RESEARCH ARTICLE OPEN ACCESS

Pareto Type II Software Reliability Growth Model– An Order Statistics Approach

Sita Kumari. Kotha[1], R.Satya Prasad[2]
Department of Information Technology and Engineering[1],
V.R.Siddhartha Engineering College,
Vijayawada
Department of Computer Science and Engineering[2],
Acharya Nagarjuna University,
Guntur
AP-India

ABSTRACT
In recent years the size and complexity of the software programs are getting increased and it is very much crucial to generate the software which is reliable, since the failures in the software may create a great loss. Several software reliability growth models exist to predict the reliability of software systems. It is very much necessary to find the reliability function for time domain data based on non-homogeneous Poisson process with a distribution model considering the order statistics approach. The main goal of this paper is to present the Pareto Type II model with order statistics as a software reliability growth model and derive the expressions for Reliability function that facilitates to compute the reliability of a product. The parameters are estimated using the maximum likelihood estimation procedure. The live data sets are analysed and the results are exhibited.
Keywords: - Software reliability, NHPP, Order Statistics, Pareto Type II Distribution, ML Estimation.

I. INTRODUCTION

Software reliability is one of the most important characteristics of software quality. Its measurement and management technologies employed during the software life cycle are essential for producing and maintaining quality/reliable software systems. Software Reliability is the probability of failure free operation of software in a specified environment during specified time [1]. Over the last several decades, many Software Reliability Growth Models (SRGMs) like Crow and Basu(1988), Goel Okumoto(1979,1984), Musa (1980), Pham (2005) and several models have been developed to greatly facilitate engineers and managers in tracking and measuring the growth of reliability as software is being improved[2]. The main goal in developing these models is to improve the software performance. During the testing phase of a software product, the failure data will be collected and these models predict the future system operability based on the failure data. As the software product is developed by human it is more likely to have faults and in this regard there was a continuous research going on in developing the software reliability growth models. Most of the models assume that the time between failures follows an exponential distribution so that the parameters vary with the errors remaining in the software system.

This paper presents the Pareto Type II model with order statistics to analyze the reliability of the software system. The main objective of this paper is to develop a model that provides a measurable software performance. The layout of this paper is as follows: Section II describes the Pareto model with order statistics and mean value function for the underlying NHPP, Section III explains the parameter estimation for Pareto Type II model with order statistic approach considering the time domain data, Section IV describes the technique used to analyse the failure data sets for live applications and section V refers to conclusion.

II. ORDER PARETO TYPE II MODEL

Software reliability is the most important and most measurable aspect of software quality and it is very customer oriented. With software Reliability it is possible to measure how well the program functions in meeting its operational requirements. Software reliability measures can promote quantitative specification of design goals and schedules the resources as required. These measures also help in the better management of project resources [4].

The user will also be benefited from software reliability measure, because the user is primarily concerned about the failure free operation of the system. If the operational needs with respect to quality are accurately specified, the user will either get a system at an excessively high price or with an excessively high operational cost .The most common approach in developing the software reliability models is the probabilistic approach [3].

The probabilistic model represents the failure occurrences and the fault removals as probabilistic events.

There are numerous software reliability models available for use according to probabilistic assumptions. They are classified into various groups, including error seeding models, failure rate models, curve fitting models, reliability growth models, Markov structure Models and non-homogenous passion process (NHPP) models[2]. Most of the models are NHPP based models.

A software system is subject to failures at random times due to the 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 assumed that the number of software failures during non-overlapping time intervals do not affect each other. It can be mentioned that for finite times t1<t2<t3<….<tn , the n random variables N(t1), {N(t2)-N(t1)}, ….. {N(tn)-N(tn-1)} are independent. It implies that the counting process {N(t), t>0} has independent increments.

Let m(t) denote 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 boundary conditions

M(t) = 0, t=0

= a, t -> ∞

Where a is the expected number of software errors need to be detected

Assume that N(t) is known to have a Poisson Probability mass function with parameters m(t) i.e.,

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

Where N(t) is called NHPP. The behaviour of software failure phenomena can be illustrated through N(t) process. Several time domain models exist in the literature [7] which specify that the mean value function m(t) will be varied for each NHPP process

The mean value function of Pareto Type II software reliability growth is given by

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

Here, we consider the performance given by the Pareto Type II software reliability growth model based on order statistics and whose mean value function is given by

$$m(t) = \left(a\left(1 - \frac{c^b}{(t+c)^b}\right)\right)^r \quad \text{----------2.2}$$

Where [m(t)/a] is the cumulative distribution function of Ordered Pareto distribution model

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

$$lim_{n\to\infty} P\{N(t) = N\} = \frac{e^{-a}.a^n}{n!} \quad \text{----------2.3}$$

This is considered as Poisson model with mean 'a'

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

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

$$= \frac{ac^b}{(t+c)^b} \quad \text{----------2.4}$$

Let Sk be the time between (k-1)th and Kth failure of the software product. It is assumed that X_k be the time up to the Kth failure. We need to find out the probability of the time between (k-1)th and Kth failures. The Software Reliability function is given by

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

III. PARAMETER ESTIMATION FOR ORDER PARETO TYPE II MODEL

In this section, the expressions are generated for estimating the parameters of the Ordered Pareto Type II model based on the time between the failures. The expressions for a, b, and c has to be derived.

Let S1, S2,…. be a sequence of times between successive software failures associated with an NHPP N(t). Let Xk be equal to

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

This represents the time at which failure k occurs. Suppose we are given with 'n' software failure times say x1, x2, … , xn, there are 'n' time instants at which the first, second, third …. nth failure of software is observed.

The mean value function of Ordered Pareto Type II is given by

$$m(t) = \left(a\left(1 - \frac{c^b}{(t+c)^b}\right)\right)^r \quad \text{----------3.1}$$

The constants a, b and c in the mean value function are called parameters of the proposed model. To assess the software reliability, it is necessary to compute the expressions for finding the values of a, b and c. For doing this, Maximum Likelihood estimation is used whose Likelihood function is given by

$$L = e^{-m(t)} * \prod_{i=1}^{n} m'(t_i) \quad \text{----------3.2}$$

The maximum likelihood estimators (MLEs) are the one that maximize the Likelihood function 'L' and the method is called maximum likelihood method of estimation.

[5]Differentiating m(t) with respect to 't'

$$m'(t) = r\left[a - \frac{ac^b}{(t+c)^b}\right]^{r-1} * \left[\frac{abc^b}{(t+c)^{b+1}}\right] \text{-------3.3}$$

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

$$logL = -a^r\left[1 - \frac{c^b}{(t+c)^b}\right]^r + \sum_{i=1}^{n} logr + \sum_{i=1}^{n}(r-1)\log\left[a - \frac{ac^b}{(t+c)^b}\right] + \sum_{i=1}^{n}[(log\, a + logb + blogc) - (b+1)\log(t_i + c)] \quad \text{------ 3.4}$$

[5]Differentiating with respect to 'a' we get

$$a^r = n * \left[\frac{(t+c)^b}{(t+c)^b - c^b}\right]^r$$

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

$$\frac{\partial Log\, L}{\partial a} = 0, \quad \frac{\partial Log\, L}{\partial b} = 0 \quad \frac{\partial^2 Log\, L}{\partial b^2} = 0$$
$$\frac{\partial Log\, L}{\partial c} = 0, \quad \frac{\partial^2 Log\, L}{\partial c^2} = 0$$

$$g(b) = \frac{nr}{(t+1)^b - 1}\log\left(\frac{1}{t+1}\right) + \sum_{i=1}^{n}(r-1)\log(t_i+1)\frac{1}{[(t_i+1)^b - 1]}$$
$$+\frac{n}{b} - \sum_{i=1}^{n} log(t_i + 1) \quad \text{----------3.5}$$

.

Second order partial derivative with respect to the parameter 'b'

$$g'(b) = nrlog\left(\frac{1}{t+1}\right)(-1) * \frac{(t+1)^b \log(t+1)}{[(t+1)^b - 1]^2} - \sum_{i=1}^{n}(r-1)\log(t_i+1)(t_i+1)^b log(t_i+1)[(t_i+1)^b - 1]^{-2} - \frac{n}{b^2} \quad \text{----3.6}$$

$$g(c) = \frac{nr}{t+c} - \sum_{i=1}^{n}\frac{(r-1)}{(t_i+c)} + \frac{n}{c} - \sum_{i=1}^{n}\left(\frac{2}{t_i+c}\right) \quad \text{--------3.7}$$

Second order partial derivative with respect to the parameter 'c'

$$g'(c) = \frac{-nr}{(t+c)^2} + \sum_{i=1}^{n}\frac{(r-1)}{(t_i+c)^2} - \frac{n}{c^2} + \sum_{i=1}^{n}\frac{2}{(t_i+c)^2} \quad \text{--3.8}$$

The values of the parameters a,b and c in the above equations are computed by using the well known iterative method like Newton Raphson method[5]. The values of b and c can be estimated by solving the equations 3.5, 3.6, 3.7 and 3.8 iteratively.

IV. ORDER STATISTICS

Order Statistics can be used for several applications. They can be used in several applications like data compression, survival analysis, Study of Reliability and many others[9]. Let X denote a continuous random variable with probability density function f(x) and cumulative distribution function F(x), and let (X_1 , X_2 , …, X_n) denote a random sample of size n drawn on X. The original sample observations may be unordered with respect to magnitude. A transformation is required to produce a corresponding ordered sample. Let ($X_{(1)}$, $X_{(2)}$, …, $X_{(n)}$) denote the ordered random sample such that $X_{(1)} < X_{(2)} < \ldots < X_{(n)}$; then ($X_{(1)}$, $X_{(2)}$, …, $X_{(n)}$) are collectively known as the order statistics derived from the parent X. The various distributional characteristics can be known from Balakrishnan and Cohen [10]. The inter-failure time data represent the time lapse between every two consecutive failures. On the other hand if a reasonable waiting time for failures is not a serious problem, we can group the inter-failure time data into non overlapping successive sub groups of size 4 or 5 and add the failure times with in each sub group. For instance if a data of 100 inter-failure times are available we can group them into 20 disjoint subgroups of size 5. The sum total in each subgroup would devote the time lapse between every 5th order statistics in a sample of size 5. In general for inter-failure data of size 'n',

if r (any natural no) less than 'n' and preferably a factor n, we can conveniently divide the data into 'k' disjoint subgroups (k=n/r) and the cumulative total in each subgroup indicate the time between every rth failure. The probability distribution of such a time lapse would be that of the r^{th} ordered statistics in a subgroup of size r, which would be equal to r^{th} power of the distribution function of the original variable (m(t)). The whole process involves the mathematical model of the mean value function and knowledge about its parameters. If the parameters are known they can be taken as they are for the further analysis, if the parameters are not known they have to be estimated using a sample data by any admissible, efficient method of estimation. This is essential because the control limits depend on mean value function, which in turn depends on the parameters. If software failures are quite frequent keeping track of inter-failure is tedious. If failures are more frequent order statistics are preferable [11].

V. DATA ANALYSIS

CSR3 Data set [12] has been taken into consideration to analyse the software reliability.

Failure No	Time Between Failures(hrs)	Failure No	Time Between Failures(hrs)
1	33	53	1
2	9	54	400
3	4	55	294
4	66	56	227
5	0.5	57	118
6	18	58	13
7	149	59	47
8	14	60	89
9	15	61	242
10	50	62	99
11	81	63	607
12	34	64	83
13	85	65	2
14	54	66	26
15	3	67	586
16	15	68	708
17	6	69	6
18	8	70	4
19	130	71	55
20	19	72	409
21	19	73	36
22	112	74	15
23	15	75	573
24	16	76	583
25	154	77	60
26	50	78	19
27	10	79	20
28	2	80	79
29	22	81	24
30	53	82	540
31	19	83	52
32	58	84	1596
33	20	85	314
34	3	86	1
35	92	87	763
36	5	88	10
37	66	89	20
38	289	90	144
39	3	91	28
40	9	92	56
41	12	93	476
42	18	94	65
43	9	95	98
44	75	96	884
45	15	97	212
46	291	98	287
47	212	99	53
48	4	100	3
49	5	101	831
50	308	102	43
51	269	103	55
52	276	104	109

CSR3 Data Set (4^{th} Order) (Michael R.Lyu., 1996a)

Failure No	4^{th} order Time Between Failures S_k days	4^{th} order cumulative Time Between Failures $X_n = \sum s_k$ days
1	112	112
2	181.5	293.5
3	180	473.5
4	157	630.5
5	163	793.5
6	162	955.5
7	216	1171.5
8	152	1323.5
9	120	1443.5
10	367	1810.5

11	114	1924.5
12	522	2446.5
13	858	3304.5
14	922	4226.5
15	267	4493.5
16	1031	5524.5
17	1322	6846.5
18	474	7320.5
19	1207	8527.5
20	178	8705.5
21	2212	10917.5
22	1088	12005.5
23	248	12253.5
24	1523	13776.5
25	555	14331.5
26	1038	15369.5

The CSR3 data set consists of 26 failures for 4th order statistics in 15369 days. By solving the equations in section III by Newton Rapson method , we can obtain the MLE's of a, b and c for CSR3 data set.

a^ = 26.026768

b^ = 1.000276

c^ = 3.961974

The estimator of the reliability function at any time x beyond 15369.5 days is given by

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

$$R\,\frac{S_{27}}{X_{(26)}}(15369.5/955.5) = e^{-[m(955.5+15369.5)-m(15369.5)]}$$

$$= e^{-[m(16325)-m(15369.5)]}$$

=0.999608

CSR3 Data set (5th Order Statistics) .(Michael R.Lyu., 1996a)

Failure No	5th order Time Between Failures Sk days	5th order cumulative Time Between Failures X n = $\sum s_k$ days
1	112.5	112.5
2	246	358.5
3	257	615.5
4	178	793.5
5	316	1109.5
6	137	1246.5
7	192	1438.5
8	372	1810.5
9	129	1939.5
10	820	2759.5
11	1240	3999.5
12	494	4493.5
13	1033	5526.5
14	1330	6856.5
15	1088	7944.5
16	761	8705.5
17	2526	11231.5
18	938	12169.5
19	723	12892.5
20	1439	14331.5

The CSR3 data set consists of 20 failures for 5th order statistics in 14331 days. By solving the equations in section III by Newton Rapson method , we can obtain the MLE's of a, b and c for CSR3 data set.

a^ = 20.00698

b^ = 0.999933

c^ = 1.047529

The estimator of the reliability function at any time x beyond 14331.5 days is given by

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

$$R\ \frac{S_{21}}{X_{20}}(14331.5/1109.5) = e^{-[m(1109.5+14331.5)-m(14331.5)]}$$

$$= e^{-[m(15441)-m(14331.5)]}$$

$$= 0.999895$$

VI. CONCLUSIONS

In this paper, the Pareto type II distribution model with order statistics has been proposed. Today 70 to 80 % of people use software and it is very much essential to produce reliable software. The proposed model has been tested with live data set for 4^{th} order and 5^{th} order and proved that it has high reliability. It is also observed that the reliability is high for 5^{th} order statistic than 4^{th} order statistics. Finally it can be concluded that the model has produced very good results and is very much comfortable to compute the reliability.

REFERENCES

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

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

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

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

[5] Sitakumari.k, Satya Prasad.R , Assessing Software Quality with Time Domain Pareto Type II using SPC SPC, IJCA, 2014

[6] Dr.R.Satya Prasad, NGeetha Rani, Prof R.R.L.Kantham, Pareto Type II Based Software Reliability Growth Model, International Journal of Software Engineering, Vol (2), 2011

[7] 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.

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

[9] Arak M. Mathai ;Order Statistics from a Logistic Dstribution and Applications to Survival and Reliability Analysis;IEEE Transactions on Reliability, vol.52, No.2; 2003

[10] Balakrishnan.N., Clifford Cohen; Order Statistics and Inference; Academic Press inc.;1991

[11] K.Ramchand H Rao, R.Satya Prasad, R.R.L.Kantham; Assessing Software Reliability Using SPC – An Order Statistics Approach; IJCSEA Vol.1, No.4, August 2011

[12] Michael R.Lyu 1996a, Handbook of Software Reliability Engineering