国, 廖源. 关于广义线性模型似然估计弱相合性的几个问题[J]. 中国科学 A 辑: 数学, 2007, 37(11): 1369-1377

[18] 朱仲义. 异方差非线性模型中的似然估计的渐进性质[J]. 河海大学学报, 1999, 24(2): 110-115

[19] Ahmed S E, Fallahpour S. Shrinkage estimation strategy in quasi-likelihood models[J]. Statistics & Probability Letters, 2012, 82(12): 2170-2179

[20] 高启兵, 吴耀华. 广义线性回归似然估计的渐进正态性[J]. 系统科学与数学, 2005, 25(6): 738-745