IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. IV (Jan – Feb. 2015), PP 49-54
www.iosrjournals.org

Burr Type III Software Reliability Growth Model

Ch.Smitha Chowdary[1], Dr.R.Satya Prasad[2], K.Sobhana[3]
[1]Research Scholar, Dept. of Computer Science, Krishna University, Machilipatnam, Andhra Pradesh. INDIA-521001
[2]Associate Professor, Dept. of Computer Science & Engineering Acharya Nagarjuna University, Nagarjuna Nagar- INDIA-522510
[3]Research Scholar, Dept. of Computer Science, Krishna University, Machilipatnam, Andhra Pradesh. INDIA-521001

Abstract : *Software Reliability Growth Model (SRGM) is used to assess software reliability quantitatively for tracking and measuring the growth of reliability. The potentiality of SRGM is judged by its capability to fit the software failure data. In this paper we propose Burr type III software reliability growth model based on Non Homogeneous Poisson Process (NHPP) with time domain data. The Maximum Likelihood (ML) estimation method is used for finding unknown parameters in the model on ungrouped data. How good does a mathematical model fit to the data is also being calculated. To assess the performance of the considered SRGM, we have carried out the parameter estimation on real software failure data sets. We also present an analysis of goodness of fit and reliability for given failure data sets.*

Keywords: *Burr type III, Goodness of fit, NHPP, ML estimation, Software Reliability, Time domain data.*

I. Introduction

Software reliability is the probability of failure-free operation of software in a specified environment during specified duration [1][2][3]. Over the decades, different statistical models have been discussed for assessing the software reliability where Wood (1996), Pham (2005), Goel & Okumoto (1979), Satya Prasad (2007) and Satya Prasad & Geetha Rani (2011) are some examples [4][5][6][7][8]. The models applicable for the assessment of software reliability are called Software Reliability Growth Models (SRGMs). SRGM is a mathematical model of how software reliability improves as faults are detected and repaired [9]. One of the important classes of SRGM that has been studied widely is Non Homogeneous Poisson Process (NHPP), which forms one of the main classes of the existing SRGMs due to its mathematical traceability and wide applicability. The NHPP based software reliability growth models have been proved quite successful in practical software reliability engineering [3].

Software reliability can be estimated after the determination of the mean value function. To determine this mean value function, model parameters can be estimated by using Maximum Likelihood Estimation (MLE). These Parameter values can be obtained by using Newton-Raphson Method.

The success of mathematical modeling approach to reliability evaluation heavily depends upon quality of failure data collected and its goodness of fit. If the selected model does not fit relatively well into the collected software testing data, we may expect a low prediction ability of this model and the decision makings based on the analysis of this model may be far from what is considered to be optimal decision [10].

This paper presents Burr type III model to analyze the reliability of a software system using time domain data. The layout of the paper is as follows: Section 2 gives the details of formulation and interpretation of the model for the underlying NHPP. Section 3 describes the background of Burr type III model. Section 4 discusses parameter estimation of Burr type III model based on time domain data. Section 5 describes the techniques used for software failure data analysis for live data. Section 6 gives the performance analysis of the presented model and Section 7 contains the conclusion.

II. NHPP Model

Software reliability models can be classified according to probabilistic assumptions based on the type of failure process. First one is, when a Markov process represents the failure process, the resultant model is Markovian Model. Second one is fault counting model that describes the failure phenomenon by stochastic process like Homogeneous Poisson Process (HPP), Non Homogeneous Poisson Process (NHPP) and Compound Poisson Process (CPP) etc. Most of the failure count models are based upon NHPP and are described in the following lines.

A software system is subject to failures at random times which are caused by errors present in the system. Suppose $\{N(t), t > 0\}$ is a counting process, then the cumulative number of failures are represented by time 't'. As there were no failures at t=0 we have

$$N(0) = 0$$

The assumption is, that the number of software failures during non overlapping time intervals will not affect each other. In other way, for any finite collection of times t1<t2<…. <tn the 'n' random variables N(t1), {N(t2)-N(t1)}, ….. {N(tn) - N(tn-1)} are independent. This implies that the counting process {N(t), t>0} has independent increments. Suppose m(t) represents the expected number of software failures by time 't', then the expected number of errors remaining in the system at any time is finite, hence m(t) is a bounded, non decreasing function of 't' with the following boundary conditions.

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

Here the above mentioned 'a' is meant for expected number of software errors to be eventually detected. Let N(t) be known to have a Poisson probability mass function with parameters m(t) i.e.

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

Then the stochastic behavior of software failure phenomenon can be described through the N(t) process. In this paper we consider m(t) as given by

$$m(t) = a\left[1 + t^{-c}\right]^{-b} \text{ ---------------- (2.1)}$$

Where [m(t)/a] is the cumulative distribution function of Burr type III distribution for the present choice.

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

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

This is also a Poisson model with mean 'a'. Let the number of errors remaining in the system at time't' be N(t).

$$\mathrm{N(t)} = \mathrm{N}(\infty) - \mathrm{N(t)}$$

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

$$= a - m(t)$$

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

Suppose S_k be the time between $(k-1)^{th}$ and k^{th} failure of the software product and X_k be the time up to the k^{th} failure, then to find out the probability time between $(k-1)^{th}$ and k^{th} failures the software reliability function is given below.

$$RS_k / X_{k-1}(s/x) = e^{-[m(x+s) - m(s)]} \text{ --------- (2.2)}$$

This expression is referred as Software Reliability.

III. Background Theory

This section presents the theory that underlines Burr Type III based NHPP model. The Burr type XII uses a wide range of skewness and kurtosis, which may be used to fit any given set of unimodal data [11]. The 'reciprocal Burr' (Burr Type III) covers a wide region that includes the region covered by Burr Type XII. Burr has suggested a number of forms of cumulative distribution functions (cdf) that would be useful for fitting data [12]. The Burr Type XII distribution F(x) is given as

$$F(x) = 1-(1+x^c)^{-b}, x>0, c>0, b>0. \text{ ------ (3.1)}$$

Here both' c' and' b' are shape parameters.

Let X be the random variable with cdf given by equation (3.1) and consider the transformation t=1/X.

$$F(t)=(1+t^{-c})^{-b} \qquad \text{--------- (3.2)}$$

Which is one of the many forms of distribution functions (Burr Type III) given by Burr.

IV. Parameters Of Burr Type III Based Time Domain Data

Burr [12] had introduced twelve different forms of cumulative distribution functions for modeling data. The task of building a mathematical model is incomplete until the unknown parameters i.e. the model parameters are estimated and validated on actual software failure data sets. In this section we develop expressions to estimate the parameters of the Burr type III model based on time domain data. Parameter estimation is given primary importance for software reliability prediction. Parameter estimation can be achieved by applying a technique of MLE which is the most important and widely used estimation technique. A set of failure data is usually collected in one of two common ways, time domain data and interval domain data. Here the failure data is collected through time domain data.

The mean value function of Burr type III model as given in equation (2.1) is

$$m(t) = a\left[1+t^{-c}\right]^{-b} \quad t>0 \text{ a, b, c} > 0$$

To assess the software reliability, 'a', 'b' and 'c' are to be known or else they can be estimated from software failure data. For estimating 'a', 'b' and 'c' for the Burr type III model expressions are derived as mentioned below. Assuming that the data are given for the occurrence times of the failures or the times of successive failures, i.e., the realization of random variables T_j for j = 1, 2,..n. Given that the data provide n successive times of observed failures T_j for 0 $t_1 \leq t_2 \leq \ldots t_n$, we can convert these data into the time between failures x_i where $x_i = t_i - t_{i-1}$ for i = 1, 2,…, n. Given the recorded data on the time of failures, the log likelihood function (LLF) takes on the following form:

$$LLF = \sum_{i=1}^{n} \log\left[\lambda(t_i)\right] - m(t_n) \quad \text{---------} \ (4.1)$$

$$LogL = \sum_{i=1}^{n} \log\left[\frac{abc}{t_i^{c+1}(1+t_i^{-c})^{b+1}}\right] - \frac{a}{\left[1+t_n^{-c}\right]^{b}} \quad \text{--------} \ (4.2)$$

$$LogL = \frac{-a}{\left(1+t_n^{-c}\right)^{b}} + \sum_{i=1}^{n}\left[\log a + \log b + \log c - (c+1)\log t_i - (b+1)\log\left(1+t_i^{-c}\right)\right] \quad \text{-----} \ (4.3)$$

Accordingly parameters 'a', 'b' and 'c' would be solutions of the equations

$$\frac{\partial LogL}{\partial a} = 0$$

$$\Rightarrow a = n\left(1+t_n^{-c}\right)^{b} \quad \text{---------} \ (4.4)$$

$$\frac{\partial LogL}{\partial b} = 0$$

$$\Rightarrow b = \frac{n}{\sum_{i=1}^{n} \log\left(1+t_i^{-1}\right) - n\log\left(1+t_n^{-1}\right)} \quad \text{-----------} \ (4.5)$$

The parameter 'c' is estimated by iterative Newton-Raphson Method using $c_{i+1} = c_i - \frac{g(c_i)}{g'(c_i)}$ where g(c) and g'(c) are expressed as follows.

$$\frac{\partial LogL}{\partial c} = 0$$

$$\Rightarrow g(c) = \frac{-n\log(t_n)}{1+t_n^{c}} + \frac{n}{c} + \sum_{i=1}^{n} \log t_i\left[-1 + \frac{2}{1+t_i^{c}}\right] \quad \text{------} \ (4.6)$$

$$\frac{\partial^2 LogL}{\partial c^2} = 0$$

$$\Rightarrow g'(c) = \frac{n(\log t_n)^2 t_n^c}{(t_n^c + 1)^2} - \frac{n}{c^2} - \sum_{i=1}^{n} \frac{2t_i^c (\log t_i)^2}{(t_i^c + 1)^2} \quad \text{------------ (4.7)}$$

The value of 'c' in the above equations (4.6) (4.7) can be obtained using Newton-Raphson iterative method. Solving the above equation yields the point estimate of the parameter 'c'.

V. Data Validity Analysis

The set of software errors analyzed here is borrowed from software development project as published in Pham [13, 14]. Data set is truncated into different proportions and used for estimating the parameters of the proposed basic discrete time model. Table 1 shows the time between failures for different software products presenting as per the size of the data set.

NTDS Data

The data set consists of 26 failures in 250 days. During the production phase 26 software errors are found and during the test phase five additional errors are found. During the user phase one error is observed and two 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. In this paper, a numerical conversion of data (Failure Time (hours)*0.01) is done in order to facilitate the parameter estimation [15] [16] [17].

Table-1: NTDS Data Set

Failure Number n	Time between Failures S_k days	Cumulative Time $X_n = S_k$ days	Failure Time(hours)*0.01
Production (Checkout) Phase			
1	9	9	0.09
2	12	21	0.21
3	11	32	0.32
4	4	36	0.36
5	7	43	0.43
6	2	45	0.45
7	5	50	0.5
8	8	58	0.58
9	5	63	0.63
10	7	70	0.7
11	1	71	0.71
12	6	77	0.77
13	1	78	0.78
14	9	87	0.87
15	4	91	0.91
16	1	92	0.92
17	3	95	0.95
18	3	98	0.98
19	6	104	1.04
20	1	105	1.05
21	11	116	1.16
22	33	149	1.49
23	7	156	1.56
24	91	247	2.47
25	2	249	2.49
26	1	250	2.5
Test Phase			
27	87	337	3.37
28	47	384	3.84
29	12	396	3.96
30	9	405	4.05
31	135	540	5.4
User Phase			
32	258	798	7.98
Test Phase			
33	16	814	8.14
34	35	849	8.49

Solving equations in Section 3 by Newton-Raphson Method (N-R) method for the NTDS software failure data, the iterative solutions for MLEs of a, b and c are as below.

$$\hat{a} = 34.465706$$

$$\hat{b} = 1.763647$$

$$\hat{c} = 1.810222$$

Table-2: Parameters Estimated through MLE

Data set (no)	Number of samples	Estimated Parameter		
		A	b	C
NTDS	26	34.465706	1.763647	1.810222
AT&T	22	26.839829	1.658692	1.000000
SONATA	30	79.831359	6.742810	0.602440
XIE	30	33.310426	2.270095	1.371974
IBM	15	20.624785	1.711630	1.447815

Here, these three values can be accepted as MLEs of 'a', 'b' and 'c'. The estimator of the reliability function from the equation (2.2) at any time x beyond 250 days is given by

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

$$= e^{-[m(50+250)-m(250)]}$$

$$=0.999221$$

VI. Performance Analysis For Goodness Of Fit

The potentiality of SRGM is judged by its capability to fit the software failure data, where the term goodness of fit denotes the question of "How good does a mathematical model fit to the data?" Experiments on a set of actual software failure data have been performed to validate the model under study and to assess its performance. The considered model fits more to the data set 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 in table 3.

Table-3: Log likelihood on different data sets

Data set (no)	Log L (MLE)	Reliability tn+50 (MLE)
NTDS	-168.070414	0.999221
AT&T	-160.264155	0.995543
SONATA	-218.086332	0.917342
XIE	-256.630664	0.999247
IBM	-109.246261	0.998116

VII. Conclusion

In this paper we propose a Burr type III software reliability growth model. This model is useful primarily for estimating and monitoring software reliability that is viewed as a measure of software quality. Equations are developed to obtain the maximum likelihood estimates of the parameters based on time domain data. The proposed discrete time models have been validated and evaluated on actual software failure data cited from real software development projects and compared with existing discrete time NHPP based model. The results are encouraging in terms of goodness of fit and predictive validity due to their applicability and flexibility. To validate the proposed approach, the parameter estimation is carried out on the data sets collected [18][19]. The data set Xie has the best fit among all datasets as 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 Xie data is the highest among the data sets which are considered.

References

[1]. Musa J.D, Software Reliability Engineering MCGraw-Hill, 1998.

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

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

[4]. Wood. A (1996), "Predicting Software Reliability", IEEE Computer, 2253-2264.

[5]. Pham. H (2005) "A Generalized Logistic Software Reliability Growth Model", Opsearch, Vol.42, No.4, 332-331.MC Graw Hill, New York.

[6]. 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.

[7]. Satya Prasad, R (2007) "Half logistic Software reliability growth model", Ph.D Thesis of ANU, India.

[8]. Satya Prasad, R and Geetha Rani, N (2011), "Pareto type II software reliability growth model", International Journal of Software Engineering, Volume 2, Issue(4) 81-86

[9]. 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.

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

[11]. Tadikamalla, Pandu R. "A Look at the Burr and Related Distributions." International Statistical Review / Revue Internationale de Statistique 48(3)(Dec., 1980): 337–344.

[12]. Burr (1942), "Cumulative Frequency Functions", Annals of Mathematical Statistics, 13, pp. 215-232.

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

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

[15]. N. R. Barraza., "Parameter Estimation for the Compound Poisson Software Reliability Model", International Journal of Software Engineering and Its Applications, http://www.sersc.org/journals/IJSEIA/vol7_no1_2013/11.pdf, vol. 7, no. 1, (2013) January, pp. 137-148.

[16]. Inayat, M. Asim Noor and Z. Inayat, "Parameter Successful Product-based Agile Software Development without Onsite Customer: An Industrial Case Study", International Journal of Software Engineering and Its Applications,http://www.sersc.org/journals/IJSEIA/vol6_no2_2012/1.pdf, vol. 6, no. 2, (2012) April, pp. 1-14.

[17]. Hassan Najadat and Izzat Alsmadi., "Enhance Rule Based Detection for Software Fault Prone Modules", International Journal of Software Engineering and Its Applications, Vol. 6, No.1, pp. 75-86, January (2012)http://www.serc.org/journals/IJSEIA/vol6_no1_2012/6.pdf.

[18]. 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.

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