

## A Software Reliability Growth Model

Khurshid Ahmad Mir

Department of Statistics, Govt. Degree College Bemina Srinagar, Kashmir, India

**Abstract:** A counting process representing the number of failures experienced in a given period of time by a system is proposed as a stochastic model for studying the reliability of the developed software. A Non Homogeneous Poisson Process (NHPP) with its mean value function specified by a Pareto model is considered. Its parameters are estimated to assess the reliability of a software system. The results are illustrated for a live software failure data.

**Key words:** Non homogeneous poisson process, mean value function, ML estimation, software, period, stochastic, model

### INTRODUCTION

Several studies have been undertaken to investigate the software error occurrence phenomenon. The objective of such studies is to improve software performance. These studies can be placed in one of the two categories. The first category emphasizes empirical analysis of data collected from software projects. The second category deals with the development of models for quantitative assessment of software performance.

Software reliability engineering is a discipline that ensures failure free operation of software at the user end by employing scientific techniques to remove the maximum number of faults. The quality of the software system has many attributes such as maintainability, portability, usability, security, reliability, availability, etc. Software reliability is the most dynamic attribute which can measure and predict the operational quality of the product.

The process of locating the faults in software to remove them is called the debugging process. The chronology of failure occurrence and fault removal can be utilized to provide an estimate of the software reliability and the level of fault content. Theory of probability plays a major role in software reliability model building. A software system is subject to failures at random times caused by errors present in the system. Let  $\{N(t), t \geq 0\}$  be a counting process representing the cumulative number of failures by time  $t$ . Since, there are no failures at  $t = 0$ , we have:

$$N(0) = 0 \quad (1)$$

It is reasonable to assume that the number of software failures during non overlapping time intervals do not affect each other. In other words for any finite collection of times  $t_1 < t_2 < \dots < t_n$ , the  $n$  random variables

$N(t_1), \{N(t_2) - N(t_1)\}, \dots, \{N(t_n) - N(t_{n-1})\}$  are independent. This implies that the counting process  $\{N(t), t \geq 0\}$  has independent increments. Several studies have been undertaken to investigate the software error occurrence phenomenon.

The objective of such studies is to improve software performance. These studies can be placed in one of the 2 categories. The first category emphasizes empirical analysis of data collected from software projects. The second category deals with the development of models for quantitative assessment of software performance. Let  $m(t)$  represent the expected number of software failures by time  $t$ . Since, the expected number of errors remaining in the system at any time is finite,  $m(t)$  is a bounded, non decreasing function of  $t$  with the following boundary conditions:

$$m(t) = 0, t = 0 = a, t \rightarrow \infty \quad (2)$$

where,  $a$  is the expected number of software errors to be eventually detected. Suppose  $N(t)$  is known to have a Poisson probability mass function with parameter  $m(t)$ :

$$P\{N(t) = n\} = \frac{[m(t)]^n \times e^{-m(t)}}{n!} \quad (3)$$

Then  $N(t)$  is called an NHPP. Various time domain models have appeared in the literature which describe the stochastic failure process by an NHPP which differ in the mean value functions  $m(t)$ . Some of them are due to Goel and Okumoto (1979), Littlewood (1981), Yamada *et al.* (1986), Musa *et al.* (1987), Hossain and Dahiya (1993) and Pham (2005). The mean value function is a non negative non decreasing function of  $t$  with a limit as  $t \rightarrow \infty$ . Specifically, all these characteristics are through with the cumulative distribution function of a continuous

random variable with the additional property as its limit is 1 as  $t \rightarrow \infty$ . Exploring this property of a distribution function multiplied by a positive constant is considered as the mean value function by many researchers to develop a number of software reliability growth models through NHPP. On these lines following Wood (1996) motivated by Littlewood (1981), the researchers have considered an NHPP with mean value function defined through the cumulative distribution of a Pareto random variable to propose it as an SRGM.

### THE PROPOSED SRGM

In this study, we consider  $m(t)$  as given by:

$$m(t) = a \left\{ 1 - \left[ 1 + \left( \frac{t}{\sigma} \right) \right]^{-\alpha} \right\} \quad (4)$$

where,  $[m(t)/a]$  is the cumulative distribution function of Pareto distribution of type IV (Johnson *et al.*, 2004) for the present choice :

$$\lim_{t \rightarrow \infty} P\{N(t) = x\} = \lim_{t \rightarrow \infty} \frac{[m(t)]^x \cdot e^{-m(t)}}{x!} = \frac{e^{-a} a^x}{x!} \quad (5)$$

Which is also a Poisson model with mean  $a$ . Let  $\overline{N}(t)$  be the number of errors remaining in the system at time  $t$ :

$$\begin{aligned} \overline{N}(t) &= N(\infty) - N(t) \\ E[\overline{N}(t)] &= E[N(\infty)] - E[N(t)] \\ &= a - m(t) = a - a \left\{ 1 - \left[ 1 + \left( \frac{t}{\sigma} \right) \right]^{-\alpha} \right\} \\ &= a(1 + t/\alpha)^{-\alpha} \end{aligned} \quad (6)$$

Let  $S_k$  be the time between  $(k-1)$ th and  $k$ th failures of the software product. Let  $X_k$  be the time up to the  $K$ th failure. Then the probability that  $S_k$  exceeds a real number  $s$ , given that the total time up to the  $(k-1)$ th failure is equal to  $x$  is:

$$P[S_k > s / X_{k-1}] = e^{-[m(x+s) - m(s)]} \quad (7)$$

This expression is called software reliability and is denoted by:

$$RS_k / X_{k-1}(s/x) \quad (8)$$

### ESTIMATION OF PARAMETERS

Let  $S_1, S_2$  be a sequence of times between successive software failures associated with an NHPP  $N\{t\}$ . Let  $X_k$  be equal to:

$$\sum_{i=1}^k S_i, \quad k=1, 2, 3, \dots$$

which represents the time to failure  $K$ . Suppose we are given  $n$  software failure times say  $x_1, x_2, \dots, x_n$  (i.e.), there are  $n$  time instants at which the 1st, 2nd, 3rd, ...,  $n$ th failures of a software are observed. This is a special case of a life testing experiment in which only one product is put to test and its successive failures are recorded alternately separated by error detections and debugging. The likelihood function of such sample data is:

$$L = e^{-m(x_n)} \prod_{k=1}^n m'(x_k) \quad (9)$$

$$L = e^{-\alpha(1 + [1 + (\frac{x_n}{\sigma})]^{-\alpha})} \prod_{k=1}^n a \frac{\alpha}{\sigma} \left[ 1 + \frac{x_k}{\sigma} \right]^{-(\alpha+1)}$$

$$\begin{aligned} \log L &= -\alpha \left\{ 1 + \left[ 1 + \left( \frac{x_n}{\sigma} \right) \right]^{-\alpha} \right\} + n \log a + \\ n \log \alpha - n \log \sigma &+ \sum_{i=1}^n \log \left[ 1 + \frac{x_i}{\sigma} \right]^{-(\alpha+1)} \end{aligned} \quad (10)$$

Then, the log likelihood equations to estimate the unknown parameters  $a, \alpha$  and  $\sigma$  are given by:

$$\begin{aligned} \frac{\partial \log L}{\partial a} &= 0 \\ \hat{\alpha} &= \frac{n}{1 - [1 + (\frac{x_n}{\sigma})]^{-\alpha}} \end{aligned} \quad (11)$$

$$\begin{aligned} \frac{\partial \log L}{\partial \alpha} &= 0 \\ \frac{\alpha \log [1 + \frac{x_n}{\sigma}]}{(1 + \frac{x_n}{\sigma})} + \frac{n}{\alpha} + \sum_{i=1}^n \log [1 + \frac{x_i}{\sigma}] &= 0 \end{aligned} \quad (12)$$

$$\begin{aligned} \frac{\partial \log L}{\partial \sigma} &= 0 \\ \frac{x_n a \alpha}{\sigma^2} (1 + \frac{x_n}{\sigma})^{-(\alpha+1)} - \frac{n}{\sigma} + \frac{\alpha+1}{\alpha^2} \sum_{i=1}^n \log [1 + \frac{x_i}{\sigma}] &= 0 \end{aligned} \quad (13)$$

The Eq. 11-13 are to be solved iteratively. When  $\alpha$  is assumed to be known only one equation that of  $\sigma$  has to be solved by numerical methods to proceed for further evaluation of reliability measures. In order to overcome, the iterative technique for solving  $\sigma$ , one may go for the method of modified ML procedure with the following approximations:

$$\frac{Z_i}{1+Z_i} \equiv \gamma_i + \delta_i Z_i, \quad i = 1, 2, 3, \dots, n-1$$

$$\frac{Z_n}{(1+Z_n)^{\alpha-1}} \equiv \gamma_n + \delta_n Z_n$$

Where;

$$Z_i = \frac{x_i}{\sigma}, \quad i = 1, 2, 3, \dots, n$$

The log likelihood Eq. 13 for  $\sigma$  can be written as:

$$\frac{1}{\alpha} \left\{ a \alpha \frac{Z_n}{(1+Z_n)^{\alpha-1}} - n + \alpha(\alpha+1) \sum_{i=1}^{n-1} \frac{Z_i}{1+Z_i} \right\} = 0$$

With the suggested approximation it becomes:

$$\frac{\partial \log L}{\partial \alpha} \equiv \frac{1}{\sigma} \left\{ a \alpha (\gamma_n + \delta_n Z_n) - n + (\alpha+1) \sum_{i=1}^{n-1} (\gamma_i + \delta_i Z_i) \right\} = 0 \quad (14)$$

We can solve Eq. 14 for  $\sigma$  to get MMLE. The corresponding estimate of  $a$  can be obtained by the Eq. 11. An estimator of software reliability can also be obtained from Eq. 7.

## APPLICATIONS

The procedures narrated and derived are explained by software failure data taken from Jelinsky and Moranda (1972). The data are originally from the U.S. Navy Fleet computer programming centre and consist of the errors in the development of software for the real-time, multi computer complex which forms the core of the Naval Tactical Data System (NTDS).

The NTDS software consisted of some 38 different modules. Each module is supposed to follow 3 stages; the production (development) phase, the test phase and the user phase.

The data are based on the trouble reports or software anomaly reports for one of the larger modules, denoted as A-module. The times (days) between software failures and additional information for this module are shown in Table 1.

Total 26 software errors are found during production phase and 5 additional errors during the test phase. One error is observed during the user phase and 2 more errors are noticed in a subsequent test phase indicating that a network of the module has taken place after the user error is found. The 26 data points are considered as an ordered random sample of size 26 supposed to have come from a

Table 1: NTDS data

1	2	3	1	2	3
<b>Production check out phase</b>			<b>Test phase</b>		
1	9	9	27	87	337
2	12	21	28	47	384
3	11	32	29	12	396
4	4	36	30	9	405
5	7	43	31	135	540
6	2	45	<b>User phase</b>		
7	5	50	32	258	798
8	8	58	<b>Test phase</b>		
9	5	63	33	16	814
10	7	70	34	35	849
11	1	71			
12	6	77			
13	1	78			
14	9	87			
15	4	91			
16	1	92			
17	3	95			
18	3	98			
19	6	104			
20	1	105			
21	11	116			
22	33	149			
23	7	156			
24	91	247			
25	2	249			
26	1	250			

1: Error Number N, 2: Time between errors  $S_k$  days, 3: Cumulative time  $x_n = \sum S_k$  days

Pareto distribution with parameters  $a$ ,  $\sigma$ . This supposition is verified using the q-q plot-correlation coefficient method at  $a = 2, 3, 4$ . It is found that Pareto distribution first reasonably well to the present data. Hence, the log likelihood equation of  $a$  and  $\sigma$  are solved simultaneously by Newton-Raphson method at  $a = 2, 3, 4$  in succession at  $a = 2, 3, 4$ . The ML estimates so obtained are at:

$a$	$\alpha$	$\sigma$
49.653079	2	1.337628
52.695625	3	1.052718
123.032906	4	1.321060

The value of  $L$ , the likelihood function is maximum at the triplet:

$$\hat{a} = 49.653079, \quad \hat{\alpha} = 1.337628, \quad \hat{\sigma} = 269.013288$$

Hence, we may accept the 3 values as ML estimates of  $a$ ,  $\alpha$ ,  $\sigma$ . The estimator of the reliability function from Eq. 7 at any time  $x$  beyond 250 days is given by:

$$Rs_k / X_{k-1}(s/x) = e^{-[m(x+s)-m(s)]}$$

$$Rs_{27} / X_{26}(250/50) = e^{-[m(50+250)-m(250)]} = 0.091961$$

## COMPARISON

For the NTDS data Goel and Okumoto (1979) have obtained ML estimates of  $a$  and  $\sigma$ , respectively as:

$$\hat{a} = 33.99 \text{ and } \hat{\sigma} = 172.71157$$

For the same data with the SRGM (Pareto approach):

$$\hat{a} = 49.653079, \hat{\sigma} = 269.013288$$

A choice between these two models for the data under consideration can be made with reference to a number of criteria. One of them is that model which gives maximum joint probability with the data and the corresponding estimates. In other words that model which gives more value for the likelihood function can be decided as a better fit. We evaluated the values of likelihood functions under Goel and Okumoto (1979), NHPP and its estimates, Pareto structure and its estimates. These are as follows: Natural log of Goel and Okumoto (1979) structure at the sample and the associated estimates is:

$$L\left(\frac{1}{a, \sigma/S}\right) = n \ln a + n \ln b - a(1 - e^{-bs_n}) - b \sum_{k=1}^n S_k = -142.6973007$$

For the same structure, the value for the proposed model is -72.295455. This shows that the present model is a better fit than that of Goel and Okumoto (1979).

## CONCLUSION

The researchers proposed an SRGM through NHPP based on Pareto type mean value function and suggested a modified ML estimator of its parameters to assess software reliability given a software failure data. Its

goodness of fit for a live data is compared. For the same data with a popular SRGM namely Goel and Okumoto (1979) with the help of likelihood functions, we found that the proposed model has more likelihood value than that of Goel and Okumoto (1979) claiming to be a better fit.

## REFERNECES

- Goel, A.L. and K. Okumoto, 1979. Time-dependent error-detection rate model for software reliability and other performance measures. *Trans. Reliability*, 28: 206-211.
- Hossain, S.A. and R.C. Dahiya, 1993. Estimating the parameters of a non-homogeneous poisson-process model for software reliability. *IEEE Trans. Reliability*, 42: 604-612.
- Jelinsky, Z. and P. Moranda, 1972. Software Reliability Research. In: *Statistica Computer Performance Evaluation*, Freiburger, W. (Ed.). Academic Press, New York, ISBN: 981-02-0640-2, pp: 465-484.
- Johnson, N.L., S. Kotz and N. Balakrishna, 2004. *Continuous Univariate Distributions*. John Wiley and Sons, New York.
- Littlewood, B., 1981. Stochastic reliability-growth: A model for fault-removal in computer-programs and hardware-design. *IEEE Trans. Reliability*, 30: 313-320.
- Musa, J.D., A. Iannino and K. Okumoto, 1987. *Software Reliability: Measurement, Prediction and Application*. McGraw-Hill, New York.
- Pham, H.A., 2005. General logistic software reliability growth model. *OPSEARCH*, 42: 322-331.
- Wood, A., 1996. Predicting software reliability. *IEEE Trans. Software Eng.*, 29: 69-77.
- Yamada, S., H. Ohtera and H. Narihisa, 1986. Software reliability growth models with testing-effort. *IEEE Trans. Reliability*, 35: 19-23.