◎研发、设计、测试◎

Weibull 分布在软件最优交付时间估算中的应用

李 钧[1],代祖华[2]
LI Jun[1],DAI Zu-hua[2]

1.西安交通大学 管理学院,西安 710049
2.西北师范大学 数信学院,兰州 730070
1.School of Management,Xi'an Jiaotong University,Xi'an 710049,China
2.College of Mathematics and Information,Northwest Normal University,Lanzhou 730070,China
E-mail:lijun56@hotmail.com

LI Jun,DAI Zu-hua.Application of Weibull distribution model in estimating the best time for software delivery.Computer Engineering and Applications,2009,45(23):67-69.

Abstract: The best software delivery time of a software product is highly determined by the software reliability estimation. Weibull distribution is an effective model for software reliability analysis,which well describes statistical characteristics of defect detection in software testing.Based on the analysis of the Weibull distribution characteristics,a Weibull distribution accumulative function fitting model is suggested,and the discontinuity problem of defect detection is resolved.Furthermore,by making use of the model,two kinds of formula to evaluate the best time for software delivery are given.Finally,an example is analyzed to prove the model's effectiveness.

Key words: software testing;Weibull distribution;software reliability;best software delivery time

摘 要:软件最优交付时间在很大程度上取决于软件可靠性估算。Weibull 分布较有效地描述了软件测试过程中错误发现的统计特征,是一种有效的软件可靠性分析模型。在分析 Weibull 分布的基础上,提出了 Weibull 分布的累计函数拟合模型,解决了错误发现的不连续问题,推导出两种利用 Weibull 累计函数模型进行软件最优交付时间的估算公式,并结合具体实例验证了模型的有效性。

关键词:软件测试;Weibull 分布;软件可靠性;软件最佳交付时间
DOI:10.3778/j.issn.1002-8331.2009.23.019 **文章编号:**1002-8331(2009)23-0067-03 **文献标识码:**A **中图分类号:**TP311

可靠性是软件产品的重要质量要素之一[1]。在软件开发过程中,可靠性通过软件测试来保证。软件的测试是一个不断反复的过程,什么时候软件达到了要求的可靠性水平,从而能够投入使用是一个关键问题。如果不具有足够的可靠性水平,过早地将软件投入使用,其后果轻则给用户带来麻烦,增加额外的软件维护成本,重则导致安全事故的发生。而测试到了一定阶段后,软件可靠性增长缓慢,继续进行测试将浪费人力、财力。对于商业软件来说,则可能影响其进入市场的时机而造成损失。因此,必须把软件可靠性这一软件质量指标数量化,用定量的数学方法来研究软件产品的可靠性[2-3]。可靠性模型是软件可靠性定量分析技术的基础,利用可靠性模型,项目管理人员就可通过分析软件失效率与软件错误的关系,对软件可靠性变化趋势进行描述和预计,从中得到的信息,将定量、客观地反映测试进展情况,为控制项目进度和资源分配提供重要的决策依据。参照硬件可靠性的研究方法,国内外学者建立了各种不同的软件可靠性模型。其中 Weibull 分布模型是由我国南京电子工程研究所提出的一种软件可靠性模型[4]。研究表明[1,4],该模型比较科学地描述了软件测试过程中软件错误数随时间分布的特征,因此有必要对 Weibull 函数模型在可靠性分析中的应用进行深入研究。

1 Weibull 分布函数的一般形式及其曲线特征[5]

Weibull 分布函数的一般形式是

$$f(t)=ma(t-r)^{m-1}\mathrm{e}^{-a(t-r)^m} \quad (1)$$

Weibull 分布函数由 m、a、r 三个参数唯一确定。其中 m 是函数曲线的形状参数,其值的大小决定了 Weibull 分布曲线的峰值位置;a 是 Weibull 比例常数,其值的大小决定了曲线陡度;m、a 值的不同组合,形成了包括偏斜分布、低峰态分布、高峰态分布和正态分布等多种分布函数;r 为位置参数,表示测试项目的起始点,实际研究中,一般取初始时刻为 0。则:

$$f(t)=mat^{m-1}\mathrm{e}^{-at^m} \quad (2)$$

图 1 给出了 Weibull 函数不同参数值的曲线基本形状。

作者简介:李钧(1956-),女,硕士学位,研究方向:软件工程;代祖华(1971-),女,博士学位,研究方向:信息系统工程与管理。
收稿日期:2009-04-03 **修回日期:**2009-05-04

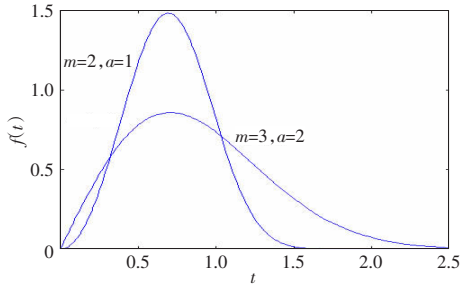

图 1 Weibull 曲线基本形状

由图 1 可以看出,当 $m\geqslant 1$ 时,函数存在一个唯一的峰值点 t^*,使得 $f(t^*)$ 达到最大值。对式(2)两边求导,并令其导数为零,可得:

$$t^*=\left(\frac{m-1}{ma}\right)^{\frac{1}{m}} \tag{3}$$

则:

$$f(t^*)=ma\left(\frac{m-1}{ma}\right)^{\frac{m-1}{m}}e^{-\frac{m-1}{m}}=\frac{m-1}{e}\left(\frac{ema}{m-1}\right)^{\frac{1}{m}} \tag{4}$$

因此,Weibull 曲线的峰值位置取决于参数 m、a 的值。

2 软件错误发现数的 Weibull 累计函数拟合模型

对软件错误发现数的统计表明[6]:在软件测试项目的早期,软件错误发生数呈不断增长的趋势,且增长到某一转折点开始逐渐减少,最终变为每天只发现一两个软件错误。这说明,在软件测试项目中,对应的软件可靠性增长趋势要推迟到测试过程的中后期,而在测试项目的早期,反而出现可靠性衰减的现象。其原因是:测试的早期阶段,工作尚待展开,测试人员对程序结构、软件特征了解甚少,故而发现的错误数较少,随着测试的深入,测试人员开始熟悉被测对象的特征,查错能力也随之增强,而随着软件缺陷的不断修复,在测试过程的中后期,错误发现数开始逐渐减少。因此,软件错误发现数的统计变化规律比较符合 Weibull 分布的曲线特征。

2.1 Weibull 分布应用于软件可靠性分析的几个假设

影响软件可靠性的因素很多,企图用某种数学关系详尽而准确描述和计算软件可靠性是不可能的。因此,需要对各种影响软件可靠性的因素进行取舍、简化和限制,省略次要因素,保留主要因素,以此作为建立数学模型和可靠性分析的基础。这些简化和限制就构成了模型应用于软件可靠性分析的假设。模型分析和应用的正确与否,直接取决于这些基本假设与实际环境的吻合程度。当人们开始研究和应用一个模型时,了解这些基本假设是至关重要的。应用 Weibull 模型在软件测试过程中进行可靠性分析的基本假设是:(1)软件的失效率正比于软件内部现有的错误数;(2)测试中一旦发现错误,即予以改正,且不引入新的错误;(3) 软件全部残余错误等概率于软件故障发生率,软件出错的各时间段独立统计。

基于以上假设,若用 $N'(t)$ 表示测试阶段某时刻 t 的错误发现数,N_0 表示软件固有的错误总数, 则软件测试过程中错误发现数随时间分布的 Weibull 函数模型为:

$$N'(t)=N_0 mat^{m-1}e^{-at^m} \tag{5}$$

模型中其他参数的定义同式(1)。

2.2 Weibull 累计函数拟合模型

虽然错误发现数以 Weibull 分布描述是合理的、统计上是显著的。但实际测试过程中的错误发现数并未按照理想情况连续发生,故有必要研究 Weibull 分布的累积函数形式,以分析在一定时间内发生的软件错误数。

对式(5)两边对时间 t 积分,则产生 Weibull 分布的累积函数如下:

$$N(t)=\int_0^t N'(t)\mathrm{d}t=\int_0^t N_0 mat^{m-1}e^{-at^m}\mathrm{d}t=N_0(1-e^{-at^m}) \tag{6}$$

其中 $N(t)$为时间 t 内累计发生的错误。显然,Weibull 累积分布曲线呈 S 曲线形态。

对式(6)进行如下变换:

$$\frac{N_0-N(t)}{N_0}=e^{-at^m} \tag{7}$$

将式(7)代入式(5)有:

$$N'(t)=ma(t-r)^{m-1}(N_0-N(t)) \tag{8}$$

其中,$N_0-N(t)$表示某时刻剩余软件错误数。由式(8)可知:在软件测试过程中,某时刻的错误发现数与软件剩余错误数成正比。

设时间单位为周,$N(t,t-1)=N(t)-N(t-1)$表示第 t 周的软件错误发现总数,则:

$$N(t,t-1)=N_0(e^{-a(t-1)^m}-e^{-at^m}) \tag{9}$$

令 $N_c(t)$表示在第 t 周测试后软件剩余的错误数,则:

$$N_c(t)=N_0-N(t)=N_0e^{-at^m} \tag{10}$$

3 Weibull 函数参数的估值方法

使用模型进行软件可靠性预测,须对模型中的未知参数做出估计。软件测试中所收集的错误发现数是参数估计的基础。有了充足、可靠的样本数据之后,还需结合科学的参数估值方法,才能得到有统计意义和使用价值的计算模型。在实际应用中,要对 Weibull 分布的三个模型参数进行估算,常用的估算方法有以下几种:

3.1 专家估算法

该方法主要用于测试计划阶段,它是利用 Weibull 函数本身的性质,通过有丰富测试经验的测试专家估计错误发生的峰值点 t_h 和测试结束时间 t_s,并进而估计 m、a。具体估算方法如下:

由式(3)可得:

$$a=\frac{m-1}{mt_h^m} \tag{11}$$

由式(6),显然 $\lim\limits_{t\to\infty}N(t)=N_0$,令 t_s 为测试结束时间,则 t_s 时,要求测试发现的累积错误总数为总错误数的 99.5%, 则式(6)满足如下等式:

$$N(t_s)=\int_0^{t_s}N_0 mat^{m-1}e^{-at^m}\mathrm{d}t=N_0(1-e^{-at_s^m})=0.995N_0$$

所以:

$$1-e^{-at_s^m}=0.995 \tag{12}$$

将式(12)代入式(11)使用迭代法求出 m 值,然后利用式(12)计算 a 值。

3.2 最小二乘法

根据样本数据, 结合线性回归的最小二乘法进行参数拟合,直至获得满意的参数值。拟合时对式(6)进行如下线性变换:

$$\ln\left(1-\frac{N(t)}{N_0}\right)=-at^m$$

$$\ln\left(\frac{1}{1-N(t)/N_0}\right)=at^m$$

$$\ln\left(\ln\left(\frac{1}{1-N(t)/N_0}\right)\right)=\ln a+m\ln t \tag{13}$$

令 $Y=\ln(\ln(1/(1-N(t)/N_0)))$,$A=\ln a$,$B=m$,$X=\ln t$,得到线性拟合方程 $Y=A+BX$。对式(13)的拟合结果可以确定 Weibull 累积分布曲线的参数 a、m。

由式(13)可以看出,由于 $Y=\ln(\ln(1/(1-N(t)/N_0)))$,所以参数拟合前需要根据样本数据对软件固有的总错误数 N_0 进行估计。进而可将各阶段累积错误数转化为占总错误数的百分比。事实上,即使在样本数据充足的情况下要对 N_0 进行定量化的估计也是极其困难的。而且,在拟合开始时就要估计其参数值,其误差无疑将对拟合结果产生一定的影响,因此这种拟合在实际应用中存在局限性。

3.3　高斯-牛顿迭代法[5]

此方法常用于对非线性函数的参数估计,其基本思想是用非线性函数的一阶 Taylor 展开式近似代替原函数,然后利用线性最小二乘法进行参数估计,通过多次迭代产生不断逼近原函数的参数估计值,最后使得原函数的残差平方和最小。在样本资料充足的条件下,这种方法计算过程简单、准确度高、可借助于软件编程实现。

设非线性函数的向量形式如下:

$$\boldsymbol{Y}=\boldsymbol{f}(t,\boldsymbol{\theta}) \tag{14}$$

其中 $\boldsymbol{\theta}$ 为参数向量,则式(14)在 θ_0(参数初值)处一阶 Taylor 展开式为:

$$\boldsymbol{Y}=\boldsymbol{f}(t,\theta_0)+V(\theta_0)(\boldsymbol{\theta}-\theta_0) \tag{15}$$

其中,$\boldsymbol{Y}=(y_1,\cdots,y_n)'$,$t=1,\cdots,n$,$V(\theta_0)=\left(\frac{\partial \boldsymbol{f}(t,\boldsymbol{\theta})}{\partial\theta_i}\right)$是 $\boldsymbol{f}(t,\boldsymbol{\theta})$关于参数向量的一阶导数矩阵,为一个 $n\times p$ 阶矩阵(式中 n 为样本个数,$i=1,\cdots,p$,p 为参数个数),则式(15)为式(14)的线性近似式。线性模型式(14)的最小二乘估计 $\boldsymbol{\theta}$ 为:

$$\boldsymbol{\theta}=\theta_0+[V'(\theta_0)V(\theta_0)]^{-1}V'(\theta_0)[Y-f(\theta_0)] \tag{16}$$

则以 θ_0 为初值的参数迭代计算公式为:

$$\theta_{i+1}=\theta_i+[V'(\theta_0)V(\theta_0)]^{-1}V'(\theta_0)[Y-f(\theta_0)] \tag{17}$$

其中 θ_{i+1},θ_i 分别为第 $i+1$ 和 i 次的迭代结果,通常要求 $\|\theta_{i+1}-\theta_i\|$ 小于一个给定常数 ε 即可停止迭代过程。

4　Weibull 分布在软件最优交付时间估算中的应用

软件测试项目中,软件的最优交付时间估算取决于不同的判断标准,以下推导在两种不同判断标准下应用 Weibull 累计函数拟合模型的估算公式。

(1)以软件在规定的运行时间内达到预期的可靠性水平作为是否停止测试交付使用的标准

用软件失效率 λ_c 表示软件预期的可靠性水平[7],假设软件在交付用户使用的状态下,用户不具有软件排错的能力,这时,可认为软件失效率为常数,软件失效率等价于软件停止测试时的错误发生率,即 $\lambda_c=N'(T)$,其中 T 是软件停止测试交付使用的时间。根据式(5)有:

$$\lambda_c=N_0maT^{m-1}e^{-aT^m} \tag{18}$$

通过数值计算法求解上述非线性方程,即可推算出软件的预期交付时间 T。

(2)以软件生存周期内最低的错误修正费用作为判断的标准

设 C_1 是在测试中发现和改正一个错误所需的费用;C_2 是软件运行中发现和改正一个错误所需的费用;C_3 是单位测试时间的费用,T 是软件停止测试交付使用的时间,用表示软件生存周期内的时间,则在$(0,t)$时间内,错误修正费用总数的期望值为:

$$C(T)=C_1N(T)+C_2(N(t)-N(T))+C_3T \tag{19}$$

对上式关于 T 求导:

$$C'(T)=C_1N'(T)+C_2(-N'(T))+C_3 \tag{20}$$

令 $C'(T)=0$,则可得:

$$N'(T)=\frac{C_3}{C_2-C_1}=N_0maT^{m-1}e^{-aT^m}$$

线形变换上式有:

$$\ln\left(\frac{C_3}{(C_2-C_1)N_0ma}\right)=(m-1)\ln T-aT^m \tag{21}$$

利用数值计算法,由式(21)可估算软件的最优交付时间 T。

5　实例分析

表 1[1]是某软件系统在 24 周测试时间中得到的错误发生数。用表 1 的数据,采用高斯-牛顿迭代法,利用 Matlab 语言编程,对该软件系统的累计失效数进行了 Weibull 模型拟合,计算得出模型参数:$N_0=2\,247$,$a=0.017\,9$,$m=1.635$,即:

$$N'(t)=66.035t^{0.635}e^{-0.017\,9t^{1.635}} \tag{22}$$

$$N(t)==2\,247(1-e^{-0.017\,9t^{1.635}}) \tag{23}$$

表 1　某软件系统 24 周测试错误发生数据

时间/周	错误数	累计错误数	时间/周	错误数	累计错误数
1	67	67	13	47	1 622
2	106	173	14	46	1 668
3	97	270	15	71	1 739
4	129	399	16	54	1 793
5	77	476	17	57	1 850
6	143	619	18	80	1 930
7	116	735	19	64	1 994
8	131	866	20	27	2 021
9	187	1 053	21	42	2 063
10	203	1 256	22	55	2 118
11	136	1 392	23	62	2 189
12	183	1 575	24	11	2 191

图 2 是由式(23)得出的拟合曲线与实测数据的比较,可见拟合曲线的形状比较符合软件测试过程累计错误发现数的统计特征。

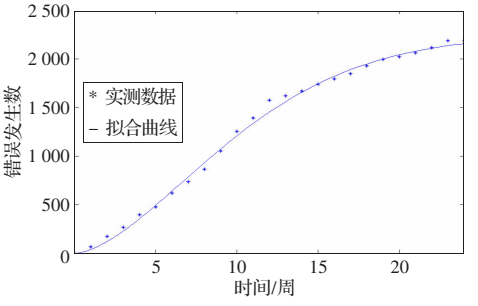

图 2　拟合曲线

(下转 106 页)