

考虑软件不同失效过程偏差的软件可靠性模型

楼俊钢 江建慧 靳 昂

(同济大学计算机科学与技术系 上海 201804)

摘 要 软件可靠性分析是根据软件失效数据等信息,通过合理建模来对软件可靠性进行预计和评价.现有的基于随机过程的可靠性模型一般采用均值过程来描述软件失效数据,然而,软件失效数据的模型化实质上应该是使其成为某个随机过程的一个样本轨迹.文中建立了考虑软件不同失效过程偏差的软件可靠性模型,用 NHPP 过程表示失效过程均值函数的变化趋势,ARMA 过程表示实际失效过程对均值过程的偏差序列.在两组公开发表的真实数据集上对模型的实验表明,新模型较之一些广泛使用的 NHPP 软件可靠性模型在拟合能力及适用性上有明显的提高,并且保持了较好的预测能力.

关键词 软件可靠性模型;样本轨迹;非齐次泊松过程模型;随机过程;自回归滑动平均过程

中图法分类号 TP302 **DOI号**: 10.3724/SP.J.1016.2010.01263

A New Software Reliability Model Considering Warps Between Different Software Failure Processes

LOU Jun-Gang JIANG Jian-Hui JIN Ang

(Department of Computer Science and Technology, Tongji University, Shanghai 201804)

Abstract The software reliability model is one of the important approaches to predict and evaluate software reliability quantitatively. The software failure data to be analyzed should be considered as a particular realization of a stochastic process. In this paper, a new software reliability growth model considering warps between different software failure processes is proposed. The experimental results based on two real data sets show that the proposed model has better prediction and curve fitting abilities than that of some other conditional NHPP software reliability growth models. The measures used for comparison are mean square of fitting error, predicted error and relative error.

Keywords software reliability model; sample realization; nonhomogeneous poisson process (NHPP); stochastic process; auto regressive moving average process

1 引 言

随着软件规模的不断扩大、结构和功能的日益复杂,软件质量越来越受到人们的重视.软件可靠性

是衡量软件质量最重要的指标之一,它指的是在规定条件下,在规定时间内,软件不发生失效的概率.软件可靠性模型是软件可靠性研究中倍受关注、非常活跃的一个领域.从 Hudson 的工作开始,到 1971 年 Jelinski- Moranda (J-M) 模型的发表以来,

收稿日期:2008-12-15;最终修改稿收到日期:2010-06-25. 本课题得到国家“八六三”高技术研究发展计划项目基金(2007AA01Z142)、上海市科学技术委员会信息技术领域重点科技攻关计划项目基金(04DZ15032,06DZ15003)资助.楼俊钢,男,1982年生,博士研究生,主要研究方向为软件测试、软件可靠性评估、性能评估等. E-mail: loujungang0210@hotmail.com. 江建慧,男,1964年生,博士,教授,博士生导师,主要研究领域为容错计算、软件可靠性工程、微处理器体系结构、计算机系统性能评估.靳昂,男,1982年生,博士研究生,主要研究方向为故障注入、软件可靠性评估.

目前已公开发表了 200 种以上的模型^[1-2]. 它们不仅是对软件可靠性进行预计、分配、分析与评价的强有力工具, 更为软件可靠性设计提供了指南. 软件可靠性模型的研究旨在根据软件可靠性数据, 以统计方法等给出软件的可靠性估计值或预测值.

随机过程模型^[3-5]是软件可靠性模型中最重要的一类, 也是在当前软件可靠性模型领域研究最多、应用最广泛的一类. 这类模型结构简单, 在很多应用中都有良好的表现. 随机过程可以描述软件在时间 t 以前发生的失效数随时间变化的随机行为, 或者描述软件发生失效时间的随机性. 随机过程模型的主要优点在于其模型的简单性, 易于理解和实施. 按照到时刻 t 的发生失效次数服从的分布, 主要分为泊松过程模型和二项式模型等. 其中, 非齐次泊松过程 (Nonhomogeneous Poisson Process, NHPP) 模型是应用最广泛的随机过程模型^[6-14], 它用一个计数过程 $\{N_t, t \in T\}$ 表示到时刻 t 为止检测到的软件失效累计数. 如果失效累计数的均值函数用 $m(t)$ 表示, 则一个基于 NHPP 过程的软件可靠性模型的一般形式为

$$P\{m(t_{i+1}) - m(t_i) = n\} = \frac{e^{-[m(t_{i+1}) - m(t_i)]} [m(t_{i+1}) - m(t_i)]^n}{n!}.$$

随机过程软件可靠性模型赖以生存的根本是程序错误行为的不确定性, 由于无法事先知道程序对于每一个可能输入的行为表现以及难以准确地预计程序将来的输入情况, 软件故障过程确实是随机的. 传统的基于随机过程的软件可靠性模型总是假定软件失效过程应该用不同的衰减曲线描述, 比如指数衰减型、S 型、威布尔型、倒线性型、几何衰减型^[1-2]等. 这些变化规律是从传统可靠性理论借鉴过来的, 一件实体产品, 譬如汽车、机床、晶体管等, 受温度、湿度、磨损及老化的作用, 它的故障强度曲线会按这些规律变化. 然而计算机软件产品有其自身的特点, 其失效行为比起实体产品来要复杂得多. 首先, 软件失效是动态的, 它一定要与人、计算机等交互作用; 其次, 软件失效的影响因素更多, 如测试工具、测试人员的思路、风格及经验、不完全排错、计算机环境因素、支撑系统 (如操作系统、编译程序、数据库等), 计算机的载荷、数据精度、数据记录误差等. 软件失效行为的特殊性和复杂性使得使用单一的软件失效过程衰减曲线不可能达到很高的精度, 而只能碰运气, 这些模型只是偶尔非常精确, 换一个环境, 或者应用到别的工程项目时, 其精度就不能满足要求了.

如当某一软件工程项目的失效过程碰巧是指数型的或是几何衰减型时, 它们的精度显得很好, 一旦换一个项目, 当它的失效过程不按这些规律变化时, 这些模型的精度就很糟糕^[12].

造成传统随机过程模型精度不高、适应性不好的原因主要有以下两点: (1) 研究人员永远无法预先得到关于程序的全部信息, 因此, 可靠性问题中的统计成分往往不能仅用单一的统计分布函数来描述, 并且随着软件项目的变化, 随机过程模型对软件故障的属性和软件失效过程所做出的许多先验假设不会永远准确; (2) 软件可靠性失效过程一般来说具有不可复制性, 即使是同一软件系统, 在不同的测试条件、使用环境下也会出现不同的失效过程, 而可靠性建模时所采用的软件可靠性失效时间或者失效间隔数据实际上仅仅是软件失效过程的一个样本轨迹. 这些问题的存在使得随机过程可靠性模型在描述不同程序的失效过程时会产生一定的偏差, 而传统的随机过程可靠性模型恰恰忽略了这些偏差.

为解决模型适用性差的问题, 研究人员将他们的精力主要集中在以下几个方面: (1) 模型选择. 每一个模型都有它自己的特性, 每一个软件工程项目也有它自己的特性, 问题是如何选择一个模型使之适合某个具体的软件工程项目. (2) 模型组合. 既然每一个模型都有它自己的特点, 那就有可能在某个具体软件工程项目上应用多个模型, 让它们互相补充, 综合分析它们的计算结果, 以形成混合模型. (3) 采用新的数学方法建模, 如相关向量机^[14]、贝叶斯统计学^[15-17]、神经网络^[18-20]、未确知理论^[21]、支持向量机^[22]、混沌理论^[23]等.

本文没有遵循上述思路, 而是另辟蹊径, 通过数据预处理的方法来弥补传统可靠性模型与实际软件失效过程存在的偏差, 作者建议在经典随机过程模型的基础上, 辅助一个补偿随机过程, 用两个互补的随机过程来为软件故障行为建模, 新模型由两部分组成: $Y_i = M_i + X_i$, 其中, M_i 反映了失效过程均值函数的变化趋势, X_i 为一时间序列, 表示实际失效过程对均值过程的偏差序列. 由随机函数的分解以及复合叠加定理^[12], 如果给定随机函数 $\{U(t)\}$ 和 $\{N(t)\}$, 则总存在 $\{N(t)\}$ 的一个补函数 $\{X(t)\}$, 使得 $\{U(t)\}$ 可以分解成 $\{N(t)\}$ 与 $\{X(t)\}$ 的复合叠加的形式. 因此从理论上来说, 将软件的动态失效行为分解成两个随机函数复合叠加是可行的; 在两组公开发表的数据集上的实验结果表明, 新模型较之一些广泛使用的 NHPP 模型在拟和能力及适用性上

有明显的提高, 并且保持了较好的预测能力.

2 相关概念

定义 1. 定义在 T 上的函数 $\{X_t(\omega), \omega \in \Omega\}$ 称为随机过程 $\{X_t, t \in T\}$ 的样本轨迹 (也可称为现实). $X_t(\omega)$ 又称为时间序列.

时间序列既可以表示随机过程本身, 也可以表示随机过程的一个样本轨迹, 本文所指的时间序列表示后者.

定义 2. 如果时间序列 $\{X_t, t \in T\}$ 对每个 $t \in T$, 有 $\text{Var}(X_t) < \infty$, 则 $\{X_t\}$ 的自协方差函数 $\gamma_X(\cdot, \cdot)$ 定义为 $\gamma_X(r, s) = E[(X_r - EX_r)(X_s - EX_s)]$, $r, s \in T$.

定义 3. 时间序列 $\{X_t, t \in T\}$ 为平稳时间序列, 如果有

- (i) $E|X_t|^2 < \infty, \forall t \in T$,
- (ii) $EX_t = EX_s, \forall t, s \in T$,
- (iii) $\gamma_X(r, s) = \gamma_X(r+t, s+t), \forall r, s, t \in T$.

定义 4. 如果过程 $\{Z_t\}$ 的均值为零、自协方差函数为 $\gamma(h) = \begin{cases} \sigma^2, & h=0 \\ 0, & h \neq 0 \end{cases}$, 则称过程 $\{Z_t\}$ 是零均值、方差为 σ^2 的白噪声, 记为 $\{Z_t\} \sim \text{WN}(0, \sigma^2)$.

定义 5. 过程 $\{X_t, t \in T\}$ 称为自回归滑动平均过程 (Auto Regressive Moving Average Process, ARMA), 表示为 $\text{ARMA}(p, q)$, 如果 $\{X_t\}$ 是平稳过程, 且对任意 t , 有

$$X_t - \phi_1 X_{t-1} - \dots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \dots + \theta_q Z_{t-q}.$$

上式可简记为 $\phi(B)X_t = \theta(B)Z_t, t \in T$, 其中, $\phi(\cdot)$ 和 $\theta(\cdot)$ 分别称为 p 次自回归多项式和 q 次活动平均多项式:

$$\begin{aligned} \phi(z) &= 1 - \phi_1 z - \dots - \phi_p z^p, \\ \theta(z) &= 1 + \theta_1 z + \dots + \theta_q z^q, \end{aligned}$$

B 定义为 $B^j X_t = X_{t-j}, j=0, 1, 2, \dots$.

定义 6. 如果 $\theta(z) \equiv 1$, 则称 $X_t = \theta(B)Z_t$ 为 p 阶自回归过程, 记作 $\text{AR}(p)$.

定义 7. 如果 $\phi(z) \equiv 1$, 则称 $\phi(B)X_t = Z_t$ 为 q 阶滑动平均过程, 记作 $\text{MA}(q)$.

3 模型的实现

虽然经典的随机过程模型不能完全刻画软件的失效行为, 但它有着雄厚的理论及实践根基, 有很多

合理成分还不能被其他理论所替代, 例如, 从随机过程软件可靠性模型中能够导出软件可靠度以外的许多其他度量值. 迄今为止, NHPP 类软件可靠性增长模型是目前应用最广、影响最大的一类模型, 这类模型依然是拟合效果最好、结构和应用最简单的随机过程模型, 并且已经成为软件可靠性工程实践中很成功的工具. 目前软件可靠性模型研究当中很重要的一部分工作是对 NHPP 类软件可靠性模型的完善. 因此, 在本文所提出的模型中, M_t 采用了均值函数为 $m(t)$ 的 NHPP 过程.

假设软件累计失效数的总体分布为服从均值函数 $m(t)$ 的 NHPP 过程, 序列 $\{m_1, m_2, \dots, m_n\}$ 和 $\{y_1, y_2, \dots, y_n\}$ 分别表示时间点 $\{t_1, t_2, \dots, t_n\}$ 的失效期望数和实际累计失效数. 那么, $\{y_1, y_2, \dots, y_n\}$ 可以看作总体分布的一个样本序列, $\{y_1 - m_1, y_2 - m_2, \dots, y_n - m_n\}$ 为时间 $\{t_1, t_2, \dots, t_n\}$ 的一个偏差时间序列, 即为新模型的 X_t 部分. 由中心极限定理可知, 样本均值的数学期望总是等于总体的均值, 因此, 当 n 足够大时, 序列 $\{y_1, y_2, \dots, y_n\}$ 均值的数学期望与序列 $\{m_1, m_2, \dots, m_n\}$ 的均值相等, $\{y_1 - m_1, y_2 - m_2, \dots, y_n - m_n\}$ 的值围绕着零上下波动, 这种波动的上下部分之和渐渐趋于相等. 在数据的平稳性检查通过后, X_t 可以看作均值为零的平稳时间序列, 使用 $\text{ARMA}(p, q)$ 过程可以对偏差过程 Y_t 有更加精确的估计.

用平稳过程模型来描述上述偏差序列的好处是可以展示在一定时期内的变动过程, 从中寻找和分析偏差的变化特征、发展趋势和规律, 反映偏差这一变量受其它各种因素影响的总结果, 并通过处理时间序列数据获得它随时间的演变特性与规律.

模型实现的步骤如下.

1. 根据软件累计失效数据, 用极大似然法求出均值函数 $m(t)$ 的参数估计值, 求得 $m(t)$ 的估计函数 $\hat{m}(t)$, 计算 $\{t_1, t_2, \dots, t_n\}$ 时的期望累计失效数估计值 $\{\hat{m}_1, \hat{m}_2, \dots, \hat{m}_n\}$;
2. 用实际累计失效序列 $\{y_1, y_2, \dots, y_n\}$ 减去 $\{\hat{m}_1, \hat{m}_2, \dots, \hat{m}_n\}$ 作为 X_t 在 $\{t_1, t_2, \dots, t_n\}$ 时的观测值 $\{x_1, x_2, \dots, x_n\}$;
3. 对 $\{x_1, x_2, \dots, x_n\}$ 进行平稳性检验, 选择 $\text{ARMA}(p, q)$ 过程的阶 p 和 q , 并估计模型的参数值;
4. 由步 1 和步 3, 可以得到接下来时间软件失效期望次数估计值 \hat{m}_{n+m} 及偏差估计值 \hat{x}_{n+m} , 则接下来时间软件实际失效次数估计值为 $\hat{y}_{n+m} = \hat{m}_{n+m} + \hat{x}_{n+m}$.

3.1 用 ARMA 模型进行预测

预测即依靠时间 $\{t_1, t_2, \dots, t_n\}$ 的观测值 $\{x_1, x_2, \dots, x_n\}$ 来预测 t_{n+m} ($m=1, 2, 3, \dots$) 时的软件累

计失效次数估计值 \hat{x}_{t+m} .

预测算法.

设 $\{X_t, t \in T\}$ 是均值为零、自协方差函数为 $\kappa(i, j) = E(X_i X_j)$ 的随机过程, 如果对一切 $n = 1, 2, \dots$, 协方差矩阵 $[\kappa(i, j)]_{i, j=1}^n$ 是非奇异阵, 令 $v_n = (X_{n-1} - \hat{X}_{n+1})^2$, 则 $\hat{X}_{n+1} (n \geq 1)$ 可用下式递归计算:

$$\hat{X}_{n+1} = \begin{cases} 0, & n=0 \\ \sum_{j=1}^n \theta_{nj} (X_{n+1-j} - \hat{X}_{n+1-j}), & n \geq 1 \end{cases} \quad (1)$$

$$\begin{cases} v_0 = \kappa(1, 1) \\ \theta_{n, n-k} = v_k^{-1} (\kappa(n+1, k+1) - \sum_{j=1}^n \theta_{k, k-j} \theta_{n, n-j} v_j) \\ v_n = \kappa(n+1, n+1) - \sum_{j=1}^n \theta_{n, n-j}^2 v_j \end{cases} \quad (2)$$

其中, $k=0, 1, \dots, n-1$.

递归顺序为 $v_0, \theta_{11}, v_1, \theta_{22}, \theta_{21}, v_2, \theta_{33}, \theta_{31}, v_3, \dots$.

投影定理^[24]. 如果 ℓ 是 Hilbert 空间 \mathcal{h} 的一个闭线性子空间, 且 $x \in \mathcal{h}$, 那么

- (1) 存在唯一元素 $\hat{x} \in \ell$, 使得 $\|x - \hat{x}\| = \inf_{y \in \ell} \|x - y\|$.
- (2) $\hat{x} \in \ell$ 且 $\|x - \hat{x}\| = \inf_{y \in \ell} \|x - y\|$ 成立的充分必要条件是 $\hat{x} \in \ell$ 且 $\hat{x} - x \in \ell^\perp$ (称 \hat{x} 为 x 在 ℓ 上的正交投影).

命题 1(投影映射的性质)^[24]. 设 \mathcal{h} 是 Hilbert 空间, P_ℓ 是到闭线性子空间 ℓ 的投影映射, 那么

- (1) $\forall x, y \in \mathcal{h}, \alpha, \beta \in C, P_\ell(\alpha x + \beta y) = \alpha P_\ell x + \beta P_\ell y$;
- (2) $\|x\|^2 = \|P_\ell x\|^2 + \|(1 - P_\ell)x\|^2$;
- (3) $\forall x \in \mathcal{h}$, 存在唯一的正交分解 $x = P_\ell x + (1 - P_\ell)x$;
- (4) 如果当 $n \rightarrow \infty$ 时, $\|x_n - x\| \rightarrow 0$, 则当 $n \rightarrow \infty$ 时, $P_\ell x_n \rightarrow P_\ell x$;
- (5) $x \in \ell \Leftrightarrow P_\ell x = x$;
- (6) $x \in \ell^\perp \Leftrightarrow P_\ell x = 0$;
- (7) $\ell_1 \subset \ell_2 \Leftrightarrow \forall x \in \mathcal{h}, P_{\ell_1} P_{\ell_2} x = P_{\ell_1} x$.

预测算法证明. 由 \hat{X}_j 的定义知, 当 $i < j$ 时, 有

$$X_i - \hat{X}_i \in \mathcal{h}_{j-1}, X_j - \hat{X}_j \in \mathcal{h}_{j-1}^\perp,$$

故

$$\{X_1 - \hat{X}_1, X_2 - \hat{X}_2, \dots, X_n - \hat{X}_n\}$$

是正交集.

设 $0 \leq k \leq n$, 以 $X_{k+1} - \hat{X}_{k+1}$ 两端取内积得

$$\begin{aligned} \langle \hat{X}_{n+1}, X_{k+1} - \hat{X}_{k+1} \rangle &= \\ \langle \sum_{j=0}^{n-1} \theta_{n, n-j} (X_{j+1} - \hat{X}_{j+1}), X_{k+1} - \hat{X}_{k+1} \rangle &= \theta_{n, n-k} v_k \end{aligned} \quad (3)$$

注意到

$$(\hat{X}_{n+1} - \hat{X}_{n+1}) \perp (X_{k+1} - \hat{X}_{k+1}), 0 \leq k \leq n,$$

故有

$$\theta_{n, n-k} = v_k^{-1} \langle \hat{X}_{n+1}, X_{k+1} - \hat{X}_{k+1} \rangle.$$

在式(1)中令 $k=n$, 有

$$\begin{aligned} \theta_{n, n-k} &= \\ v_k^{-1} (\kappa(n+1, k+1) - \sum_{j=0}^{k-1} \theta_{k, k-j} \langle X_{n+1}, X_{j+1} - \hat{X}_{j+1} \rangle) & \end{aligned} \quad (4)$$

又由式(3)以及下式

$$\langle X_{n+1}, X_{j+1} - \hat{X}_{j+1} \rangle = v_j \theta_{n, n-j}, 0 \leq j < n,$$

式(4)可以表示为

$$\theta_{n, n-k} = v_k^{-1} (\kappa(n+1, k+1) - \sum_{j=0}^{k-1} \theta_{k, k-j} \theta_{k, k-j} \theta_{n, n-j} v_j),$$

由投影定理和命题 1 可得

$$\begin{aligned} v_n &= \|X_{n+1} - \hat{X}_{n+1}\|^2 + \|X_{n+1}\|^2 - \|\hat{X}_{n+1}\|^2 \\ &= \kappa(n+1, k+1) - \sum_{j=0}^{k-1} \theta_{n, n-k}^2 v_k. \end{aligned}$$

从而式(2)得证.

证毕.

3.2 ARMA 模型的参数估计

ARMA(p, q) 模型参数的估计指根据计算得到的偏差数据观测值序列 $\{x_1, x_2, \dots, x_n\}$ 和选定的阶 p 和 q 的值, 对均值、自回归系数和滑动平均系数 $\{\varphi_i, \theta_j\} (i=1, 2, \dots, p, j=1, 2, \dots, q)$ 及白噪声方差 σ^2 的估计.

观测值序列 $\{x_1, x_2, \dots, x_n\}$ 的高斯似然函数为

$$L(\phi, \theta, \sigma^2) = (2\pi\sigma)^{-n/2} (r_0, \dots, r_{n-1})^{-1/2} \cdot$$

$$\exp[-1/2\sigma^2 \sum_{j=1}^n (X_j - \hat{X}_j)^2 / r_{j-1}],$$

其中,

$$r_j = E(X_{i+1} + \hat{X}_{i+1})^2 / \sigma^2 \quad (5)$$

求 $L(\phi, \theta, \sigma^2)$ 对 σ^2 的偏导数, 可得极大似然估计 $\hat{\phi}, \hat{\theta}, \hat{\sigma}^2$, 满足 $\hat{\sigma}^2 = n^{-1} \sum_{j=1}^n (X_j - \hat{X}_j)^2 / r_{j-1}$, 而 $\hat{\phi}, \hat{\theta}$ 是最小化的 ϕ 和 θ 的估计:

$$l(\phi, \theta) = \ln \left[n^{-1} \sum_{j=1}^n (X_j - \hat{X}_j)^2 + n^{-1} \sum_{j=1}^n \ln r_{j-1} \right].$$

对任意给定的一对参数向量, 可利用预测算法递归计算 $\theta_{i-1}, r_{i-1}, \hat{X}_i$, 使 $l(\phi, \theta)$ 最小的 ϕ, θ 的值即

是 ϕ, θ 的极大似然估计, 再根据式(5)计算 σ^2 的极大似然估计.

4 实验比较

为检验本文所提出的建模思路的有效性, 并验证所建立软件可靠性模型的拟合能力及预测能力, 我们使用均值函数分别为 $at^2 + bt + c$ 和 $a(1 - e^{-bt})$ 的

NHPP 过程(分别记为 M1、M2)与 ARMA(2, 2)、ARMA(4, 0)、ARMA(0, 4)过程两两组合的新模型, 并与 G-O model^[1-2]、Yamada Delayed S-Shaped^[25]、SRGM With Logistic TEF^[26]、SRGM With Rayleigh TEF^[27]、Delayed S-Shaped Model With Logistic TEF^[28]、Delayed S-Shaped Model With Rayleigh TEF^[28] 等 6 个使用广泛的模型进行了拟合能力、预测能力的比较. 用于比较模型的均值函数见表 1.

表 1 用于实验比较的 NHPP 模型及其均值函数

模型	均值函数
SRGM With Logistic TEF	$a(1 - e^{-bt})$
SRGM With Rayleigh TEF	$a(1 - (1 + bt)e^{-bt})$
Delayed S-Shaped Model With Logistic TEF	$a(1 - e^{-b(\frac{N}{1+Ae^{-at}} - \frac{N}{1+A})})$
Delayed S-Shaped Model With Rayleigh TEF	$a(1 - e^{-b(N \times (1 - e^{-ct^m}))})$
G-O model	$a(1 - (1 + e^{b(\frac{N}{1+Ae^{-at}} - \frac{N}{1+A})})e^{-b(\frac{N}{1+Ae^{-at}} - \frac{N}{1+A})})$
Yamada Delayed S-Shaped	$a(1 - (1 + b(N \times (1 - e^{-ct^m})))e^{-b(N \times (1 - e^{-ct^m}))})$

4.1 数据集及比较标准

4.1.1 数据集

评估模型使用的数据集见表 2 和表 3, 它们是从实际工程项目中收集来的, 经过分析和整理, 精度高, 专门为比较、验证和评价软件可靠性模型而设计.

表 2 数据集 1^[29]

时间/week	软件累计失效数	时间/week	软件累计失效数
1	15	11	233
2	44	12	255
3	66	13	276
4	103	14	298
5	105	15	304
6	110	16	311
7	146	17	320
8	175	18	325
9	179	19	328
10	206		

表 3 数据集 2^[30]

时间/week	软件累计失效数	时间/week	软件累计失效数
1	16	11	81
2	24	12	86
3	27	13	90
4	33	14	93
5	41	15	96
6	49	16	98
7	54	17	99
8	58	18	100
9	69	19	100
10	75	20	100

我们使用的数据集 1 来自于 Ohba 的论文^[29], 收集于一个 PL/L 数据库应用系统, 代码行数大约为

1317000LOC. 测试时间为 19 周, 共发生了 328 次失效. 虽然数据集比较陈旧, 但便于将我们的工作与以往软件可靠性模型研究人员的工作进行直接比较. 数据集 2 来自于 Wood^[30] 收集于 Tandem 公司 4 个独立软件版本的第 1 个, 代码行数约为 1317000LOC. 测试时间为 20 周, 共发生了 100 次失效.

4.1.2 比较标准

(1) 曲线拟合能力

模型的拟合能力使用误差平方和 (Mean Square of fitting Error, MSE)^[28] 及预测误差 (Predicted Error, PE)^[31] 度量:

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}$$

$$PE_i = Actual_i - predicted_i,$$

其中, n 表示失效数据集中失效样本的数量, \hat{y}_i 表示到 t_i 时刻为止故障累计数的估算值, y_i 表示到 t_i 时刻为止故障累计数的实测值. MSE 的值越小, 曲线拟合得越好, PE_i 的值越小, 曲线拟合得也越好.

(2) 模型预测能力

采用相对误差 (Relative Error, RE) 来评价模型预测能力^[1], 定义为 $RE = \frac{m(t_q) - q}{q}$.

假设在软件测试结束时观测到 q 个软件失效, 使用失效数据 $t_a (t_a \leq t_n)$ 估测软件可靠性模型均值函数的参数, 然后将这些参数值代入到模型中, 可以得到软件测试结束时间 t_n 的累积失效数的估计值,

得到的估计值与实际值进行比较. *RE* 值越小, 说明预测能力越好.

4.2 偏差数据的平稳性检验

应用 ARMA 模型, 首先需要对数据的平稳性进行检查. 数据序列平稳的特征为: 样本序列的自相关函数在某一固定水平线附近摆动, 且按周期性逐渐衰减, 所以这些时间序列基本上都是平稳的, 可以用平稳过程模型模拟.

析图来判别数据序列的平稳性.

图 1(a)~(d) 为数据集 1 和 2 上采用 M1 和 M2 模型的估计值与实际值的差值得到的 4 个样本序列. 从图中可以看出, 样本序列数据的自相关系数均在某一固定水平线附近摆动, 且按周期性逐渐衰减, 所以这些时间序列基本上都是平稳的, 可以用平稳过程模型模拟.

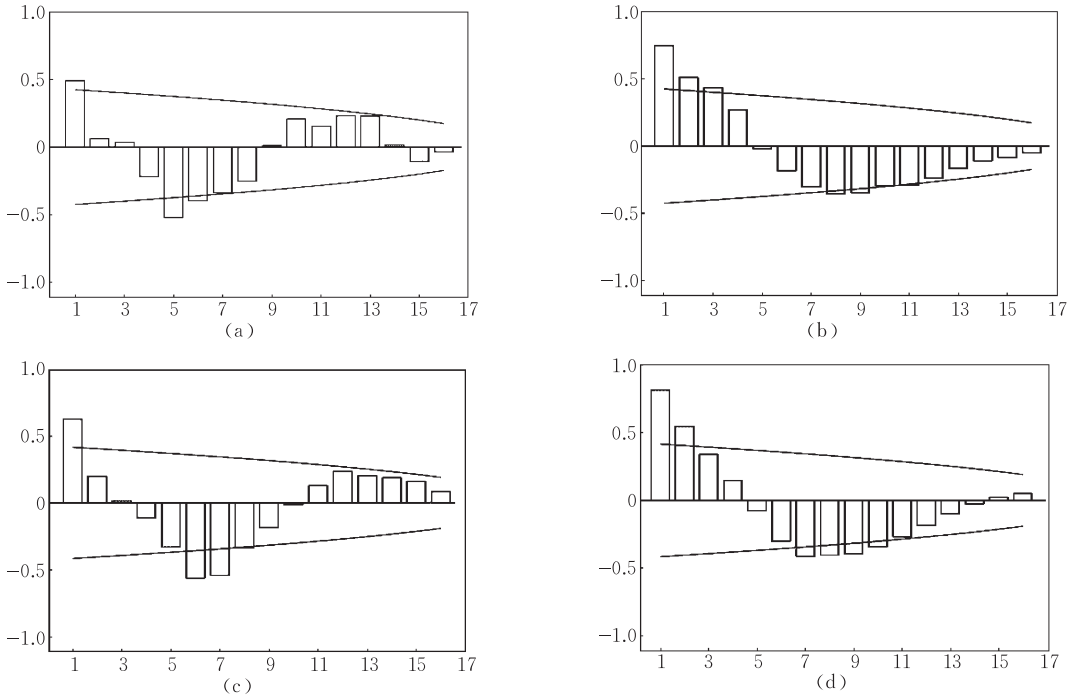


图 1 4 个偏差样本序列上的自相关分析图

4.3 模型性能分析

4.3.1 数据集 1

表 4 为新模型在数据集 1 和 2 上 ARMA 部分的参数估计值, 表 5 列出了模型在 NHPP 部分的极大似然估计值, 包括新模型和用于比较的 6 个

NHPP 模型. 此外, 表 5 还列出了各模型应用于数据集 1 上所得的 *MSE* 值和 *RE* 值. Logistic TEF 和 Rayleigh TEF 的另外几个参数的极大似然估计值分别为 $N = 54.84$, $A = 13.03$, $\alpha = 0.226$, $N = 49.32$, $c = 0.0137$.

表 4 ARMA 的参数估计

模型	数据集 1	数据集 2
M1 + ARMA(2,2)	$\phi_1 = 1.00, \phi_2 = 0.202, \theta_1 = -0.475, \theta_2 = 0.472$	$\phi_1 = 0.439, \phi_2 = -0.095, \theta_1 = -0.537, \theta_2 = -0.214$
M1 + ARMA(4,0)	$\phi_1 = 0.800, \phi_2 = -0.574, \theta_1 = 0.532, \theta_2 = -0.527$	$\phi_1 = -0.946, \phi_2 = 0.373, \theta_1 = 0.196, \theta_2 = -0.272$
M1 + ARMA(0,4)	$\theta_1 = -1.09, \theta_2 = -0.264, \theta_3 = -0.179, \theta_4 = -0.489$	$\theta_1 = -1.01, \theta_2 = -0.506, \theta_3 = -0.043, \theta_4 = 0.121$
M2 + ARMA(2,2)	$\phi_1 = 0.471, \phi_2 = 0.180, \theta_1 = -0.452, \theta_2 = 0.118$	$\phi_1 = 1.781, \phi_2 = -0.913, \theta_1 = 0.650, \theta_2 = 0.287$
M2 + ARMA(4,0)	$\phi_1 = 0.986, \phi_2 = -0.464, \theta_1 = 0.585, \theta_2 = -0.394$	$\phi_1 = 1.206, \phi_2 = -0.397, \theta_1 = 0.237, \theta_2 = 0.341$
M2 + ARMA(0,4)	$\theta_1 = -1.11, \theta_2 = -0.446, \theta_3 = -0.410, \theta_4 = -0.581$	$\theta_1 = -1.34, \theta_2 = -1.10, \theta_3 = -0.536, \theta_4 = -0.09$

表 5 数据集 1 上模型的比较

模型	参数估计值	MSE	RE
SRGM With Logistic TEF	$a = 395.57, b = 4.16 \times 10^{-2}$	114.06	2.14×10^{-3}
SRGM With Rayleigh TEF	$a = 459.09, b = 2.73 \times 10^{-2}$	268.42	-7.77×10^{-3}
Delayed S-Shaped Model With Logistic TEF	$a = 339.96, b = 1.21 \times 10^{-1}$	634.83	7.11×10^{-7}
Delayed S-Shaped Model With Rayleigh TEF	$a = 333.18, b = 1.00 \times 10^{-1}$	798.49	-4.46×10^{-2}
G-O model	$a = 513.15, b = 5.37 \times 10^{-2}$	222.14	-2.90×10^{-5}

(续 表)

模型	参数估计值	MSE	RE
Yamada Delayed S-Shaped	$a=384.05, b=2.19 \times 10^{-1}$	640.67	1.15×10^{-1}
M1+ARMA(2,2)	$a=-0.4474, b=27.2681, c=-14.5666,$	59.11	-2.07×10^{-3}
M1+ARMA(4,0)	$a=-0.4474, b=27.2681, c=-14.5666$	56.18	-1.43×10^{-3}
M1+ARMA(0,4)	$a=-0.4474, b=27.2681, c=-14.5666$	56.59	-1.80×10^{-3}
M2+ARMA(2,2)	$a=513.15, b=-0.0537,$	89.81	-1.38×10^{-3}
M2+ARMA(4,0)	$a=513.15, b=-0.0537,$	77.22	-4.03×10^{-4}
M2+ARMA(0,4)	$a=513.15, b=-0.0537,$	70.34	-1.37×10^{-4}

从表 5 中可以看出,新模型的 MSE 值要好于用于比较的 6 个模型,而 RE 值的数量级为 $10^{-4} \sim 10^{-3}$ 之间,比新模型表现更好的模型有 Delayed S-Shaped Model With Logistic TEF 和 G-O 模型,其它 4 个模型的 RE 值要稍差些。

图 2 描述了各模型在数据集 1 上的 PE 值。从

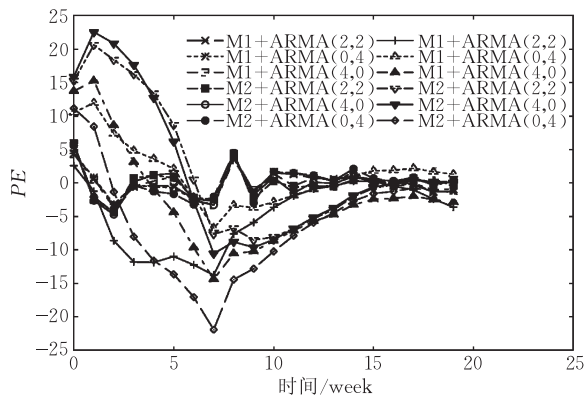


图 2 数据集 1 上模型 PE 值的比较

图中可以看出,我们的模型更加贴近于中间线,这说明新模型的拟合能力要优于其它各模型。

4.3.2 数据集 2

表 6 中列出了模型在 NHPP 部分的极大似然估计值,包括新模型和用于比较的 6 个 NHPP 模型。此外,表 6 还列出了各模型应用于数据集 2 上所得的 MSE 值和 RE 值。Logistic 和 Rayleigh TEF 的另外几个参数的极大似然估计值分别为 $N=10505, A=15.24, \alpha=0.285, N=10534, c=0.0149$ 。

从表 6 中可以看出,新模型在 MSE 值上的表现要好于其它模型,而 RE 值的数量级在 $10^{-4} \sim 10^{-3}$ 之间,排在 Delayed S-Shaped Model With Logistic TEF 和 Delayed S-Shaped Model With Rayleigh TEF 之后,但好于其它 4 个模型。

图 3 描述了各模型在数据集 2 上的 PE 值。从图中可以看出,所提出的新模型更加贴近于中间线,这说明模型的拟合能力要好于其它各模型。

表 6 数据集 2 上模型的比较

模型	参数估计值	MSE	RE
SRGM With Logistic TEF	$a=107.66, b=2.66 \times 10^{-4}$	22.76	-1.27×10^{-2}
SRGM With Rayleigh TEF	$a=110.61, b=2.66 \times 10^{-4}$	39.69	-9.51×10^{-3}
Delayed S-Shaped Model With Logistic TEF	$a=101.86, b=6.35 \times 10^{-4}$	92.66	1.22×10^{-6}
Delayed S-Shaped Model With Rayleigh TEF	$a=102.14, b=5.78 \times 10^{-4}$	107.97	-1.67×10^{-6}
G-O model	$a=112.48, b=1.20 \times 10^{-1}$	30.27	2.20×10^{-2}
Yamada Delayed S-Shaped	$a=102.26, b=3.45 \times 10^{-1}$	94.99	1.49×10^{-2}
M1+ARMA(2,2)	$a=-0.2154, b=9.3205, c=2.4904$	3.23	-1.21×10^{-3}
M1+ARMA(4,0)	$a=-0.2154, b=9.3205, c=2.4904$	3.02	-8.91×10^{-4}
M1+ARMA(0,4)	$a=-0.2154, b=9.3205, c=2.4904$	3.18	1.32×10^{-3}
M2+ARMA(2,2)	$a=112.48, b=-0.120$	3.61	4.33×10^{-4}
M2+ARMA(4,0)	$a=112.48, b=-0.120$	4.13	1.87×10^{-4}
M2+ARMA(0,4)	$a=112.48, b=-0.120$	4.94	-3.30×10^{-4}

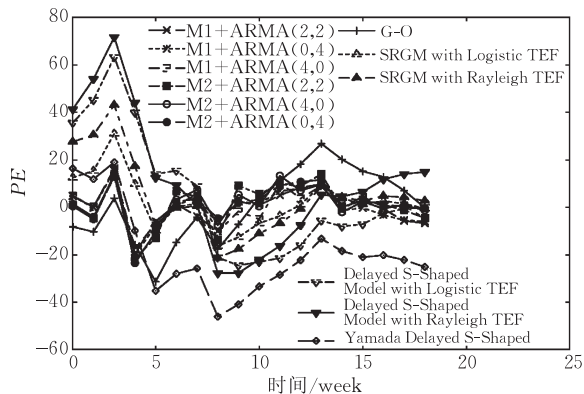


图 3 数据集 2 上模型 PE 值的比较

使用数据集 1 和数据集 2,经过上述比较过程可以得出以下结论:

(1)新模型的拟合能力要好于其它模型,这在数据集 2 上表现得尤为明显,并且新模型对两组不同特性的失效数据都有较好的适用性;

(2)在预测能力上,新模型排名也比较靠前。

总体来说,考虑样本与均值的差别对提高软件可靠性模型的拟合能力和预测能力都是有益的,值得进行更加深入的研究。

5 总 结

本文通过实验说明新模型的拟合能力要好于其它模型,这在数据集 2 上表现得尤为明显,而且新模型对两组不同特性的失效数据都有较好的适用性;从理论上,由于新模型不对软件失效强度变化作任何统计规律的假定,因此,它能改善软件可靠性模型的适应性,提高模型的拟合能力.在许多 NHPP 类模型与其它随机过程可靠性模型的比较中已经说明,NHPP 类模型在拟合效果、结构和应用方面具有优势.因此,我们可以认定本文中所选择的模型有代表意义,新模型与其它的随机过程模型相比,依然可以取得类似的结果.

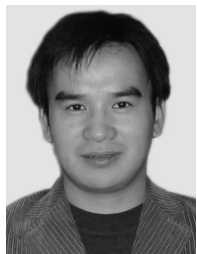
值得说明的是,本文所提出的方法不是为了从根本上解决造成传统随机过程模型精度不高、适应性不强的问题,而是通过数据预处理来弥补传统可靠性模型与实际软件失效过程存在的偏差.在经典随机过程模型的基础上,辅助一个补偿随机过程,用两个互补的随机过程来为软件故障行为建模.首先通过拟合得到了失效期望值,然后用自回归滑动平均过程进行了误差的控制,利用自回归滑动平均过程的调节作用,提高了模型的拟合能力和预测能力.

下一步的工作包括:采用除 NHPP 类之外的随机过程模型在更多数据集上对新模型的适用性进行验证;研究如何更加精确地描述传统可靠性模型与实际软件失效过程存在的偏差,以进一步提高模型的预测能力.

参 考 文 献

- [1] Musa J D. Software Reliability Engineering. New York: McGraw Hill, 1999
- [2] Lyu M. Handbook on Software Reliability Engineering. New York: McGraw Hill, 1996
- [3] Xie M, Yang B. A study of the effect of imperfect debugging on software development cost. IEEE Transactions on Software Engineering, 2003, 29(5): 471-473
- [4] Kuo S Y, Huang C Y, Lyu M R. Framework for modeling software reliability, using various testing-efforts and fault-detection rates. IEEE Transactions on Reliability, 2001, 50(3): 310-320
- [5] Huang C Y, Lyu M R, Kuo S Y. A unified scheme of some nonhomogenous poisson process models for software reliability estimation. IEEE Transactions on Software Engineering, 2003, 29(3): 261-269
- [6] Wu Y P, Hu Q P, Xie M. Modeling and analysis of software fault detection and correction process by considering time dependency. IEEE Transactions on Reliability, 2007, 56(4): 629-642
- [7] Teng X, Pham H. A new methodology for predicting software reliability in the random field environments. IEEE Transactions on Reliability, 2006, 55(3): 458-468
- [8] Huang C Y, Huang W C. Software reliability analysis and measurement using finite and infinite server queuing models. IEEE Transactions on Reliability, 2008, 57(1): 192-203
- [9] Dai Y S, Xie M, Poh K L. Modeling and analysis of correlated software failures of multiple types. IEEE Transactions on Reliability, 2005, 54(1): 100-106
- [10] Zhang X, Teng X, Pham H. Considering fault removal efficiency in software reliability assessment. IEEE Transactions on Systems, Man and Cybernetics, Part A, 2003, 33(1): 114-120
- [11] Liu Hong-Wei, Yang Xiao-Zong, Qu Feng, Dong Jian. A Software Reliability Growth Model with Bell-Shaped Fault Detection Rate Function. Chinese Journal of Computers, 2005, 28(5): 908-913(in Chinese)
(刘宏伟, 杨孝宗, 曲峰, 董剑. 一个基于响铃形故障检测率函数的软件可靠性增长模型. 计算机学报, 2005, 28(5): 908-913)
- [12] Zou Feng-Zhong, Li Chuan-Xiang. A dual stochastic model for software reliability. Journal of Software, 1998, 10(1): 69-73(in Chinese)
(邹丰忠, 李传湘. 双随机软件可靠性模型的建模. 软件学报, 1998, 10(1): 67-73)
- [13] Zhao Jing, Zhang Ru-Bo, Gu Guo-Chang. Study on software reliability growth model considering failure dependency. Chinese Journal of Computers, 2007, 30(10): 1713-1720(in Chinese)
(赵靖, 张汝波, 顾国昌. 考虑故障相关的软件可靠性增长模型研究. 计算机学报, 2007, 30(10): 1713-1720)
- [14] Lou Jun-Gang, Jiang Jian-Hui, Shuai Chun-Yan, Jin Ang. Software reliability prediction model based on relevance vector machine//Proceedings of the 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009, 1: 229-233
- [15] Bai C G. Bayesian network based software reliability prediction with an operational profile. The Journal of Systems and Software, 2005, 77(3): 103-112
- [16] Campodónico S, Singpurwalla N D. Bayesian analysis of the logarithmic-Poisson execution time model based on expert opinion and failure data. IEEE Transactions on Software Engineering, 1994, 20(9): 677-683
- [17] Dai Y S, Xie M, Long Q, Hui S. Uncertainty analysis in software reliability modeling by Bayesian approach with maximum-entropy principle. IEEE Transactions on Software Engineering, 2007, 33(11): 781-795

- [18] Karunanithi N, Whitley D, Malaiya Y K. Prediction of software reliability using connectionist models. *IEEE Transactions on Software Engineering*, 1992, 18(7): 563-574
- [19] Karunanithi N, Whitley D, Malaiya Y K. Using neural networks in reliability prediction. *IEEE Software*, 1992, 9(4): 53-59
- [20] Su Yu-Shen, Huang Chin-Yu. Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models. *Journal of Systems and Software*, 2007, 80(4): 606-615
- [21] Zhang Yong-Qiang, Sun Sheng-Juan. Software reliability modeling based on unascertained theory. *Journal of Software*, 2006, 17(8): 1681-1687(in Chinese)
(张勇强, 孙胜娟. 基于未确知理论的软件可靠性建模. *软件学报*, 2006, 17(8): 1681-1687)
- [22] Tian L, Noore A. Dynamic software reliability prediction: An approach based on support vector machines. *International Journal of Reliability, Quality and Safety Engineering*, 2005, 12(4): 309-321
- [23] Zou Feng-Zhong, Li Chuan-Xiang. A chaotic model for software reliability. *Chinese Journal of Computers*, 1999, 24(3): 281-291(in Chinese)
- (邹丰忠, 李传湘. 软件可靠性混沌模型. *计算机学报*, 1999, 24(3): 281-291)
- [24] Rudin W. *Real and Complex Analysis*. New York: McGraw Hill, 1987
- [25] Yamada S, Ohba M, Osaki S. S-shaped software reliability growth modeling for software error detection. *IEEE Transactions on Reliability*, 1983, 32(5): 475-478
- [26] Parr F N. An alternative to the Rayleigh curve for software development effort. *IEEE Transactions on Software Engineering*, 1980, 6(3): 291-296
- [27] Kan S H. *Metrics and Models in Software Quality Engineering*, New Jersey: Addison-Wesley, 2003
- [28] Huang C Y, Kuo S Y, Lyu M R. An assessment of testing-effort dependent software reliability growth models. *IEEE Transactions on Reliability*, 2007, 56(2): 198-211
- [29] Ohba M. Software reliability analysis models. *IBM Journal on Research and Development*, 1984, 28(5): 428-443
- [30] Wood A P. Predicting software reliability. *IEEE Computer*, 1996, 29(11): 69-77
- [31] Shepperd M, Schofield C. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering*, 1997, 23(11): 736-743



LOU Jun-Gang, born in 1982, Ph. D. candidate. His research interests include dependable computing, software reliability evaluation and computer system performance evaluation.

JIANG Jian-Hui, born in 1964, Ph. D., professor, Ph. D. supervisor. His research interests include fault-tolerant computing, computer system performance evaluation, software reliability engineering and micro-processor architecture.

Jin Ang, born in 1982, Ph. D. candidate. His research interests include software reliability engineering, fault injection.

Background

In modern society, computers are used for many different applications, such as nuclear reactors, aircraft, banking systems, and hospital patient monitoring systems. As the demand of the application quality becomes higher and higher, the research of the computer software reliability becomes more and more essential. The software reliability is defined as the probability that the software will operate without a failure under a given environmental condition during a specified period of time. To date, the software reliability model is one of the most important tools in software reliability assessment.

The software failure data to be analyzed should be considered as a particular realization of a stochastic process. In this paper, a new software reliability growth model considering warps between different software failure processes is proposed. The experimental results based on two real data sets show that the proposed model has better prediction and curve fitting abilities than that of some other conditional NHPP

software reliability growth models. The measures used for comparison are mean square of fitting error, predicted error and relative error.

This paper focuses on the evaluation accuracy and prediction precision of software reliability model considering warps between different software failure processes. In the other work, the authors have worked on the software reliability models based on Relevance Vector Machine (RVM), Gray theory and so on. Some progress has been achieved. Furthermore, the authors adopted certain algorithms to select the most appropriate model parameters.

This work was supported by the National High Technology Research and Development Program (863 Program) of China under grant No. 2007AA01Z142, and by the Key Science-Technology Project of Science and Technology Commission of Shanghai, China under grant Nos. 04DZ15032, 06DZ15003.