第13卷 第3期 自 动 化 学 报 Vol.13, No. 3
1987 年 5 月 ACTA AUTOMATICA SINICA May, 1987

超几何分布模型与软件可靠性

黄 锡 滋
（成都电讯工程学院）

摘 要

本文提出了用超几何分布进行软件可靠性试验时，根据试验数据计算软件可靠度、失效率及 MTBF 的方法，从而拓宽了模型的应用范围.

一、引 言

软件可靠性是七十年代才发展起来的一个新的学科分支，随着计算机的推广应用，这个学科已十分引人注目. 可以预料，在信息社会中，软件可靠性的重要性必将更为突出.

迄今为止，评价软件可靠性的方法和模型大约可分三类[1]，即面向错误数的模型、面向数据的模型和面向时间的模型. 本文主要讨论面向错误数的模型. 这个模型是 Mills 和 Basin 于 1972 年首先提出的[2]. 它是用超几何分布来分析试验数据，并对软件的固有错误数作出估计. 这个方法已被广为援引并在实际中有了一些应用，然而这个方法给出的是错误数的点估计值，对区间估计缺乏深入分析. 1984 年作者对这个方法的区间估计作了分析，推导出了相应的计算公式[3]. 最近的工作还表明，这个方法不仅可用来直接估计软件本身的错误数，而且也可用来计算软件的其它可靠性参量，例如可靠度、失效率和 MTBF，从而扩展了它的使用范围，使之成为一个在软件可靠性评估中有竞争力的数学模型.

二、可靠度、失效率及 MTBF 的计算

Nelson 关于软件可靠性的定义方法，是一种面向输入数据的方法[1]. 令 J 表示程序的总输入数据，J_e 表示导致程序出错的输入数据数，F 表示程序运行一次的失效概率，Nelson 定义

$$F=\frac{J_e}{J}. \tag{1}$$

软件运行一次的可靠度为

$$R=1-J_e/J. \tag{2}$$

软件运行 m 次的可靠度为

本文于 1985 年 4 月 5 日收到.

$$R(m)=(1-J_c/J)^m. \tag{3}$$

假设导致程序给出错误结果的输入数据正比于软件中存在的错误数,即

$$J_c \propto D. \tag{4}$$

其中 D 为软件中存在的错误数. 则

$$J_c/J=rD. \tag{5}$$

下面利用超几何分布模型的试验结果来估计比例常数的数值. 令 $j_i\, i\in(1, 2, \cdots, n)$ 表示在第 $i-1$ 次失效发生后直至 i 次失效发生这段时间中的测试次数,则

$$j=\sum_{i=1}^{n} j_i \tag{6}$$

是测试的总次数. 令 N_0 表示程序调试前的固有错误数, N_1 表示程序中加入的人为错误数,从式(5)可得

$$r_i=\frac{1}{j_i(N_0+N_1-i+1)}, \quad i\in(1, 2, \cdots, n). \tag{7}$$

再对各个 r_i 取平均值,得

$$\bar{r}=\sum_{i=1}^{n} r_i/n. \tag{8}$$

在程序调试前,其固有错误数为 N_0, 所以程序原有的可靠度为

$$R=(1-\bar{r}N_0). \tag{9}$$

Mills 和 Basin 给出 N_0 的点估计值为

$$\hat{N}_0=\left[\frac{N_1(n-k)}{k}\right]_{\text{integer}} \tag{10}$$

其中 k 为已检测出来的错误中的人为错误数,所以

$$R=\left(1-\bar{r}\,\frac{N_1(n-k)}{k}\right), \tag{11}$$

$$R(m)=\left(1-\bar{r}\,\frac{N_1(n-k)}{k}\right)^m. \tag{12}$$

程序调试结束后,所有的人为错误及已经发现的固有错误都被清除,这时程序的可靠度为

$$R=(1-\bar{r}(N_0-n+k)), \tag{13}$$

$$R(m)=(1-\bar{r}(N_0-n+k))^m. \tag{14}$$

有关软件可靠性定义的另一种方法是援用有关硬件可靠性的定义,这时还可采用失效率、MTBF 等参量表示其可靠性特征.

在调试的过程中,记下每一次运行的执行时间,用 $x_{1i}, i\in(1, 2, \cdots, j-n)$ 表示各次成功运行的时间, $x_{2i}\, i\in(1, 2, \cdots, n)$ 表示各次失败运行的时间,则平均成功运行一次的时间为

$$\Delta t_1=\frac{1}{j-n}\sum_{i=1}^{j-n} x_{1i}. \tag{15}$$

平均失败运行一次的时间为

$$\Delta t_2=\frac{1}{n}\sum_{i=1}^{n} x_{2i}. \tag{16}$$

因此，程序中如存在 D 个错误，则其平均运行一次的时间为

$$\Delta t' = \Delta t_1(1-\bar{r}D) + \Delta t_2 \bar{r}D. \tag{17}$$

如果软件经过调试后已交付使用，则 D 表示交付使用时程序中的残存错误数．令 t_m 表示要求的运行时间，则在 $(0\ t_m)$ 区间中平均运行次数为

$$m = t_m/\Delta t'. \tag{18}$$

从式(3)得到

$$R(t_m) = (1-\bar{r}D)^{t_m/\Delta t'}. \tag{19}$$

改写式(19)为

$$R(t_m) = e^{\frac{t_m}{\Delta t'}\ln(1-\bar{r}D)}. \tag{20}$$

则

$$R(t_m) = e^{-\frac{t_m}{\Delta t'}(\bar{r}D+\frac{1}{2}(\bar{r}D)^2+\frac{1}{3}(\bar{r}D)^3+\cdots\cdots)} \tag{21}$$

由此可得软件的失效率为

$$\lambda = \frac{1}{\Delta t'}\left(\bar{r}D + \frac{1}{2}(\bar{r}D)^2 + \frac{1}{3}(\bar{r}D)^3 + \cdots\right). \tag{22}$$

软件的 MTBF 为

$$\mathrm{MTBF} = \frac{\Delta t'}{\bar{r}D + \frac{1}{2}(\bar{r}D)^2 + \frac{1}{3}(\bar{r}D)^3 + \cdots}. \tag{23}$$

当 $\bar{r}D \ll 1$ 时，

$$\lambda = \frac{1}{\Delta t'}\bar{r}D = \frac{\bar{r}D}{(1-\bar{r}D)\Delta t_1 + \bar{r}D\Delta t_2}. \tag{24}$$

从式(22)及式(24)可见软件的失效率取决于软件中的固有错误数，当错误数不变时，失效率为常数．错误数减少，失效率也随之降低，而且当 $\bar{r}$ 很小时，从式(17)知 $\Delta t'$ 也近似于常数，这时

$$\frac{\bar{r}}{\Delta t'} \approx \Phi. \tag{25}$$

Φ 为常数．因此

$$\lambda = \Phi D. \tag{26}$$

式(26)与杰林斯基软件可靠性的基本假设完全相同．

参　考　文　献

[1] 黄锡滋，软件可靠性预测方法的进展，自动化学报，**11**(1985)，1—11.

[2] Schick G. J., Wolverton R. W., An analysis of competing software reliability modles, *IEEE Trans*. SE. 4 (1978), 104—119.

[3] Huang Xizi, The hypergeometric distribution modle for predicting the reliability of softwares, *Microelectron. Reliab*. **24**(1984), 11—20.

HYPERGEOMETRIC DISTRIBUTION MODEL AND SOFTWARE RELIABILITY

HUANG XIZI

(*Chengdu Institute of Radio Engineering*)

ABSTRACT

Primarily, the hypergeometric distribution model was only used to estimate the number of errors of softwares. This paper shows how to calculate the reliability, the failure rate and the MTBF of softwares quantitatively in accordance with the result of the test designed for using hypergeometric distribution model. Then, the applied scope of the model may be expanded.