Burr Type XII Software Reliability Growth Model

R. Satya Prasad, Ph.D
Associate Professor
Department of CSE
Acharya Nagarjuna University
Guntur, India

K.V. Murali Mohan
Associate Professor
Department of ECE
H.M. Institute of Tech. and Science
Hyderabad, India

G. Sridevi
Associate Professor
Department of CSE
Nimra Women's College of Engg
Vijayawada, India

ABSTRACT

Software Reliability Growth model (SRGM) is a mathematical model of how the software reliability improves as faults are detected and repaired. The development of many SRGMs over the last several decades have resulted in the improvement of software facilitating many engineers and managers in tracking and measuring the growth of reliability. This paper proposes Burr type XII based Software Reliability growth model with time domain data. The unknown parameters of the model are estimated using the maximum likelihood (ML) estimation method. Reliability of a software system using Burr type XII distribution, which is based on Non-Homogenous Poisson process (NHPP), is presented through estimation procedures. The performance of the SRGM is judged by its ability to fit the software failure data. How good does a mathematical model fit to the data is also being calculated. To access the performance of the considered SRGM, we have carried out the parameter estimation on the real software failure datasets.

General Terms

Software failure data, Mean value function.

Keywords

Software Reliability, Burr type XII distribution, NHPP, ML Estimation

1. INTRODUCTION

Software reliability is defined as the probability of failure free software operation for a specified period of time in a specified environment (Lyu, 1996) (Musa et al.. 1987).SRGM is a mathematical model of how the software reliability improves as faults are detected and required (Quadri and Ahmad, 2010). Among all SRGMs developed so far a large family of stochastic reliability models based on a Non-Homogeneous Poisson Process known as NHPP reliability model has been widely used. Software Reliability is the most dynamic quality characteristic which can measure and predict the operational quality of the software system during its intended life cycle. To identify and eliminate human errors in software development process and also to improve software reliability, the Statistical Process Control concepts and methods are the best choice. If the selected model does not fit the collected software testing data relatively well. We would expect a low prediction ability of this model and the decision makings based on the analysis of this model would be far from what is considered to be optimal decision (xie et al., 2001). This paper presents a method for model validation.

2. RELATED RESEARCH

This section presents the theory that underlies the proposed distributions and maximum likelihood estimation for complete data. If 't' is a continuous random variable with pdf: $f(t;\theta_1,\theta_2,...,\theta_k)$. Where $\theta_1,\theta_2,...,\theta_k$ are k unknown constant parameters which need to be estimated, and cdf: $F(t)$. Where, the mathematical relationship between the pdf and cdf is given by: $f(t)=\frac{d(F(t))}{dt}$. Let 'a' denote the expected number of faults that would be detected given infinite testing time in case of finite failure NHPP models. Then, the mean value function of the finite failure NHPP models can be written as: $m(t)=aF(t)$. Where, $F(t)$ is a cumulative distributive function. The failure intensity function $\lambda(t)$ in case of the finite failure NHPP models is given by: $\lambda(t)=aF^{'}(t)$ [8].

2.1 NHPP Model

There are numerous software reliability growth models available for use according to probabilistic assumptions. The Non Homogenous Poisson Process (NHPP) based software reliability growth models are proved to be quite successful in practical software reliability engineering [4]. Model parameters can be estimated by using maximum Likelihood Estimate (MLE). NHPP model formulation is described in the following lines.

A software system is subjected 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', where t is the failure intensity function, which is proportional to the residual fault content.

Let $m(t)$ represent the expected number of software failures by time 's'. The mean value function $m(t)$ is finite valued, non-decreasing, non-negative and bounded with the boundary conditions.

$$m(t)=\begin{cases}0, & t=0\\ a, & t\to\infty\end{cases}$$

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 parameters $m(t)$ i.e.,

$$P\{N(t)=n\}=\frac{[m(t)]^n.e^{-m(t)}}{n!}, n=0,1,2\ldots\infty$$

Then $N(t)$ is called an NHPP. Thus the stochastic behaviour of software failure phenomena can be described through the $N(t)$ process. Various time domain models have appeared in the literature that describes the stochastic failure process by an NHPP which differ in the mean value function $m(t)$.

2.2. Proposed Model Description

In this paper, we propose to monitor software quality using SPC based on Burr Type XII distribution model. The Burr distribution has a flexible shape and controllable scale and location which makes it appealing to fit to data. It is frequently used to model insurance claim sizes [5]. The mean value function and intensity function of Burr Type XII NHPP model are as follows.

The Cumulative distributive function (CDF) is given by

$$m(t)=\int_0^1 \lambda(t)\,dt=a\left[1-\left(1+t^c\right)^{-b}\right] \qquad (2.2.1)$$

$$=a\,F(t)$$

The Probability Density Function (PDF) of Burr XII distribution are given, respectively by

$$\lambda(t)=a\left(cbt^{c-1}\Big/\left(1+t^c\right)^{b+1}\right)=a\ f(t)$$

Where t>0, a>0, b>0 and c>0 denote the expected number of faults that would be detached given infinite testing time in case of finite failure NHPP models. In order to have an assessment of the software reliability, a, b and c are unknown parameters and estimated by using Newton Raphson method. Expressions are now delivered for estimating 'a', 'b' and 'c' for the Burr type XII model.

$$p\{N(t)=n\}=\frac{[m(t)]^n e^{-m(t)}}{n!}$$

We conduct an experiment and obtain N independent observations $t_1,t_2,\ldots,t_n$. The likelihood function for time domain data [4] is given by

$$\lim_{n\to\infty} p\{N(t)=n\}=\frac{e^{-a}.a^n}{n!}$$

This is also a Poisson model with mean 'a'.

Let N (t) be the number of errors remaining in the system at time't'.

$$N(t)=N(\infty)-N(t)$$

$$E[N(t)]\ \ =E[N(\infty)]-E[N(t)]$$

$$=a-m(t)$$

$$=a-a\left[1-(1+t^c)^{-b}\right]$$

$$=a(1+t^c)^{-b}$$

Let S_k be the time between $(k-1)^{th}$ and k^{th} failure of the software product. Let X_k be the time up to the k^{th} failure. Let us find out the probability that time between $(k-1)^{th}$ and k^{th} failures, i.e., S_k exceeds a real number 's' given that the total time up to the $(k-1)^{th}$ failure is equal to x.

i.e., $P\left[S_k>\frac{s}{X_{k-1}}=x\right]$

$$R\ S_k/X_{k-1}(s/x)=\ e^{-[m(x+s)-m(s)]} \qquad (2.2.2)$$

This Expression is called Software Reliability.

3. ILLUSTRATING THE MLE

3.1. Parameter Estimation based on Time Domain Data

In this section we develop expressions to estimate the parameters of the Burr type XII model based on time domain data. Parameter estimation is of primary importance in software reliability prediction.

A set of failure data is usually collected in one of two common ways, time domain data and time domain data. In this paper parameters are estimated from the time domain data.

The mean value function of Burr type XII model is given by

$$m(t)=a\left[1-\left(1+t^c\right)^{-b}\right],\ \ t\ge 0 \qquad (3.1.1)$$

$$L=\prod_{i=1}^{N} abct_i^{c-1}/(1+t_i^c)^{b+1}.e^{-a\left[1-(1+t^c)^{-b}\right]}$$

$$LogL=-a+a(1+t^c)^{-b}+\sum_{i=1}^{n}\left[Log\,a+Log\,b+Log\,c+(c-1)\log t_i-(b+1)\log(1+t_i^c)\right] \quad (3.1.2)$$

Taking the Partial derivative with respect to 'a' and equating to '0'.

i.e., $\frac{\partial Log\ L}{\partial a} = 0$

$$\therefore\ a = \frac{n(1+t^c)^b}{(1+t^c)^b - 1} \qquad (3.1.3)$$

The parameter 'b' is estimated by iterative Newton Raphson Method using

$b_{n+1} = b_n - \frac{g(b)}{g'(b)}$, Where $g(b) and\ \ g'(b)$ are expressed as follows.

$$g(b) = \frac{\partial LogL}{\partial b} = 0$$

$$\frac{\partial Log\ L}{\partial b} = g(b) = \frac{n\log\left(\frac{1}{t+1}\right)}{(t+1)^b - 1} + \frac{n}{b} - \sum_{i=1}^{n}\log(\mathrm{t}_i + 1) \qquad (3.1.4)$$

$$g'(b) = \frac{\partial^2 LogL}{\partial b^2} = 0$$

$$\frac{\partial^2 LogL}{\partial b^2} = g'(b) = -n\left[\log\left(\frac{1}{t+1}\right)\left\{\frac{(t+1)^b\log(t+1)}{\left[(t+1)^b - 1\right]^2}\right\} + \frac{1}{b^2}\right] \qquad (3.1.5)$$

The parameter 'c' is estimated by iterative Newton Raphson Method using

$c_{n+1} = c_n \ - \frac{g(c_n)}{g'(c_n)}$

Where $g(c)\ and\ g'(c)$ are expressed as follows.

$g(c) = \frac{\partial LogL}{\partial c} = 0$

$$\frac{\partial LogL}{\partial c} = g(c) = \frac{-n}{(1+t^c)}\log(t) + \frac{n}{c} - \sum_{i=1}^{n} 2\log(t)\frac{t_i^c}{(1+t_i^c)} + \sum_{i=1}^{n}\log t_i \qquad (3.1.6)$$

$g'(c) = \frac{\partial^2 LogL}{\partial c^2} = 0$

$$\frac{\partial^2 Log\ L}{\partial c^2} = g'(c) = \frac{nt^c\log t}{(1+t^c)^2}\log t - \frac{n}{c^2} - 2\log t.\sum_{i=1}^{n} t_i^c\log t_i\left\{\frac{1}{(1+t_i^c)^2}\right\} \qquad (3.1.7)$$

4. DATA ANALYSIS

The set of software errors analyzed here is borrowed from software development project as published in Pham (2006) [8].

Table 1: Naval Tactical Data System Software Dataset

Failure Number (n)	Time Between Failures (x_k) days	Cumulative Time (S_n)
1	9	9
2	12	21
3	11	32
4	4	36
5	7	43
6	2	45
7	5	50
8	8	58
9	5	63
10	7	70
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
Test Phase		
27	87	337
28	47	384
29	12	396
30	9	405
31	135	540
User Phase		
32	258	798
Test Phase		
33	16	814
34	35	849

Solving equations by Newton Raphson Method for the NTDS test data, the iterative solutions for MLEs of a, b and c are

$$a = 26.105273$$

$$b = 0.998899$$

$$c = 0.998903$$

Table 2 : Parameters Estimated through MLE

Dataset	Number of samples	Estimated Parameters		
		a	b	c
NTDS	26	26.105273	0.998899	0.998903
Xie	30	30.040800	0.999825	0.999619
AT & T	22	22.032465	0.999859	0.999611
IBM	15	15.051045	0.999530	0.999196
SONATA	30	30.016391	0.999958	0.999920

Hence, we may accept these three values as MLSs of a, b, c. The estimator of the reliability function from the equation (2.2.2) at any time x beyond 250 hours is given by

$$R\, S_k / X_{k-1}(s/x) = e^{-[m(x+s)-m(s)]}$$

$$R s_7 / x_6 (250/50) = e^{-[m(50+250)-m(250)]}$$

$$= e^{-[m(300)-m(250)]}$$

$$= 0.98269972$$

5. METHOD OF PERFORMANCE ANALYSIS

The performance of SRGM is judged by its ability to fit the software failure data. The term goodness of fit denotes the question of "How good does a mathematical model fit to the data?". In order to validate the model under study and to assess its performance, experiments on a set of actual software failure data have been performed. The considered model fits more to the dataset whose Log Likelihood is most negative. The application of the considered distribution function and its Log Likelihood on different datasets collected from real world failure data is given below.

Table 3 : Log Likelihood on different datasets

Data Set	Log L (MLE)	Reliability (t_n+x)
NTDS	- 38.477204	0.98269972
Xie	- 44.402007	0.99616300
AT & T	- 18.949300	0.99573348
IBM	- 27.365034	0.98841341
SONATA	- 46.991596	0.99700207

6. CONCLUSION

To validate the proposed approach, the parameter estimation is carried out on the data sets collected from (Xie et al., 2002; Pham, 2006; Ashoka, 2010).Out of the data sets that were collected, the model under consideration best fits the data of SONATA using MLE approach. Since, it is having the highest negative value for the log likelihood. The reliability of all the data sets are given in Table 3.The reliability of the model over SONATA data is high among the data sets which were considered.

7. REFERENCES

[1] Kimura, M., Yamada, S., Osaki, S., 1995. "Statistical Software reliability prediction and its applicability based on mean time between failures". Mathematical and Computer Modelling Volume 22, Issues 10-12, Pages 149-155.

[2] Ashoka, M., (2010), "Sonata Software Limited" data set, Bangalore.

[3] Lyu, M.R., (1996), "Handbook of Software Reliability Engineering", MCGraw-Hill, New York.

[4] Musa, J.D., Iannino, A., Okumoto, k., 1987. "Software Reliability: Measurement Prediction Application". McGraw -Hill, New York.

[5] Hee-cheul Kim., "Assessing Software Reliability based on NHPP using SPC", International Journal of Software Engineering and its Applications, vol.7,No.6 (2013), pp.61-70.

[6] Pham. H., 2003. "Handbook of Reliability Engineering", Springer.

[7] Pham. H., 2006. "System software reliability", Springer.

[8] Swapna S. Gokhale and Kishore S.Trivedi, 1998. "Log-Logistic Software Reliability Growth Model". The 3rd IEEE International Symposium on High-Assurance Systems Engineering. IEEE Computer Society.

[9] Xie, M., Goh. T.N., Ranjan.P., (2002) "Some effective control chart procedures for reliability monitoring" – Reliability engineering and System Safety 77 143 -150 ¸2002.

[10] Satya Prasad, R., "Half logistic distribution for software reliability growth model", Ph.D thesis, 2007.

[11] Goel, A.L., Okumoto, K., 1979. Time-dependent error detection rate model for software reliability and other performance measures. IEEE Trans. Reliab. R-28, 206-211.

[12] Satya Prasad, R., Shaheen., Krishna Mohan, G., "Two step Approach for Software reliability: HLSRGM", IJCTT-Volume 4 Issue 10-Oct 2013.

[13] Wood, A., (1996), "Software Reliability Growth models", Tandem Computers, Technical report 96.1.

[14] Quadri, S.M.K and Ahmad, N., (2010), "Software Reliability Growth Modelling with new modified Weibull testing-effort and optimal release policy", International Journal of Computer Applications, Vol.6, No.12.

[15] Xie, M., Yang, B. and Gaudoin, O. (2001), "Regression goodness-of-fit Test for Software reliability model validation ", ISSRE and Chillarege Corp.