Pareto Type II Based Software Reliability Growth Model

Dr.R.Satya Prasad *profrsp@gmail.com*
Associate Professor, Dept.of Computer Science & Engg.
Acharya Nagarjuna University,
Nagarjuna Nagar- 520510.
INDIA.

N.Geetha Rani *geetha.neppala@gmail.com*
Associate Professor, Department of Computer Science,
Abhinav Institute of Management & Tech.
Singaryakonda – INDIA

Prof.R.R.L.Kantam *kantam_rrl@rediffmail.com*
Professor, Department of Statistics
Acharya Nagarjuna University,
Nagarjuna Nagar- 520510.
INDIA.

Abstract

The past 4 decades have seen the formulation of several software reliability growth models to predict the reliability and error content of software systems. This paper presents Pareto type II model as a software reliability growth model, together with expressions for various reliability performance measures. Theory of probability, distribution function, probability distributions plays major role in software reliability model building. This paper presents estimation procedures to access reliability of a software system using Pareto distribution, which is based on Non Homogenous Poisson Process (NHPP).

Keywords: Software Reliability, NHPP, Pareto Type II Distribution, Parameter Estimation.

1. INTRODUCTION

Software reliability is the probability of failure free operation of software in a specified environment during specified duration [Musa 1998]. Several models have been proposed during the past 4 decades for accessing reliability of a software system for example Crow and Basu(1988), Goel and Okumoto (1979,1984), Musa(1980), Pham(2005), Ramamurthy and Bastani(1982), Zhang,Teng and Pham(2003), Malaiya, Karunanithi and Verma(1992) and Wood(1996). The objective of such models is to improve software performance. These models are concerned with forecasting future system operability from the failure data collected during the testing phase of a software product. Most of the models assume that the time between failure follows an exponential distribution with parameter that varies with the number of errors remaining in the software system. A software system is a product of human work and is very likely to contain faults. The accuracy of software reliability growth models when validated using the very few available data sets varies significantly and thus despite the existence of numerous models, none of them can be recommended unreservedly to potential users.

This paper presents a Pareto type II model to analyze the reliability of a software system. Our objective is to develop a parsimonious model whose parameters have a physical interpretation and which can yield quantitative measure for software performance assessment. The layout of the paper is as follows: Section 2 describes the development and interpretation of the mean value function for the underlying NHPP. Section 3 discusses parameter estimation of Pareto type II model based on time between failure data. Section 4 describes the techniques used for software failure data analysis for a live data and Section 5 contains conclusions.

2. PARETO MODEL DEVELOPMENT

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

A software system is subject to failures at random times caused by errors present in the system. Let {N(t), t >0} be a counting process representing the cumulative number of failures by time t. Since there are no failures at t=0 we have

$$N(0) = 0$$

It is 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 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.

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 bounded, non decreasing function of 't' with the following boundary conditions.

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

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.

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

then N(t) is called an NHPP. Thus the stochastic behavior of software failure phenomena can be described through the N(t) process. Various time domain models have appeared in the literature (Kantam and Subbarao, 2009) which describe the stochastic failure process by an NHPP which differ in the mean value functions m(t).

In this paper we consider m(t) as given by

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

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

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

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

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

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

$$N(t) = N(\infty) - N(t)$$
$$E[N(t)] = E[N(\infty)] - E[N(t)]$$
$$= a - m(t)$$
$$= a - a\left[1 - \frac{c^b}{(t+c)^b}\right]$$
$$= \frac{ac^b}{(t+c)^b}$$

Let be the time between (k-1)th and kth failure of the software product. Let be the time up to the kth failure. Let us find out the probability that time between (k-1)th and kth failures, i.e. exceeds a real number 's' given that the total time up to the (k-1)th failure is equal to x, i.e. P[> s / = x]

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

This Expression is called Software Reliability.

3. PARAMETER ESTIMATION OF PARETO TYPE II MODEL

In this section we develop expressions to estimate the parameters of the Pareto type II model based on time between failure data. Expressions are now derived for estimating 'a', 'b' and 'c' for the model.

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

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

which represents the time to failure k. Suppose we are given 'n' software failure times say $x_1, x_2, \dots x_n$, there are 'n' time instants at which the first, second, third … nth 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 alternatively separated by error detections and debugging.

The mean value function of Pareto type II model is given by

$$m(t) = a\left[1 - \frac{c^b}{(t+c)^b}\right], \quad t \geq 0 \qquad (3.1)$$

The constants 'a' , 'b' and 'c' which appear in the mean value function and various other expressions are called parameters of the model. In order to have an assessment of the software reliability a, b and c are to be known or they are to be estimated from software failure data. Expressions are now derived for estimating 'a', 'b' and 'c' for the model.

The required likelihood function is given by

$$L = e^{-m(x_n)} . \prod_{i=1}^{n} m'(x_i) \qquad (3.2)$$

values of a, b and c that would maximize L are called maximum likelihood estimators (MLEs) and the method is called maximum likelihood (ML) method of estimation.

$$L = e^{-a\left[1 - \frac{c^b}{(x_n+c)^b}\right]} . \prod_{i=1}^{n} \frac{abc^b}{(x_i+c)^{b+1}} \qquad (3.3)$$

Then the log likelihood equation to estimate the unknown parameters a, b and c are given by

$$\text{LogL} = -a\left[1-\frac{c^b}{(x_n+c)^b}\right]+\sum_{i=1}^{n}\left[\log a+\log b+b\log c-(b+1)\log(x_i+c)\right] \qquad (3.4)$$

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

$$\frac{\partial LogL}{\partial a}=0, \quad \frac{\partial LogL}{\partial b}=0, \quad \frac{\partial^2 LogL}{\partial b^2}=0,$$

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

Substituting the expressions for m(t) (3.1) in the above equations, taking logarithms, differentiating with respect to ‘a’, ‘b’, ‘c’ and equating to zero, after some joint simplications we get

$$a = \frac{n\,(x_n+c)^{\,b}}{(x_n+c)^{\,b} - c^b} \qquad (3.5)$$

$$g(b) = \frac{n\log\frac{1}{x_n+1}}{(x_n+1)^b-1} + \frac{n}{b} - \sum_{i=1}^{n}\log(x_i+1) \qquad (3.6)$$

Second order partial derivative of L with respect to the parameter ‘b’

$$g'(b) = -n\log\frac{1}{x_n+1}\left[\frac{(x_n+1)^{\,b}\,\log(x_n+1)}{[(x_n+1)^b-1]^2}\right] - \frac{n}{b^2} \qquad (3.7)$$

$$g(c) = \frac{n}{x_n+c} + \frac{n}{c} - \sum_{i=1}^{n}\frac{2}{x_i+c} \qquad (3.8)$$

Second order partial derivative of L with respect to the parameter ‘c’

$$g'(c) = -\frac{n}{(x_n+c)^2} - \frac{n}{c^2} + \sum_{i=1}^{n}\frac{2}{(x_i+c)^2} \qquad (3.9)$$

The values of ‘b’ and ‘c’ in the above equations can be obtained using Newton Raphson Method. Solving the above equations simultaneously, yields the point estimates of the parameters a, b and c. These equations are to be solved iteratively and their solutions in turn when substituted in the log likelihood equation of ‘a’ would give analytical solution for the MLE of ‘a’. However when ‘b’ is assumed to be known only one equation that of ‘c’ has to be solved by numerical methods to proceed for further evaluation of reliability measures.

4. NTDS SOFTWARE FAILURE DATA ANALYSIS

In this Section, we present the analysis of NTDS software failure data, taken from Jelinski and Mornda(1972). The data are originally from the U.S. Navy Fleet Computer Programming Centre, and consists of the errors in the development of software for the real time, multi computer complex which forms the core of the Naval Tactical Data Systems (NTDS). The NTDS software consisted of some 38 different modules. Each module was supposed to follow three 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 summarized in the below table.

Error Number n	Time between Errors Sk days	Cumulative Time $x_n = \sum S_k$ days
Production (Checkout) Phase		
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

TABLE 4.1 NTDS Data

The data set consists of 26 failures in 250 days. 26 software errors were found during production phase and five additional errors during test phase. One error was observed during the user phase and two more errors are noticed in a subsequent test phase indicating that a network of the module had taken place after the user error was found.

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

a^ = 55.018710
b^ = 0.998899
c^ = 278.610091

Hence, we may accept these three values as MLEs of a, b, c. The estimator of the reliability function from the equation (2.2) at any time x beyond 250 days is given by

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

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

$$= 0.081677$$

5. CONCLUSION

In this paper we have presented Pareto software reliability growth model with a mean value function. It provides a plausible description of the software failure phenomenon. This is called Pareto Type II Model. This is a simple method for model validation and is very convenient for practitioners of software reliability.

6. REFERENCES

[1] CROW, .H, and BASU, A.P. (1988). "Reliability growth estimation with missing data-II", Proceeding annual Reliability and Maintainability Symposium, 26-28.

[2] 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.

[3] Jelinski, Z and Moranda, P.B (1972) "Software reliability research", In:W.Freiberger,(Ed) Statistical Computer Performance Evaluation, New York:Academic Press 465-497.

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

[5] Musa,J.D. (1980) "The Measurement and Management of Software Reliability", Proceeding of the IEEE vol.68, No.9, 1131-1142.

[6] Pham. H (2005) "A Generalized Logistic Software Reliability Growth Model", Opsearch, Vol.42, No.4, 332-331.

[7] Ramamurthy, C.V., and Bastani, F.B.(1982). "Software Reliability Status and Perspectives", IEEE Transactions on Software Engineering, Vol.SE-8, 359-371.

[8] R.R.L.Kantam and R.Subbarao, 2009. "Pareto Distribution: A Software Reliability Growth Model". International Journal of Performability Engineering, Volume 5, Number 3, April 2009, Paper 9, PP: 275- 281.

[9] J.D.Musa and K.Okumoto,"A Logorithmic Poisson Execution time model for software reliability measure-ment", proceeding seventh international conference on software engineering, orlando, pp.230-238,1984.

[10] ZHANG,X., TENG,X. and PHAM,H. CONSIDERING FAULT REMOVAL EFFICIENCY IN SOFTWARE RELIABILITY ASSESSMENT, IEEE Transactions on Systems, Man and Cybernetics-part A, Vol.33, No.1, 2003; 114-120.

[11] MALAIYA, Y.K., KARUNANITHI, N., and VERMA, P. PREDICTABILITY OF SOFTWARE RELIABILITY MODELS, IEEE Transactions on Reliability, Vol, No.4. 1992; 539-546.

[12] WOOD, A. predicting software Reliability, IEEE Computer, 1996; 2253-2264.