

软件可靠性增长模型的不确定性量化研究^{*}

许家俊, 姚淑珍

(北京航空航天大学 计算机学院, 北京 100191)

通讯作者: 许家俊, E-mail: xujiajun@buaa.edu.cn



摘要: 数量激增的软件系统被开发出来为用户提供了极大的便利,但也给系统开发带来了极大的不确定性.故障调试过程中的故障检测率(fault detection rate,简称 FDR)是不规律变化的,且通常被描述为白色噪音.白色噪音具有马尔可夫性,但在实践中,噪音出现非马尔可夫性是普通现象,而噪音呈现马尔可夫性仅仅是例外.在许多真实情况下,白色噪音的理想化假设是不足的.真正的不规则因素总是非马尔可夫相关性的.提出了一个新的模型来量化调试过程相关的环境不确定性因素.基于广泛应用于软件故障检测过程的非齐次泊松过程(non-homogeneous Poisson process,简称 NHPP)模型,将环境的不确定性考虑为任意分布和时间相关性的噪声.通过与一些现有模型比较,新的框架表现出更接近实际观测数据的特征.除了常用的关注故障数的平均值以外,还提供了公式以计算其累积密度函数(CDF)和概率密度函数(PDF),从而获得调试过程的完整统计信息.

关键词: 可靠性;不确定性;NHPP;噪音;相关性

中图法分类号: TP311

中文引用格式: 许家俊,姚淑珍.软件可靠性增长模型的不确定性量化研究.软件学报,2017,28(7):1746-1758. <http://www.jos.org.cn/1000-9825/5108.htm>

英文引用格式: Xu JJ, Yao SZ. Characterizing uncertainty of software reliability growth model. Ruan Jian Xue Bao/Journal of Software, 2017, 28(7): 1746-1758 (in Chinese). <http://www.jos.org.cn/1000-9825/5108.htm>

Characterizing Uncertainty of Software Reliability Growth Model

XU Jia-Jun, YAO Shu-Zhen

(School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

Abstract: More and more software systems have been developed to provide great flexibility to customers, but they also introduce great uncertainty to software development. The fault detection rate (FDR) within the fault detection process shows an irregular fluctuation and is usually modeled as a white noise. White noise is Markovian, but Non-Markov is the rule while Markov is the exception. In many cases the white noise idealization is insufficient, as real fluctuations are always correlated noise (non-Markovian noise). This study proposes a novel model to quantify the uncertainties associated with the debugging process. Based on the Non-homogeneous Poisson process (NHPP) model for software fault detection process, the environmental uncertainties are considered collectively as a noise of arbitrary distribution and correlation structure. Through a number of comparisons with existing methods, the new model exhibits a closer fitting to observation data. In addition to focusing on the mean value of detected-fault number, this work provides a formula to compute its cumulative density function (CDF) and probabilistic density function (PDF), thus encapsulating full statistical information of the debugging process.

Key words: reliability; uncertainty; NHPP; noise; correlation

软件可靠性工程为高质量软件系统的开发和应用带来了前所未有的机遇.在最近几十年,数量激增的软件

* 基金项目: 航空科学基金(2013ZC51023)

Foundation item: Aeronautical Science Foundation of China (2013ZC51023)

收稿时间: 2015-08-09; 修改时间: 2016-03-22; 采用时间: 2016-05-31; jos 在线出版时间: 2016-10-11

CNKI 网络优先出版: 2016-10-12 16:26:56, <http://www.cnki.net/kcms/detail/11.2560.TP.20161012.1626.023.html>

系统开发出来,为用户提供了极大的便利.随着这些软件系统日益成熟和复杂,其可靠性评估被定义为在一定条件和特定时间下无故障运行的概率^[1],成为开发商预测系统行为的关键和艰巨任务.

软件可靠性增长模型(software reliability growth model,简称 SRGM)使用故障检测时间或在特定时间内的故障数来评价系统的可靠性.故障检测过程被普遍认为是一个非齐次泊松过程(NHPP)^[2-6],并对故障检测率(fault detection rate,简称 FDR)有不同的假设.Goel 等人^[7]扩展了 Jelinsky 和 Moranda^[8]的早期工作,并提出了经典指数 SRGM,俗称为 Goel-Okumoto 模型(G-O 模型).一些学者^[9-13]随后断言 G-O 模型无法准确模拟故障检测的动态过程,取而代之,他们提出一个依赖时间的 FDR 来反映其随时间光滑/规律的变化.SRGM 包括延迟型模型^[9]、拐点 S 形模型^[10],包括工作量方程的 SGRM^[11]和扩展的具有变化点功能的 G-O 模型.总而言之,大多数模型假定软件测试检测到的累计故障数和检测时间存在一个确定的关系.

然而,在不确定性环境中,上述假设 FDR 为确定性的假设不再有效.在实际软件开发过程中,FDR 随着环境、人员和资源的变化而变化,而且未知突然变化,即不确定变化,从而显著影响传统故障检测模型.为了应对这些挑战,1994 年,Yamada 等人^[14]首次将不确定性干扰因素建模为白色噪音,提出了新的模型(白色噪音模型),也是最早使用白色噪音的模型.随后的大多数随机 NHPP 类模型都是在此基础上扩展的.这些模型一般将环境的集体因素描述为白色噪音,即马尔可夫噪音.与传统模型相比,这种假设噪音具有马尔可夫性的方法毫无疑问地通过采用简单的数学方法使模型更接近现实.而在实践中,噪音出现非马尔可夫性是普通现象,而噪音呈现马尔可夫性仅仅是例外^[15,16].在许多真实情况下,白色噪音的理想化假设是不足的:真正的不规则因素总是相关性的,且相应的随机波动通常为相关性噪声.这类相关性噪声符合非马尔可夫随机过程.在实际应用中,多个不具有可加性的相关性噪音是不可避免的.例如,来自环境等诸多干扰信号^[16].故障检测过程中的 FDR 从时域上看在任意两个相邻时刻的状态是相关的,即非马尔可夫噪音,使得 FDR 呈现为非马尔可夫相关性噪音.而只有更贴近实际的量化不确定性干扰因素,才能建立更贴近实际的软件可靠性模型,也才能更真实地描述故障检测过程,并计算软件可靠性和相关的性质.

在本文中,我们提出了用一个替代的随机框架来描述故障检测过程的模型,并且考虑了环境干扰因素的影响,将其建模为任意分布和时间相关性的噪音.第 1 节介绍研究现状.第 2 节对新的随机模型进行详细说明.第 3 节通过 4 组历史数据验证我们的方法并与传统方法进行比较.最后的结论在第 4 节中给出.

1 研究现状

到目前为止,各种 NHPP 类 SGRM 提出用以描述故障检测过程,进而评估系统可靠性.基于 Jelinsky 和 Moranda^[8]的早期工作,Goel 和 Okumoto 提出了一个指数型的软件可靠性模型(G-O 模型).模型假定检测到的故障的均值与剩余的故障数是成比例的,并且故障检测率 FDR 是一个恒定的数.这种模型提供了一类数学简化方法,不过,鉴于软件测试过程的复杂性,这种常数的 FDR 过于简化了故障检测过程的动态特征.

为了解决这个问题,平稳变化的 FDR 被结合到后来的模型中.Yamada 等人^[9]发现,故障检测率符合 S 型生长曲线,并提出了延迟的 S 形的软件可靠性增长模型.采用软件故障检测过程中的学习现象,Ohba^[10]后来提出了另一个 S 形的拐点的软件可靠性增长模型.Yamada 等人^[11]进一步将工作量函数与检测故障数的关系相结合集成到模型中.总体而言,平稳变化的 FDR 的假设在一定程度上符合观察结果,并在许多测试情况下显示了预测的有效性和准确性.

除了测试工作量,许多其他因素,如工作资源的分布、策略和运行环境等,都会影响故障检测过程,因此,软件 FDR 可能呈现规律性变化^[17].Huang 等人^[13]指出,软件故障检测与修复过程在开发与运行中是不同的.他们提出将统一理论和多重因素一起合并到 NHPP 类 SRGM 中,并发现 FDR 在检测过程中的平稳变化特征.Zhao 等人^[18]将环境因素纳入其模型中.考虑到故障检测和修正的时间延迟,Huang 等人^[12]提出了有限和无限的排队论的不完美软件可靠性增长模型.后来,Huang 等人^[19]还使用多个变化点来处理不完美的调试问题.软件 FDR 呈现规律性变化的可靠性模型是早期模型的改进^[20-22],它们通过改变点前后两个不同的 FDR 来描述故障点前后的环境变化.换句话说,当环境因素在故障检测阶段发生改变时,FDR 也可能明显地非单调变化^[23].

在不确定的环境中,我们不能再假定 FDR 是确定性的.Yamada 等人^[14]断言,如果软件系统的规模很大,则在故障排除阶段检测到的故障数也随之增大.同时,在故障检测过程中被校正和检测的故障数的差与初始的故障数相比很小.在实际故障检测过程中,FDR 随着环境、人员和资源的变化而变化,而且未知突然变化,即不确定变化.为描述这些不确定因素,1994 年,Yamada 等人^[14]首次将不确定性干扰因素建模为白色噪音,提出了白色噪音模型,也是最早使用白色噪音的模型.随后提出的大多数 NHPP 模型都是在此基础上扩展的,如 2005 年 Kapur 等人^[24]提出的 SGRM,2009 年 Kapur 等人^[25]提出的模型和 2010 年 Tamura 等人^[26]提出的 NHPP 模型等.白色噪声过程是一种理想化的数学模型,在任意两个相邻时刻的状态是不相关的,即随时间的起伏变化极快,将白色噪音定义为一维 Wiener 过程来描述故障检测过程,即故障检测过程中的失效率受白色噪音影响的过程或称为马尔可夫过程.这种随机模型毫无疑问地在数学方法上采用了更简单的故障估计理论.

从上面的讨论我们可以发现,已有的随机 NHPP 模型假设 FDR 的随机干扰因素为白色噪音.在实践中,噪音出现非马尔可夫性是普通现象,而噪音出现马尔可夫性仅仅是例外^[15,16].基于白色噪声的过程通常称为马尔可夫随机过程.然而,在许多实际情况下,白色噪声的理想化假设是不足的:真正的不规则因素总是相关性的,且相应的随机波动通常为相关性噪声.这类相关性噪声符合非马尔可夫随机过程^[15].在实际应用中,多个不具有可加性的相关性噪音是不可避免的,例如,来自环境等诸多干扰信号^[16].故障检测过程中的失效率从时域上看在任意两个相邻时刻的状态是相关的,即非马尔可夫噪音,使得软件 FDR 呈现出基于非马尔可夫噪音的相关且不规律变化.因此,研究在具有时间相关性噪音影响下的 NHPP 类 SRGM 符合实际的软件故障检测过程,具有重要的理论意义和实际价值.在研究过程中,我们希望引入时间相关性的环境噪声来进一步改善这种模型.

2 问题描述

2.1 模型描述

在本文中,我们采用连续时间 NHPP^[24-26]来建模故障计数过程 $\{N(t), t \geq 0\}$.NHPP 已广泛使用在许多实际领域^[11],这一模型假设:

- (1) 软件故障检测过程是一个具有连续状态空间的随机过程;
- (2) 系统的故障由系统中的剩余故障在一个随机时间引起;
- (3) 在时间区间 $(t, t+\delta t)$ 内,检测的平均故障数与剩余故障数是成比例的;
- (4) 一个检测的故障被发现后即被排除且没有带来新的故障.

那么,NHPP 过程可以表示为

$$P\{N(t) = k\} = \frac{[M(t)]^k}{k!} e^{-M(t)}, \quad k = 0, 1, 2, \dots \quad (1)$$

其中, $M(t)$ 是时间区间 $(0, t]$ 内检测的故障数, $M(t)$ 的时间演化经常被描述为故障失效强度函数 $\lambda(t)$.

$$\frac{dM}{dt} = \lambda(t) = r(t)(a - M) \quad (2)$$

其中, a 为初始故障总数,非负函数 r 代表在时间 t 的故障检测率.

由于故障检测率 $r(t)$ 受一些随机环境影响^[16],我们遵循以往的研究^[14,24-26],将 $r(t)$ 表示为一个平均值 $\langle r(t) \rangle$ 和一个均值为 0(随机)的波动 $r'(t)$ 的和.

$$r(t) = \langle r(t) \rangle + r'(t), \quad \langle r'(t) \rangle = 0 \quad (3)$$

与早期的将随机波动建模为白色噪音的方法不同,我们扩展了这个假设并描述噪音的两点协方差为

$$\langle r'(t)r'(s) \rangle = \sigma_r^2 \rho \left(\frac{t-s}{\tau} \right) \quad (4)$$

其中, σ_r^2 代表噪音的强度, ρ 表示相关性方程, t 与 s 为两个不同时间点, τ 为相关性时间长度.当噪音的相关性时间无限小时,相关性方程变成狄拉克函数(Dirac Delta function)并被称为“白色噪音”;其他时候,它被称为“彩色噪音”.

将式(3)代入式(2)中,我们得到一个不确定性 FDR 的随机微分等式:

$$\frac{dM}{dt} = [\langle r(t) \rangle + r'(t)](a - M) \tag{5}$$

故障检测率 $\langle r(t) \rangle$ 表示一个故障的平均失效率,在不同的模型中有不同的表现形式.在表 1 中,本文列出了 5 个基于 NHPP 模型的具有确定性 $\langle r(t) \rangle$ 的数学表达式,即 Goel-Okumoto SRGM(G-O model),generalized G-O SRGM(gG-O Model),Delay S-shaped SRGM (DSS model),Inflection S-shaped SRGM(ISS model),Model with change-point(CP)和 Log-Exponential SRGM(log-exp).其中, b 为 $r(t)$ 中的常量, d 为 $r(t)$ 中的塑形因子, w 为权重, t_{cp} 为改变点的时间, b_1 和 b_2 分别为改变点 t_{cp} 前后的 FDRs.

Table 1 The FDR (fault detection rate) of deterministic SRGMs

表 1 确定性 SRGMs 的故障检测率 FDR $\langle r(t) \rangle$

模型	FDR $\langle r(t) \rangle$
Goel-Okumoto SRGM (G-O model)	b
generalized G-O SRGM (gG-O Model)	bt^d
Delay S-shaped SRGM (DSS model)	$b^2t / (1 + bt)$
Inflection S-shaped SRGM (ISS model)	$b / (1 + we^{-bt})$
Model with change-point (CP)	$\begin{cases} b_1, & t < t_{cp} \\ b_2, & t > t_{cp} \end{cases}$
Log-Exponential SRGM (log-exp)	$dbe^bt (e^bt - 1)^{(d-1)} / [1 + (e^bt - 1)^d]$

这里需要指出的是,故障检测率模型(5)描述了一个完美的调试过程.在许多情况下,时间延迟、故障排除失败和新故障的引入,将导致出现不完善的调试问题^[12,13,18,19].在目前的研究中,我们的重点在于提出一个量化环境因素不确定性的框架.为不失一般性,此框架随后也可以扩展到不完善的调试过程.

2.2 模型求解

我们采用广义 G-O 模型^[27]中的方法来表示平均故障检测率与剩余的故障数关系的函数.

$$\langle r(t) \rangle = bt^d \tag{6}$$

其中, d 是一个塑形参数,而常量 b 代表了故障检测率,且 $b \in (0, 1)$.其中, $b \rightarrow 0$ 说明模型的故障检测能力很差.

使用分离变量法,一个累计故障数 M 的解析解可推导为

$$M = a \left\{ 1 - \exp \left[- \int_0^t r'(t') dt' - \frac{bt^{d+1}}{d+1} \right] \right\} \tag{7}$$

值得一提的是,解(7)包括一个随机干扰随时间的积分, $I_{r'} = \int_0^t r'(t') dt'$,因而变成了一个随机积分问题.在相关性时间长度 τ 与我们关注的时间 t 的基础上,我们采用 3 个不同的近似方案^[28]来计算随机积分的概率密度函数 $f_{I_{r'}}$.详细的推导过程在附录 A1 中给出.

现在,故障数 M 的均值和方差可推导为

$$\langle M \rangle = \int_{-\infty}^{\infty} a \left[1 - \exp \left(-I_{r'} - \frac{bt^{d+1}}{d+1} \right) \right] \cdot f_{I_{r'}}(I_{r'}, t) dI_{r'} \tag{8a}$$

$$\sigma_M^2 = \int_{-\infty}^{\infty} \left\{ a^2 \left[1 - \exp \left(-I_{r'} - \frac{bt^{d+1}}{d+1} \right) \right]^2 - \langle M \rangle^2 \right\} \cdot f_{I_{r'}}(I_{r'}, t) dI_{r'} \tag{8b}$$

这里,我们注意到噪音的分布函数 $f_{r'}$ 或噪音的相关性方程 ρ 没有进一步的假设.在随后的数值研究中,除非特别说明,否则均用一个高斯函数来描述随机干扰,即 $r' \sim N(0, \sigma_r^2)$,且噪音的相关性方程为指数函数,即 $\rho(s, t; \tau) = \exp(-|t-s|/\tau)$.

3 数值仿真

3.1 数据集

为了验证模型,我们用4组软件失效数据集:(1) DS1来自一个基于Web的集成会计ERP系统;(2) DS2来自一个开源项目管理软件^[29,30];(3) DS3和(4) DS4来自坦丁姆计算机公司(Tandem Computers Inc.)的第1次和第4次软件发布版本^[31].在4组数据中,两个基于广域网的系统(DS1和DS2)更容易受环境因素影响,而Tandem软件在一个更封闭和稳定的环境中开发.因此,我们通过噪声方差来体现这种差异,即基于Web的系统具有更大的噪声 σ_r^2 .数据集DS1,DS2,DS3和DS4的测试时间分别为60(months),29(weeks),20(weeks)和19(weeks).如需详细信息,请参考列于附录A2中的失效数据.

3.2 模型参数评估

我们的相关性模型中有5个输入参数:初始故障总数 a ,故障检测率 b ,塑形参数 d ,噪音的强度 σ_r^2 和相关时间 τ .给定一组数据,前3个参数可以使用矩估计法、最小二乘法、最大似然法和贝叶斯方法获得^[32-34]且相应的支持工具包括SPSS等.对于噪声参数,我们的相关性噪音强度 σ_r^2 继承了之前的白色噪声研究中的强度^[14,24,25];关于噪声的相关时间,由于故障修复时间以天为单位^[35],因此,我们设置 $\tau=0.1$ (month)或(week)供以后分析使用.

3.3 模型验证

为了验证时间相关性模型(correlated model),本文与经典的确定性SGRM模型(G-O model)和流行的白色噪声随机模型(white-noise model)进行了对比.图1展示了在不同的数据集和测试时间上,相关性模型、G-O模型和白色噪声模型对故障数 $\langle M \rangle$ 的预测结果.所有3个模型的参数都已经调整为最适合它们的数值.如图1所示,虽然相关的模型高估了DS1的故障数,也低估了DS2,DS3和DS4的故障数,但是整体上,与G-O模型和白色噪声模型相比,本文的预测结果最匹配于4组实际数据.

为了更好地验证相关性模型的性能,我们采用如下5个评价标准.

a) 预测相对误差(predicted relative error,简称PRE)

PRE是在时间 t 时观察值和估计值的相对距离,通常被称为预测效率.

$$PRE(t) = \frac{\langle M(t) \rangle - q(t)}{q(t)} \quad (9)$$

其中, q 是观测数据中 t 时刻的故障均值.一个完美的数据拟合使得 $PRE(t)=0$.^[36,37]

图2展示了在不同的测试时间上,相关性模型、G-O模型和白色噪声模型的PRE结果.3个模型具有类似的趋势:开始时具有很大的预测误差(距0的偏差),稍后逐渐趋于可观测值.这个结果是符合预期的,因为在早期只有很少的数据来衡量模型的参数;随着时间的消耗,更多的数据变得可用,模型的精度有所提高,因此,它们的PRE接近0.总体而言,相关性模型产生的PRE最小.

b) 均方误差(mean square error,简称MSE)

$$MSE = \frac{1}{K} \sum_{i=1}^K [\langle M(t_i) \rangle - q(t_i)]^2 \quad (10)$$

这里, K 代表观测点的总数.MSE的值越小说明模型的预测性越好^[26,37].

c) 决定系数(coefficient of determination,简称 R^2)

R^2 度量模型如何线性回归逼近真实的数据点. $R^2=1$ 说明模型的预测结果完美地接近实际.

$$R^2 = 1 - \frac{\sum_{i=1}^K [\langle M(t_i) \rangle - q(t_i)]^2}{\sum_{i=1}^K [q(t_i) - \langle q \rangle]^2}, \quad \text{where } \langle q \rangle = \frac{1}{K} \sum_{i=1}^K q(t_i) \quad (11)$$

d) 误差均方根(root mean square prediction error,简称RMSPE)

RMSPE包含两个附加指标:平均的预测错误(the average of prediction errors,简称Bias)和平均相对误差(the

predicted relative variation,简称 PRV).

$$RMSPE = \sqrt{Bias^2 + PRV^2} \tag{12a}$$

其中,

$$Bias = \frac{1}{K} \sum_{i=1}^K [\langle M(t_i) \rangle - q(t_i)] \tag{12b}$$

$$PRV = \sqrt{\frac{\sum_{i=1}^K [\langle M(t_i) \rangle - q(t_i) - Bias]^2}{K - 1}} \tag{12c}$$

Bias,PRV 和 RMSPE 的值越小,代表模型越接近实际.

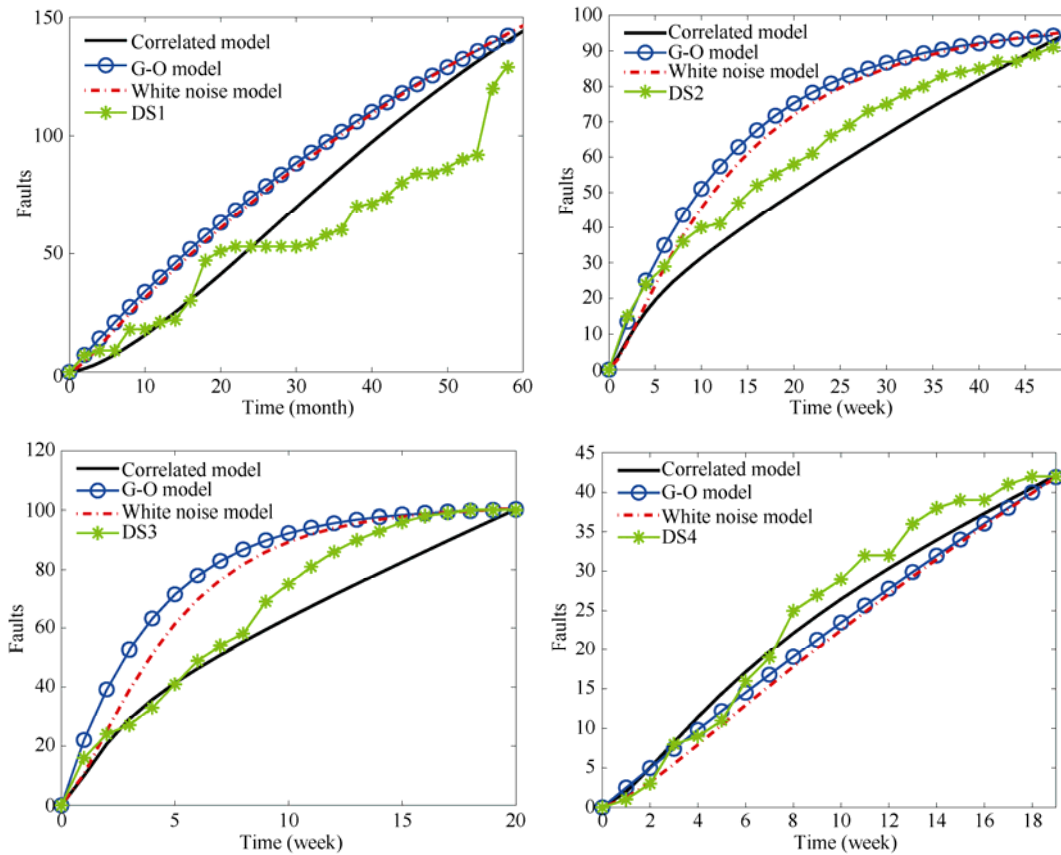


Fig.1 Expected number of faults, $\langle M \rangle$, computed from the correlated model, G-O model and white-noise model, are compared with four data sets

图1 在4个数据集基础上,相关性的模型、G-O模型及白噪声模型分别计算出的预期故障数 $\langle M \rangle$

相关性模型、G-O模型和白色噪声模型在数据集 DS1,DS2,DS3 和 DS4 上的评估结果展示在表2中.尽管在 DS1 中 PRV 有一个较高的值,但是相关性模型在4个数据集上对所有指标都展示出了对实际数据的最好的拟合度(goodness-of-fit).

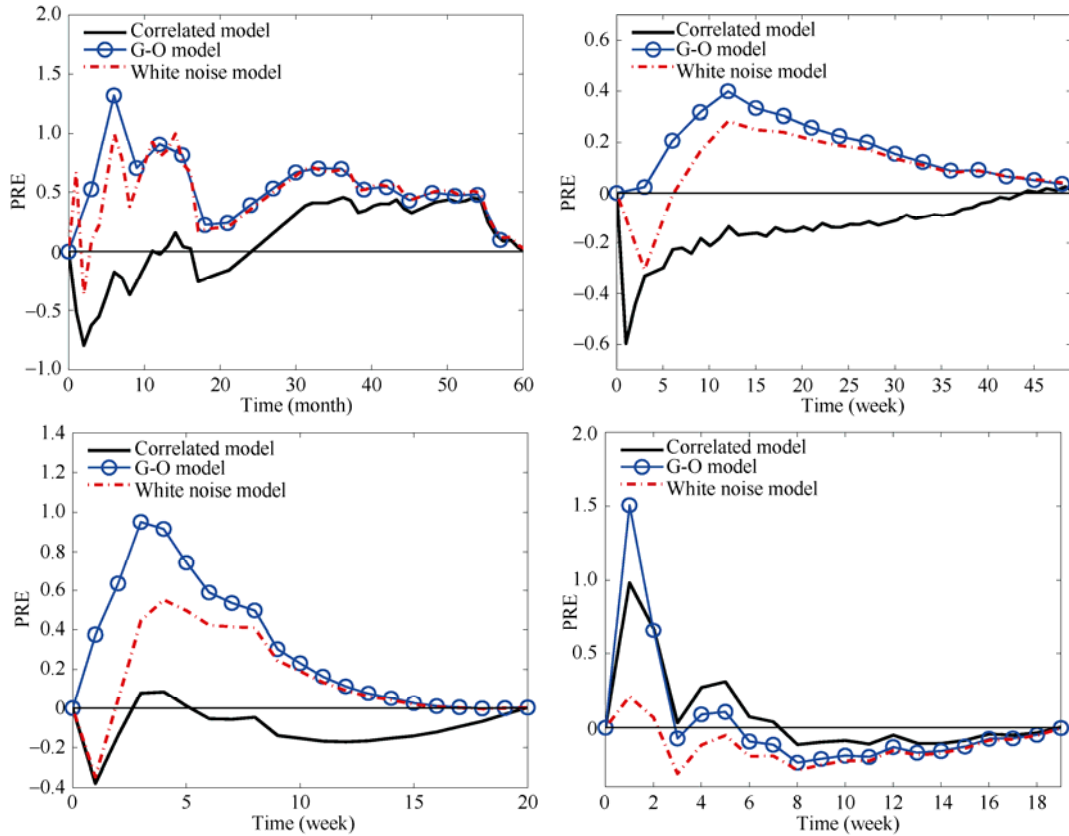


Fig.2 Predicted relative error (PRE) for correlated model, G-O model and white-noise model, from four data sets
 图2 在4个数据集上分别计算相关性模型、G-O模型与白噪声模型的预测相对误差PRE值

Table 2 The MSE, R^2 , Bias, PRV and RMSPE results of correlated model, G-O model and white noise model for DS1, DS2, DS3 and DS4

表2 相关性模型、G-O模型和白色噪声模型在DS1,DS2,DS3与DS4上的MSE, R^2 ,Bias,PRV及RMSPE结果

模型	数据集	参数设置	MSE	R^2	Bias	PRV	RMSPE
相关性模型	DS1	$a=445, b=0.04047, d=0.028, \sigma_r^2=0.0002236, \tau=0.1$	411.840 4	0.653 1	12.542 7	16.088 4	20.399 9
	DS2	$a=308, b=0.5446, d=0.026, \sigma_r^2=0.0002228, \tau=0.1$	46.194 3	0.913 3	-5.944 2	3.329 8	6.813 3
	DS3	$a=618, b=0.846, d=0.025, \sigma_r^2=0.0002219, \tau=0.1$	78.658 2	0.903 2	-6.675 6	5.990 8	8.969 5
	DS4	$a=269, b=0.3046, d=0.024, \sigma_r^2=0.0002207, \tau=0.1$	5.928 4	0.968 2	-0.974 3	2.292 6	2.491 0
G-O模型	DS1	$a=249, b=0.0146$	847.959 9	0.285 8	25.617 9	13.961 9	29.175 5
	DS2	$a=97, b=0.0746$	123.789 6	0.767 8	9.694 0	5.517 1	11.154 0
	DS3	$a=101, b=0.246$	318.577 2	0.608 1	13.575 3	11.889 3	18.045 6
	DS4	$a=173, b=0.0146$	15.314 7	0.917 8	-2.714 6	2.896 1	3.969 4
白色噪声模型	DS1	$a=522, b=0.0126, d=0.028, \sigma_r^2=0.0002236$	816.378 4	0.312 4	24.601 8	14.654 3	28.635 6
	DS2	$a=197, b=0.06246, d=0.026, \sigma_r^2=0.0002228$	82.049 3	0.846 1	7.077 5	5.711 7	9.094 8
	DS3	$a=203, b=0.226, d=0.025, \sigma_r^2=0.0002219$	157.913 9	0.805 8	8.832 9	9.170 6	12.732 6
	DS4	$a=377, b=0.0126, d=0.024, \sigma_r^2=0.0002207$	20.409 4	0.890 5	-3.698 1	2.666 1	4.558 9

3.4 软件可靠性和软件可靠性增值率

我们已在 4 组数据集上验证了相关性模型,现在着手利用这个模型来预测评估软件可靠性.由于环境噪音 $r(t)$ 的影响,检测到故障的累积数量在一个时间跨度内将不会是确定的值.软件可靠性是从在 NHPP 模型基础上的故障累计数和检测时间计算而来.换言之,环境的不确定性影响了故障计数过程(NHPP),并进一步影响了软件的可靠性.本节的目的是估算在这个不确定性调试过程下的软件可靠性.为不失一般性,随后的研究中,相关性模型将采用以下参数: $a=308, b=0.005446, d=0.026, \sigma_r^2=0.0002228$ 和 $\tau=0.1$.

软件可靠性 $R(t;T)$ 被广泛地定义为在时间区间 $[T, T+t]$ 内软件不发生故障的概率.基于 NHPP 的软件可靠性可以表示为

$$R(t;T) = e^{-[M(T+t)] - M(T)} \tag{13}$$

使用相关性模型,软件可靠性 $R(t;T)$ 在时间 $T=2, 10$ 和 50 周的 3 个截图如图 3 所示.根据表达式(13),当在 T 固定且在相对于 T 的时间 $t=0$ 时, $R(t;T) = e^{-[M(T)] - M(T)} = e^0 = 1$. 在 3 个时间戳上的可靠性在初始时具有峰值,但是随后逐渐减慢,最后在 $t \rightarrow \infty$ 时 $R(t;T)$ 接近于 0.这是我们预计的结果,因为在早期,大部分的故障尚未被检测到,随着时间的流逝,更多的 bug 被检测出来,导致软件可靠性的下降;当系统所有的故障都被清除时, $R(t;T)$ 达到平稳状态.

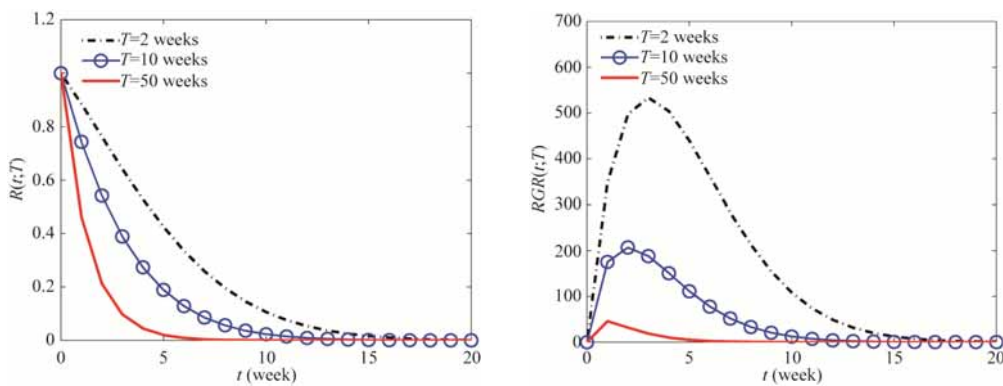


Fig.3 Temporal evolutions of software reliability, $R(t;T)$, and reliability growth rate, $RGR(t;T)$, at time $T=2, 10, 50$ weeks, using the correlated model with parameters $a=308, b=0.005446, d=0.026, \sigma_r^2=0.0002228$, and $\tau=0.1$

图 3 软件可靠性 $R(t;T)$ 和软件可靠性的增长速度 $RGR(t;T)$, 在时间 $T=2, 10, 50$ 周, 使用参数为 $a=308, b=0.005446, d=0.026, \sigma_r^2=0.0002228$ 和 $\tau=0.1$ 的相关性模型的时间演化

为了更进一步地研究可靠性,我们现在评估软件可靠性的增长率 $RGR(t;T)$.当时间间隔 t 趋近于 0 时,平均增长率的极限称为瞬时增长率.此时, $RGR(t;T)$ 也可表示为

$$RGR(t;T) = \frac{d}{dt} R(t;T) \tag{14}$$

软件可靠性增长率与可靠性无必然联系,可靠性增长率很大时,可靠性可以很小;可靠性很大时,可靠性增长率也可以很小.当软件的可靠性增长率加速度为 0 时,软件的可靠性就均匀增加.当软件的可靠性增长率保持大小不变时,软件的可靠性就匀速改变.在软件调试过程中,如果软件的可靠性增长率不为 0,则软件的可靠性就处于变化状态.如果软件的可靠性增长率大于 0,则其可靠性快速增加;如果软件的可靠性增长率小于 0,则其可靠性快速减小.将软件可靠性增长率作为可靠性变化的度量标准,是观察软件可靠性是否稳定的指标.

以本文的相关性模型为例, $RGR(t;T)$ 可通过模型公式解(8a)和(8b)来计算.

$$RGR(t;T) = aR \int_{-\infty}^{\infty} \frac{e^{-\frac{I_r^2}{2\sigma_r^2}}}{\sigma_r \sqrt{2\pi}} \left[\frac{d\sigma_r}{dt} \left(1 - e^{-I_r - \frac{br^{d+1}}{d+1}} \right) \frac{\sigma_r^2 - I_r^2}{\sigma_r^3} - b(T+t)^d e^{-I_r - \frac{br^{d+1}}{d+1}} \right] dI_r \tag{15}$$

其中,方差 $d\sigma_f/dt$ 的推导可参见附录 A1 中的表 3.

图 3 显示了通过模型(15)计算的 $T=2, 10$ 和 50 周时对应的软件可靠性增长率 RGR. 尽管最初表现为上升趋势,但是 3 个时段的 RGR 随后都收敛到 0 并保持平稳. 这也和故障调试过程是一致的,由于更多的故障随着检测时间的推进被排除. 当所有的故障都被排除时,软件可靠性增长率 RGR 达到 0. 根据表达式(15),当 RGR 的初始值为 0 时,在我们的例子中,是第 80 周.

3.5 检测故障的概率分布密度函数(PDF)和累积密度函数(CDF)

传统的软件可靠性评估依赖于平均故障检测 $\langle M \rangle$, 通过提出一个随机 FDR, 人们可以得到 M 完整的统计信息, 从而可以对软件系统进行概率风险评估. 对公式解(7)进行统计分析, 我们能够获得故障移除的概率分布密度函数(probabilistic density function, 简称 PDF) $f_M(m; t)$ 和累积密度函数(cumulative density function, 简称 CDF) $F_M(m; t)$ 且 $f_M dm = f_{I_r} dI_r$.

$$f_M(m; t) = \begin{cases} \frac{e^{-\frac{1}{2}\sigma_f^2}}{\sigma_f(a-m)\sqrt{2\pi}} \left[\ln\left(1-\frac{m}{a}\right) + \frac{bt^{d+1}}{d+1} \right]^2, & m < a \\ 0, & m > a \end{cases} \quad (16)$$

$$F_M(m; t) = \begin{cases} \frac{1}{2} \left[1 - \operatorname{erf} \left(\frac{\ln\left(1-\frac{m}{a}\right) + (bt^{d+1})/(d+1)}{\sigma_f\sqrt{2}} \right) \right], & m < a \\ 1, & m > a \end{cases} \quad (17)$$

其中, $\operatorname{erf}(\cdot) \in [-1, 1]$, 代表误差函数(error function), σ_f^2 表示随机积分 I_r 的方差, 可见附录 A1.

使用参数 $a=308, b=0.005446, d=0.026, \sigma_r^2=0.0002228$ 和 $\tau=0.1$ 时, PDF 和 CDF 在时间 $t=2, 10$ 和 50 周的 3 个快照分别如图 4 所示.

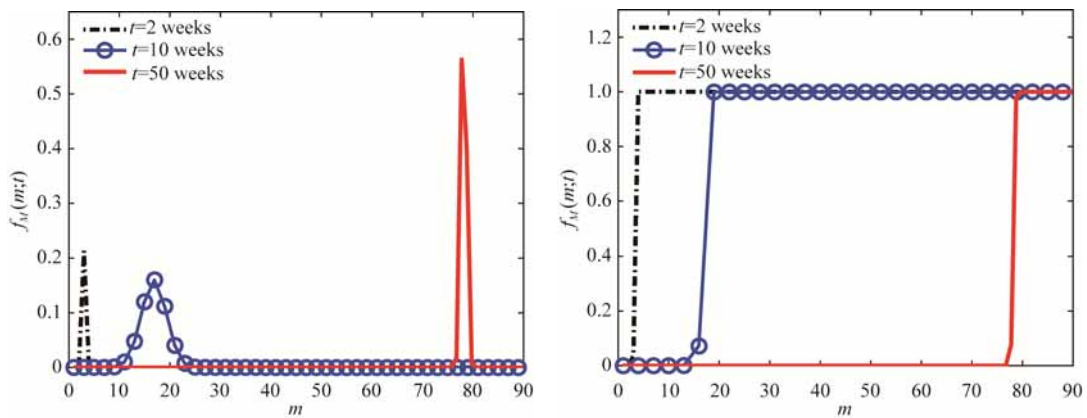


Fig.4 Snapshots of the PDF, $f_M(m; t)$, and the CDF, $F_M(m; t)$, at time $t=2, 10$ and 50 weeks, using parameters $a=308, b=0.005446, d=0.026, \sigma_r^2=0.0002228$, and $\tau=0.1$

图 4 在时间 $t=2, 10$ 和 50 周, 使用参数 $a=308, b=0.005446, d=0.026, \sigma_r^2=0.0002228$ 和 $\tau=0.1$ 时, 概率密度函数(PDF)和累积密度函数(CDF)的快照

计算 PDF 和 CDF 的主要目的是研究故障数在受到环境不确定性噪声 $r(t)$ 影响时的概率分布情况. 由于故障 PDF 和 CDF 是根据随机微分等式(5)逐步推导而来, 根据随机微分等式可知初始条件时间 t 对 PDF 和 CDF 的值有较大的影响. 我们可以使用时间 t 来观测, 在 $t=2, 10$ 或 50 周时故障数的可能情况. 通过给定的初始条件时间 t , 模型可以计算 $t=2, 10$ 或 50 周时故障的概率密度函数. 以 $t=2$ 为例, $t=2$ 时的 PDF 给出了此时故障数的可能的分布情况; $t=2$ 时的概率密度函数也给出了 $t+8$ 的故障数的确定值和这个值的概率, 但是这个确定值是在 $t=2$

时的初始条件下计算出的结果.在 $t=10$ 的初始条件下计算的故障数也是一个概率密度函数,这个概率密度函数和 $t=2$ 时计算的 $2+8$ 时所计算的故障数的确定值可能不存在等同关系.

由于受不确定性噪音的影响,故障数不再是一个确定值,变成了一个连续型随机变量(PDF).我们预测的不确定性展示在 PDF 宽度上.因为 PDF 给出了每个给定的时间条件下故障的可能性,所以每个估计的故障下限和上限可通过跟踪 PDF 得到.通过分析一个特定时间的 PDF,可找到检测故障最小数目和最大数目.例如,它们在 10 周时分别是 4 和 21.CDF 的值先从 2 周增加到 10 周;之后落在 50 周,因为此时大部分的故障都已被检测到.图 4 所示的 PDF 和 CDF 给出了在噪音影响下 T 时刻故障数的可能性.当不确定性噪音对故障检测过程影响消失时,即 PDF 宽度为 0 时,说明几乎所有的故障都已被检测到,而这个时间是 80 周.

4 结束语

现有的 NHPP 类 SRGM 假定检测到故障的累积数和软件调试的时间之间存在确定性的关系,应用具有一定的局限性.对于不确定环境影响,本文将环境因素统一表示成任意分布和相关性结构的噪声,提出了一种用于量化环境中软件故障检测过程不确定性的相关性模型,并进一步推导了故障检测数的显式解:均值、方差、PDF 和 CDF.经本文分析后可以得出以下主要结论:(1) 在 4 组实际数据集上,本文所提出的软件可靠性增长模型比经典的 G-O 模型和白色噪声模型具有更好的拟合度;(2) 本文所提出的软件可靠性增长模型在预测初期不确定性较大,但随着检测时间的推进而逐渐减少,因此初始阶段需要更多的维护;(3) 故障检测人员可借助本文所提出的软件可靠性增长模型评估检测过程中的小概率事件,以完成软件的不确定性风险评估.

References:

- [1] IEEE STD-729-1991. Standard Glossary of Software Engineering Terminology. 1991.
- [2] Lyu M. Handbook of Software Reliability Engineering. Los Alamitos: McGraw-Hill, 1996.
- [3] Pham H. System Software Reliability. Springer-Verlag, 2006. [doi: 10.1007/1-84628-295-0]
- [4] Carnes P. Software reliability in Weapon systems. In: Proc. of the the 8th Int'l Symp. on Software Reliability Engineering. 1997. 114–115. [doi: 10.1109/CSSRE.1997.637851]
- [5] Schneidewind N, Keller T. Application of reliability models to the space shuttle. IEEE Software, 1992,9(4):28–33. [doi: 10.1109/52.143099]
- [6] Almering V, Genuchten M, Cloudt G, Sonnemans P. Using software reliability growth models in practice. IEEE Software, 2007, 24(6):82–88. [doi: 10.1109/MS.2007.182]
- [7] Goel A, Okumoto K. Time-Dependent error-detection rate model for software reliability and other performance measures. IEEE Trans. on Reliability, 1979,R-28(3):206–211. [doi: 10.1109/TR.1979.5220566]
- [8] Jelinski Z, Moranda P. Software reliability research. In: Proc. of the Statistical Methods for the Evaluation of Computer System Performance. Academic Press, 1972. 465–484. [doi: 10.1016/B978-0-12-266950-7.50028-1]
- [9] Yamada S, Ohba M, Osaki S. S-Shaped reliability growth modeling for software error detection. IEEE Trans. on Reliability, 1983, 32(5):475–484. [doi: 10.1109/TR.1983.5221735]
- [10] Ohba M. Inflection S-shaped software reliability growth model. LNCS, 1984,235:144–162. [doi: 10.1007/978-3-642-45587-2_10]
- [11] Yamada S, Hishitani J, Osaki S. Software-Reliability growth with a weibull test-effort: A model and application. IEEE Trans. on Reliability, 1993,42(1):100–106. [doi: 10.1109/24.210278]
- [12] Huang C, Huang W. Software reliability analysis and measurement using finite and infinite server queuing models. IEEE Trans. on Reliability, 2008,57(1):192–203. [doi: 10.1109/TR.2007.909777]
- [13] Huang C, Hung T. Software reliability analysis and assessment using queueing models with multiple change-points. Computers and Mathematics with Applications, 2010,60(7):2015–2030. [doi: 10.1016/j.camwa.2010.07.039]
- [14] Yamada S, Nishigaki A, Kimura M. A stochastic differential equation model for software reliability assessment and its goodness of fit. Int'l Journal of Reliability and Applications, 2003,4(1):1–11.
- [15] Kampen NGV. Remarks on non-Markov processes. Brazilian Journal of Physics, 1998,28(2):90–96. [doi: 10.1590/S0103-97331998000200003]
- [16] Luo S, Fu S, Song H. Quantifying non-markovianity via correlations. Physical Review A, 2012,86(4):3397–3403. [doi: 10.1103/PhysRevA.86.044101]

- [17] Zhang X, Pham H. An analysis of factors affecting software reliability. *Journal of Systems and Software*, 2000,50(1):43–56. [doi: 10.1016/S0164-1212(99)00075-8]
- [18] Zhao J, Liu HW, Cui G, Yang XZ. Software reliability growth model with change-point and environmental function. *Journal of Systems and Software*, 2006,79(11):1578–1587. [doi: 10.1016/j.jss.2006.02.030]
- [19] Huang C, Lyu M. Estimation and analysis of some generalized multiple change-point software reliability models. *IEEE Trans. on Reliability*, 2011,60(2):498–514. [doi: 10.1109/TR.2011.2134350]
- [20] Zou F. A change-point perspective on the software failure process. *Software Testing, Verification and Reliability*, 2003,13(2): 85–93. [doi: 10.1002/stvr.268]
- [21] Shyur H. A stochastic software reliability model with imperfect debugging and change-point. *Journal of Systems and Software*, 2003,66(2):135–141. [doi: 10.1016/S0164-1212(02)00071-7]
- [22] Huang C. Performance analysis of software reliability growth models with test-effort and change-point. *Journal of Systems and Software*, 2005,76(2):181–194. [doi: 10.1016/j.jss.2004.04.024]
- [23] Zhao M. Change-Point problems in software and hardware reliability. *Communications in Statistics-Theory and Methods*, 1993, 22(3):757–768. [doi: 10.1080/03610929308831053]
- [24] Tamura Y, Yamada S. A flexible stochastic differential equation model in distributed development environment. *European Journal of Operational Research*, 2006,168(1):143–152. [doi: 10.1016/j.ejor.2004.04.034]
- [25] Kapur PK, Anand S, Yamada S, Yadavalli VSS. Stochastic differential equation-based flexible software reliability growth model. *Mathematical Problems in Engineering*, 2009,2009(4):266–287. [doi: 10.1155/2009/581383]
- [26] Tamura Y, Yamada S. Performance evaluation of reliability assessment method based on stochastic differential equation model for a large-scale open source solution. *Int'l Journal of System Assurance Engineering and Management*, 2010,1(4):324–329. [doi: 10.1007/s13198-011-0035-z]
- [27] Xie M. *Software Reliability Modeling*. World Scientific Publishing, 1991. [doi: 10.1142/1390]
- [28] Wang P, Tartakovsky DM, Jarman JKD, Tartakovsky AM. CDF solutions of buckley-leverett equation with uncertain parameters. *Multiscale Modeling and Simulation*, 2013,11(1):118–133. [doi: 10.1137/120865574]
- [29] Hsu CJ, Huang CY, Chang JR. Enhancing software reliability modeling and prediction through the introduction of time-variable fault reduction factor. *Applied Mathematical Modelling*, 2011,35(1):506–521. [doi: 10.1016/j.apm.2010.07.017]
- [30] SourceForge.net. An open source software website. 2008, <http://sourceforge.net>
- [31] Wood A. Predicting software reliability. *IEEE Computer*, 1996,29(11):69–77. [doi: 10.1109/2.544240]
- [32] Massey W, Parker G, Whitt W. Estimating the parameters of a nonhomogeneous Poisson process with linear rate. *Telecommunication Systems*, 1996,5(4):361–388. [doi: 10.1007/BF02112523]
- [33] Leite J, Rodrigues J, Milan L. A Bayesian analysis for estimating the number of species in a population using nonhomogeneous poisson process. *Statistics & Probability Letters*, 2000,48(2):153–161. [doi: 10.1016/S0167-7152(99)00198-4]
- [34] Yang Z, Liu C. Estimating parameters of OHBA three-parameters NHPP models. In: *Proc. of the Computer Science and Service System*, 2011. 1259–1261. [doi: 10.1109/CSSS.2011.5974854]
- [35] Weib C, Premraj R, Zimmermann T, Zeller A. How long will it take to fix this bug. In: *Proc. of the 29th Int'l Conf. on Software Engineering Workshop on Mining Software Repositories*. 2007,9(8):1. [doi: 10.1109/MSR.2007.13]
- [36] Musa J, Iannino A, Okumoto K. *Software Reliability: Measurement, Prediction, Application*. New York: McGraw-Hill, 1989.
- [37] Pillai K, Nair VSS. A model for software development effort and cost estimation. *IEEE Trans. on Software Engineering*, 1997, 23(8):485–497. [doi: 10.1109/32.624305]
- [38] Huang CY, Lin CT. Software reliability analysis by considering fault dependency and debugging time lag. *IEEE Trans. on Reliability*, 2006,55(3):436–450. [doi: 10.1109/TR.2006.879607]
- [39] Ditlevsen O, Mohr G, Hoffmeyer P. Integration of non-Gaussian fields. *Probabilistic Engineering Mechanics*, 1996,11(1):15–23. [doi: 10.1016/0266-8920(95)00023-2]
- [40] Durrett R. *Probability: Theory and Examples*. New York: Cambridge University Press, 2010.
- [41] White H. *Asymptotic Theory for Econometricians*. New York: Academic Press, 2001.

附录

A1 随机场积分

这里,我们采用文献[28]的结果来计算随机积分 I_r' ,并列在表 3 中.

Table 3 Approximation of distribution for the random integral:

$$I_r = \int_0^t r'(t')dt' = \sum_{i=1}^N I_i = \sum_{i=1}^N \int_{(i-1)\Delta}^{i\Delta} r'(t')dt, \text{ where } \Delta=t/N$$

表 3 随机积分 $I_r = \int_0^t r'(t')dt' = \sum_{i=1}^N I_i = \sum_{i=1}^N \int_{(i-1)\Delta}^{i\Delta} r'(t')dt$ 的分布近似,其中, $\Delta=t/N$

Cases	$t \ll \tau$	$t \gg \tau$	Others
Distribution f_I	f_r	Gaussian	Gaussian
Mean	0	0	0
Variance σ_I^2	$\sigma_r^2 t^2$	$2\sigma_r^2 t$	$2N\sigma_r^2 \left[\int_0^\Delta (\Delta-t')\rho(t')dt' + \sum_{i=2}^N \int_0^\Delta \int_{(i-1)\Delta}^{i\Delta} \rho(t'-t'')dt'dt'' \right]$
$\frac{d\sigma_I}{dt}(t;T)$	σ_r	$\frac{1}{\sqrt{2(T+t)}}$	$\frac{1}{\sigma_I} \left\{ \int_0^\Delta \rho(t')dt' + \sum_{i=2}^N \left[\int_0^\Delta i\rho(i\Delta-t'')dt'' - \int_0^\Delta (i-1)\rho[(i-1)\Delta-t'']dt'' + \int_{(i-1)\Delta}^{i\Delta} \rho(t'-\Delta)dt' \right] \right\}$

当 $t \ll \tau$ 时,随机积分可以近似为一个常数,即 $I_r \approx r't$. 它的分布类型就是积分部分 f_r .

当 $t \gg \tau$ 时,随机积分可以是近似为不相关的随机波动的和,其中, $\rho(r''-t') = \delta(r''-t')$, 且不相关的随机波动具有相同的分布.

对于所有其他情况,我们可以使用文献[39]中的方法和将积分区间 $[0, t]$ 分成 N 个区间,使得每个子区间的长度为 $\Delta=t/N$.

$$I_r = \int_0^t r'(t')dt' = \sum_{i=1}^N I_i = \sum_{i=1}^N \int_{(i-1)\Delta}^{i\Delta} r'(t')dt \tag{18}$$

已知环境噪音 $r'(t)$ 是一个有连续样本函数的固定过程,所有的随机项 $(i=1, \dots, N)$ 具有相同的均值和方差.

$$\sigma_i^2 = \sigma_r^2 \int_{(i-1)\Delta}^{i\Delta} \int_{(i-1)\Delta}^{i\Delta} \rho(t'-t'')dt'dt'' = 2\sigma_r^2 \int_0^\Delta (\Delta-t')\rho(t')dt' \tag{19}$$

我们可以计算区间之间的两点协方差:

$$Cov(I_1, I_i) = \sigma_r^2 \int_0^\Delta \int_{(i-1)\Delta}^{i\Delta} \rho(t'-t'')dt'dt'', \quad i \geq 2 \tag{20}$$

依据相关性随机参数的中心极限定理^[40,41], $I_r(t) = \sum_{i=1}^N I_i$ 是 $(N \rightarrow \infty)$ 接近均值为 0 和方差为 NV 的高斯型, 其中,

$$V = \sigma_i^2 + 2\sum_{i=2}^N Cov(I_1, I_i) < \infty \tag{21}$$

A2 数据集

用于模型验证的 4 个数据集的数值分别列于表 4~表 7 中.

Table 4 The failure data of Web-based accounting ERP system (DS1) from August 2003 to July 2008

表 4 基于 Web 的会计 ERP 系统(DS1)从 2003 年 8 月~2008 年 7 月的失效数据

Time unit (month)	Detected faults	Time unit (month)	Detected faults	Time unit (month)	Detected faults	Time unit (month)	Detected faults
1	1	16	30	31	53	46	84
2	7	17	45	32	54	47	84
3	7	18	47	33	56	48	84
4	9	19	49	34	58	49	85
5	9	20	51	35	59	50	86
6	9	21	53	36	60	51	89
7	12	22	53	37	63	52	90
8	18	23	53	38	70	53	90
9	18	24	53	39	71	54	92

Table 4 The failure data of Web-based accounting ERP system (DS1) from August 2003 to July 2008 (Continued)

表 4 基于 Web 的会计 ERP 系统(DS1)从 2003 年 8 月~2008 年 7 月的失效数据(续)

Time unit (month)	Detected faults	Time unit (month)	Detected faults	Time unit (month)	Detected faults	Time unit (month)	Detected faults
10	18	25	53	40	71	55	108
11	18	26	53	41	72	56	120
12	21	27	53	42	74	57	128
13	22	28	53	43	74	58	129
14	22	29	53	44	80	59	139
15	27	30	53	45	84	60	146

Table 5 The failure data of open source project management software (DS2) from August 2007 to July 2008

表 5 开源项目管理软件(DS2)从 2007 年 8 月~2008 年 7 月的失效数据

Time unit (week)	Detected faults	Time unit (week)	Detected faults	Time unit (week)	Detected faults	Time unit (week)	Detected faults
1	9	14	47	27	70	40	85
2	15	15	49	28	73	41	85
3	19	16	52	29	74	42	87
4	24	17	52	30	75	43	87
5	28	18	55	31	75	44	87
6	29	19	57	32	78	45	89
7	32	20	58	33	79	46	89
8	36	21	61	34	80	47	91
9	36	22	61	35	82	48	91
10	40	23	64	36	83	49	94
11	41	24	66	37	83	-	-
12	41	25	67	38	84	-	-
13	45	26	69	39	84	-	-

Table 6 The failure data of Tandem software (DS3) from the first release

表 6 Tandem 软件(DS3)第 1 版发布的失效数据

Time unit (week)	Detected faults	Time unit (week)	Detected faults	Time unit (week)	Detected faults	Time unit (week)	Detected faults
1	16	6	49	11	81	16	98
2	24	7	54	12	86	17	99
3	27	8	58	13	90	18	100
4	33	9	69	14	93	19	100
5	41	10	75	15	96	20	100

Table 7 The failure data of Tandem software (DS4) from the fourth release

表 7 Tandem 软件(DS4)第 4 版发布的失效数据

Time unit (week)	Detected faults	Time unit (week)	Detected faults	Time unit (week)	Detected faults	Time unit (week)	Detected faults
1	1	6	16	11	32	16	39
2	3	7	19	12	32	17	41
3	8	8	25	13	36	18	42
4	9	9	27	14	38	19	42
5	11	10	29	15	39	-	-



许家俊(1985 -),男,湖北潜江人,博士,主要研究领域为软件可靠性工程,服务计算,Petri 网理论与应用,不确定性量化。



姚淑珍(1965 -),女,博士,教授,博士生导师,主要研究领域为软件工程,形式化方法,Petri 网理论与应用,网络信息安全。