

负指数分布排错时间的软件可靠性模型

孙 靖, 刘晓明

(解放军理工大学通信工程学院 南京 210007)

【摘要】传统的软件可靠性模型大多假设排除错误的时间忽略不计, 基于假设过于苛刻, 影响了模型应用的范围和精度。因此在JM模型的基础上, 提出了排错时间为负指数分布的软件可靠性模型及本模型的极大似然参数估计方法。通过用马尔可夫过程描述系统的状态, 分析了模型的数值求解方法, 并讨论了使用在本模型中的失效预测、剩余错误预测和等待排出的错误数预测公式。同时利用实际的工程数据对本模型的应用过程进行了验证。

关键词 软件可靠性模型; 软件可靠性; 排错时间; JM模型

中图分类号 TP391 文献标识码 A

A Software Reliability Model with Exponential Repair Time

SUN Jing, LIU Xiao-ming

(PLA University of Science and Technology Nanjing 210007)

Abstract Conventional software reliability models assume that the time to remove a fault is negligible. This rigid assumption limits the models' prediction precision and their application. Based on JM model, we propose a software reliability prediction model involving fault-remove time which followed exponential distribution. We provide the maximum likelihood parameter estimation equations of this model. We also provide several reliability prediction equations by using markov process. Illustrations of the proposed model by using real data are provided.

Key words software reliability model; software reliability; fault-remove time; JM model

在现有的软件可靠性模型中, 绝大多数都假设排除错误的时间忽略不计, 这些模型的可靠性预测都依赖于这条假设。显然, 忽略排错时间的假设过于苛刻, 尤其在软件工程的实践中, 排错过程已经成为影响软件可靠性的重要环节。在得不到排错过程数据的情况下, 忽略排错时间使得可靠性模型变得简单, 同时也降低了模型精度和结果的可信度。随着软件开发过程管理水平的提高, 在有条件获得排错过程的有关数据条件下, 则应将考虑排错过程对可靠性的影响, 本文提出了考虑排错时间的可靠性模型。

1 JM模型

JM模型(Jelinski and Moranda Model, JM)由文献[1]提出, 是最早的软件可靠性模型之一, 该模型对软件可靠性定量分析技术的建立和发展做出了重要的贡献^[2, 3]。

模型基本假设如下: 1) 软件测试的方式与使用方式相同; 2) 软件每个错误导致系统失效的可能性相同; 3) 软件中的错误相互独立, 各次失效的间隔时间也相互独立; 4) 当失效发生后, 导致失效的软件错误立刻被排除, 排错时间忽略不计, 且不引入新的错误; 5) 在测试开始时, 软件中有 N_0 个错误。 N_0 是一个未知常数; 6) 软件的失效率 Z 在每个失效间隔内是常数, 其数值与软件中残留的错误数成正比, 比例系数为 ϕ 。假

收稿日期: 2003-06-01

作者简介: 孙 靖(1974-), 男, 博士生, 主要从事软件工程与软件可靠性方面的研究; 刘晓明(1956-), 男, 教授, 主要从事软件工程与指挥自动化方面的研究。

假设1)保证了使用测试数据对可靠性进行了模型评估的正确性。假设2)使得不同的失效率具有相同的分布特性。假设3)排除了错误之间的相互影响。假设4)则将测试阶段中的查错、排错两种活动简化为只有查错一种活动,忽略了排错过程和排错结果对可靠性的影响。假设1)~4)广泛应用在多种软件可靠性模型之中^[2,3]。

根据JM模型的假设5)、6)可以得知,在第*i*个测试区间内,失效率函数为:

$$Z(\Delta t / t_{i-1}) = \Phi(N_0 - i + 1)$$

假设失效发生时刻的随机变量为 ξ ,用 $F(t)$ 表示 ξ 的分布函数。根据状态转移概率的定义可以得到:

$$Z(\Delta t | t) = \lim_{\Delta t \rightarrow 0} \frac{P\{t + \Delta t \geq \xi > t | \xi > t\}}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{P\{t + \Delta t \geq \xi > t\}}{\Delta t P\{\xi > t\}} = \lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{h(1 - F(t))} = \frac{F'(t)}{1 - F(t)} \quad (1)$$

求解这个一阶常系数微分方程,在初始条件 $F(0)=0$ 的情况下,可得分布函数:

$$F(t) = 1 - \exp\{Z(\Delta t | t_{i-1})t\} = 1 - \exp\{-\Phi(N_0 - i + 1)t\} \quad (2)$$

可见,系统在经历了*i*-1次失效后,第*i*次失效发生的时间服从参数 $\Phi(N_0 - i + 1)$ 的负指数分布。其密度函数 $f(t) = F'(t) = \Phi(N_0 - i + 1)\exp\{-\Phi(N_0 - i + 1)t\}$, 可靠性函数 $R(t) = \exp\{-\Phi(N_0 - i + 1)t\}$, 第*i*次失效与第*i*-1次失效的间隔时间 $= MTBF_i = \frac{1}{\Phi(N_0 - i + 1)}$, 则第*k*次失效发生的时间 $t_k = \sum_{i=1}^k MTBF_i$ 。失效次数的均值公式为:

$$m(t) = N_0 [1 - \exp(-\Phi t)] \quad (3)$$

剩余错误数的均值为:

$$e(t) = N_0 \exp(-\Phi t) \quad (4)$$

2 负指数分布排错时间的软件可靠性模型

2.1 基本假设

本模型的假设包括JM模型的假设1)~3)和假设5),对假设4)和假设6)进行如下修改。

2.1.1 对假设4)的修改: 负指数时间的排错过程

假设4.1),每次的软件失效是由一个错误引起的;假设4.2),当失效发生后,排错活动立刻进行;假设4.3),每次排除一个错误,排错顺序按照失效的顺序进行,如果新的排错任务到达时上一个错误仍未被排除,则新的排错任务进入排队;假设4.4),排错是完美的,引起失效的错误一定能够排除,且不引入新的错误;假设4.5),错误的排除时间服从负指数分布。在上述排错过程假设的5个方面中,假设4.1)可以从原JM模型的假设2)和假设3)推导得到;假设4.2)和假设4.3)反映了大多数软件排错活动的特点;假设4.4)对实际的排错活动进行了简化,本模型暂不考虑不完美排错的情况。

负指数分布能够较好地描述许多人工活动的服务时间^[4];文献[5]对178个排错时间样本进行了分析,认为排错时间服从负指数分布;在文献[4]中,通过Shuttle项目中包含85个数据项的排错时间样本分析,认为其排错时间也服从参数为平均排错时间的负指数分布。由于负指数分布的排错时间具有较为广泛的代表性,因此,本模型假设排错时间服从参数为 μ_0 的负指数分布,其分布函数为: $F(t) = 1 - \exp(-\mu_0 t)$ 。

2.1.2 对假设6)的修改: 失效过程假设

假设6.1),软件的瞬时失效率 Z 在每个失效间隔内是常数,其数值与软件中残留的错误数成正比,比例系数为 Φ ,假设6.2)一旦错误被排除,立刻以新的软件版本为对象进行测试。本模型假设6.1)具有与原JM模型相同的实质,但瞬时失效率具有不同的表达式。假设当前已经发生了*i*次失效,且已修复了*j*个错误($j \leq i$),则称系统当前处于状态(*i*, *j*)。根据假设6),处于状态(*i*, *j*)的系统的瞬时失效率为 $Z(t | i) = \Phi(N_0 - j)$ 。从上式可以看出,在初始错误一定的情况下,瞬时失效率只与已经排除的错误数有关,而与失效次数无关。这是本模型与JM模型失效过程的最大区别。假设6.2)符合一般的测试过程,该假设将影响到从失效时刻得到失效间隔时间的处理。

2.2 模型的马尔可夫过程

如果用二元组(*i*, *j*)表示系统的当前状态,其中*i*为失效次数,*j*为已经修复的错误数, N_0 为初始错误数,则系统状态的变化构成一个连续时间的马尔可夫过程,如图1所示,其中瞬时状态转移概率为:

$$\lambda_{(i,j)} = \begin{cases} \Phi(N_0 - j) & 0 \leq i \leq N_0 - 1, 0 \leq j \leq i \\ 0 & \text{其他} \end{cases} \quad \mu_{(i,j)} = \begin{cases} \mu_0 & 0 \leq i \leq N_0, 0 \leq j \leq i - 1 \\ 0 & \text{其他} \end{cases} \quad (5)$$

2.3 参数的极大似然估计

假设测试过程中测得失效间隔时间为 $\{x_{i,j}\}$ (i 为失效次数, j 为已经修复的错误数, $i=1,2,\dots,n, j \leq i$)。

则极大似然函数为: $\ln L(N_0, \Phi) = \sum_{i=1}^n (\ln \Phi(N_0 - j + 1) - \Phi(N_0 - j + 1)x_{i,j})$ 。

$$\text{令 } \frac{\partial \ln L(N_0, \Phi)}{\partial N_0} = 0, \quad \frac{\partial \ln L(N_0, \Phi)}{\partial \Phi} = 0,$$

可得到方程组:

$$\begin{cases} \sum_{i=1}^n \frac{1}{N_0 - j + 1} - \sum_{i=1}^n \Phi x_{i,j} = 0 \\ \frac{n}{\Phi} - \sum_{i=1}^n (N_0 - j + 1)x_{i,j} = 0 \end{cases} \quad (6)$$

利用数值计算方法求解上述方程组, 可以得到点估计值 \hat{N}_0 和 $\hat{\Phi}$ 。如果测得实际用于排错的时间为 $y_1, y_2, y_3, \dots, y_n$, 则得到 μ_0 的点估计值:

$$\hat{\mu}_0 = \left(\frac{1}{n} \sum_{i=1}^n y_i \right)^{-1} \quad (7)$$

2.4 失效时间预测

通过图1所示的马尔可夫过程, 可以列出平衡方程组

$$\begin{cases} P'_{i,j}t = -(\lambda_{i,j} + \mu_{i,j})P_{i,j}t + (\lambda_{i-1,j})P_{i-1,j}t + (\mu_{i,j-1})P_{i,j-1}t \quad (i = 0,1,2,\dots,N_0, j = 0,1,2,\dots,i) \\ P_{i,j}(0) = \begin{cases} 1 & i = 0, j = 0 \\ 0 & \text{其他} \end{cases} \end{cases} \quad (8)$$

由于 $N_0, \lambda_{i,j}, \mu_{i,j}$ 是已知的, 上述方程组只包含有限个常系数微分方程, 因而是可解的。求解上述方程组, 即得到 $P_{i,j}(t)$ 。失效次数的均值函数为:

$$m(t) = \sum_{i=0}^N \sum_{j=0}^i iP_{i,j}(t) \quad (9)$$

剩余错误数均值的预测公式:

$$e(t) = \sum_{i=0}^N \sum_{j=0}^i (N_0 - j)P_{i,j}(t) \quad (10)$$

待排除错误数均值的预测公式:

$$c(t) = \sum_{i=0}^N \sum_{j=0}^i (i - j)P_{i,j}(t) \quad (11)$$

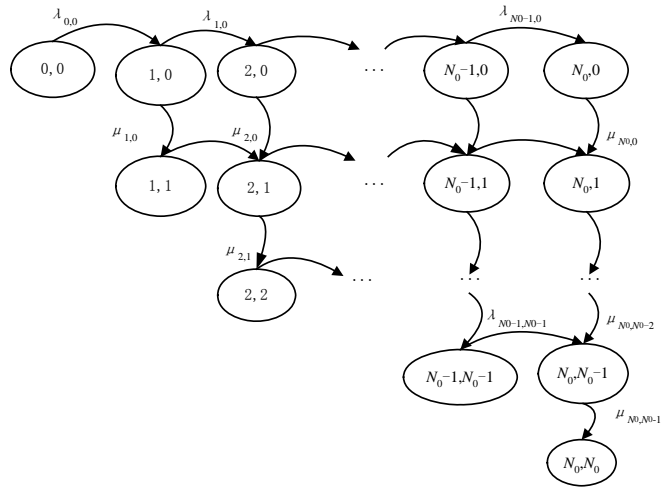


图1 负指数排错时间软件可靠性预测模型的状态变化图

3 模型应用

3.1 原始数据

文献[6]介绍了NASA的Shuttle项目某航电软件子系统OIO模块的失效时间数据和排错时间数据, 如图2所示。在图2中, 事件后标注的数字表示事件次数。排错3表示在该时刻排除了3个错误, 失效2表示在该时刻发生了2次失效。

3.2 数据处理与参数估计

以原始数据为基础, 得出失效间隔时间数据和排错消耗时间数据。失效间隔时间为两个相邻失效时刻之差。对于在某个时刻发生多次失效情况(如4、5次), 将平均分配失效间隔时间。排错消耗时间等于错误修正时刻减去错误发生时刻。对于在某个时刻排除多个错误的情况, 也将平均分配排错消耗时间。根据表1中的数据, 应用式(6)、(7)进行参数估计, 得到本模型的参数分别为: $N_0=8.987\ 38, \Phi=0.112\ 02, \mu_0=0.247\ 35$ 。

利用式(4), 得到JM模型的参数分别为: $N_0=7.17789$, $\Phi=0.07106$ 。

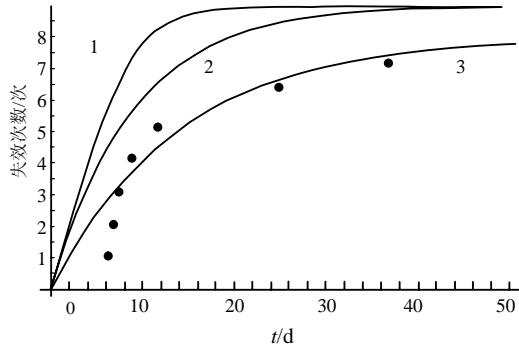


图2 失效时间预测

表1 OIO软件模块的失效和排错数据

事件	时刻/d	处理后的数据		
		JM模型失效间隔时间/d	本文模型失效间隔时间/d	模型排错消耗时间/d
失效	5.77	5.77	5.77	3.30
失效	5.90	0.13	0.13	1.78
失效	7.53	1.63	1.63	1.78
排错	9.07	1.20	1.97	1.78
失效(2)	11.47	1.20	1.97	6.30
排错(3)	16.80	13.20	7.87	6.30
失效	24.67	11.50	6.77	9.60
排错(2)	29.40			
失效	36.17			
排错	45.77			

3.3 模型预测结果

利用式(3)、(5)、(8), 对OIO模块分别应用JM模型和本模型进行失效预测, 预测曲线如图3所示。

在图2中, 线条1为本文模型的预测结果, 线条3为JM模型的预测结果, 线条2为当 $N_0=8.98738$, $\Phi=0.11202$ 时使用JM模型的预测结果。图中的黑点为实际的失效时间。从图中可以看出, 对于OIO模块的失效数据, JM模型预测结果具有最佳的近似性。然而, 该数据集只包含7个失效事件, 对于失效预测来说显得太少。对比线条1和2, 在参数相同的情况下, 考虑排错时间的模型在同一时刻具有更多的失效次数。

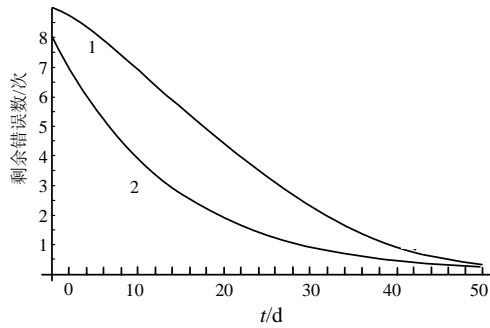


图3 剩余错误数预测

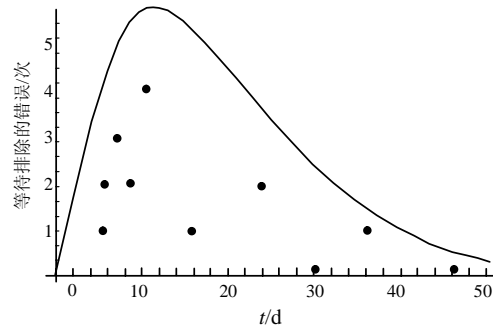


图4 等待排除的错误数量预测

应用式(10), 预测软件的剩余错误数, 如图3所示。图3中线条1为本模型的预测结果, 线条2是JM模型的预测结果(应用式(4))。由于考虑了排错时间, 本模型预测的初始错误数大于JM模型的初始错误数, 错误减少的速率较JM模型小。在同一时刻内, 本模型的剩余错误数大于JM模型的剩余错误数。

应用式(11), 可以预计有多少已经测出的错误正在等待排除, 如图4所示。图4中的线条是本模型的预计结果, 黑点为测量值。

参 考 文 献

- [1] Jelinski Z, Moranda P B. Software reliability research[C]. Proceedings of the Statistical Methods for the Evaluation of Computer System Performance, Academic Press, 1972. 465-484
- [2] Michael R Lyu编, 刘喜成译. 软件可靠性工程手册[M]. 北京: 电子工业出版社, 1997. 51-52
- [3] 黄锡滋. 软件可靠性安全性与质量保证[M]. 北京: 电子工业出版社, 2002. 86-92
- [4] Kleinrick L. Queuing systems, Volume 1: Theory[M]. John Wiley & Sons, New York, 1975
- [5] Musa J D. Software reliability measurement, Prediction, Application[M]. McGraw - Hill, New York, 1987
- [6] Schneidewind N F. Modeling the fault correction process[C]. Proceedings of the 12th ISSRE, (IEEE Computer Society, Hong Kong, November, 2001). IEEE Press.185-201