IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 9, Sept 2018

www.ijasrm.com

ISSN 2455-6378

Considering Residual Faults of Inverse Rayleigh Software Reliability Growth Model

BNV Uma Shankar[1], Dr G.Sridevi[2]

[1]Asst.Professor,Department of Science & Humanities,CVR College of Engineering, Hyderabad, India

[2]Professor,Department of CSE, Malla Reddy Institute of Technology, Secunderabad, India

Abstract

A Non Homogenous Poisson Process (NHPP) with its mean value function generated by the cumulative distribution function of inverse Rayleigh distribution is considered. It is modeled to assess the failure phenomenon of developed software. When the failure data is in the form of number of failures in a given interval of time the model parameters are estimated by the maximum likelihood method. 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.

Keywords: *NHPP, SRGM, Maximum Likelihood Estimation, Inverse Rayleigh Distribution.*

1. Introduction

It is well-known that computers are used in diverse areas for various applications. The growing importance of software dictates that a reliable software is by all means essential. A software itself does not fail unless the faults within the software result in its failure. Generally, software faults are more difficult to handle. All design faults are present from the time the software is installed in the computer. A software fault inherent in a program is not dangerous unless and until it results in a failure of software. Accordingly, the concept of software reliability is rather dependent on the failure of a software and its frequency rather than the unknown number of faults latent in the software. Therefore, the term software reliability may be defined as the probability of failure free functioning of a software rather than the faults contained in it. However we cannot risk out the fact that software reliability depends on the number of faults also. In this regard, theory of probability and hence statistical analysis have become essential in the development of a model that can be used to evaluate the reliability of real world software systems. Software reliability models are statistical models which can be used to make predictions about a software system's failure rate , given the failure history of the system. The models make assumptions about a fault discovery and removal process. These assumptions determine the form of the model and the meaning of the model's parameters. Some recent works in this regard are by (Sridevi et al., 2015), (Yamada et al., 1986), (Vara Prasad Rao et al., 2013) , (Pham 2000) , (Huang et al., 2000), (Kapur et al., 2002), (Pham and Zhang 2003), (Yamada et al 2003), Sridevi et al., 2013), (Kapur et al., 2005), (Pham 2005), (Quadri et al., 2010), (Satya Prasad R., 2007). With this backdrop, we study the modeling of software reliability as a Non Homogenous Poisson Process (NHPP) with mean value function based on inverse Rayleigh distribution. Similar attempts based on Pareto distribution is made by (Kantam et al., 2009). The genesis and the development of the model with the necessary input about a Non Homogenous Poisson Process are presented in Section 2, the proposed model description is presented in Section 3, and the maximum likelihood (ML) estimations are given in Section 4. The method of performance analysis is given in Section 5 and Summary and Conclusions are given in Section 6.

2. SRGM as A Non Homogenous Process

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. 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.,

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

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 $m(t)$.

3. Proposed Model Description

Since a number of distributions is available in statistical science, one can think of a number of NHPP models each based on a CDF. The first and foremost of such models is due to (Goel and Okumoto 1979) which is based on the well-known exponential distribution. Later many such models have been suggested and studied by various researchers that can be found in (Wood 1996), (Pham 2000) and references therein. The probability density function of inverse Rayleigh distribution (IRD) with scale parameter b is

$$f(x) = \frac{2b}{x^3} e^{\left(\frac{-b}{x^2}\right)}, \quad x > 0, b > 0$$
(2)

Its cumulative distribution function is (cdf) is given by

$$F(x) = e^{\left(\frac{-b}{x^2}\right)}, \quad x > 0, b > 0$$
(3)

We consider an NHPP with the mean value function given in terms of the CDF of inverse Rayleigh distribution (IRD) given as

$$m(t) = ae^{\frac{-b}{t^2}}, a > 0, t > 0 \quad (4)$$

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

$$RS_k / X_{k-1}(s/x) = e^{-[m(x+s)-m(s)]} \quad (5)$$

This expression is called Software Reliability.

4. Illustrating Ml Estimation

In this section we develop expressions to estimate the parameters of the inverse Rayleigh distribution model based on interval 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, interval domain data and time domain data. In this paper parameters are estimated from the interval domain data.

The mean value function of inverse Rayleigh model is given by

$$m(t) = ae^{\frac{-b}{t^2}}, a > 0, t > 0$$
(6)

The Log likelihood function to get the estimates of parameters of the NHPP shall be of the form

IJASRM
International Journal of Advanced Scientific Research and Management, Volume 3 Issue 9, Sept 2018
www.ijasrm.com
ISSN 2455-6378

$$LogL = \sum_{i=1}^{k} (n_i - n_{i-1}) \log\left[m(t_i) - m(t_{i-1})\right] - m(t_k) \quad (7)$$

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

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

$$a = y_n e^{\frac{b}{t_n^2}} \quad (8)$$

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

$$g(b) = \sum_{i=1}^{n} \frac{t_{i-1}^2 . e^{\frac{-b}{t_{i-1}^2}} - t_i^2 e^{\frac{-b}{t_i^2}}}{e^{\frac{-b}{t_i^2}} - e^{\frac{-b}{t_i^2 - 1}}} (y_i - y_{i-1}) - y_n t_n^2 \quad (9)$$

Solving the equations (8) and (9) simultaneously for a given sample data we get the ML estimates of a and b. The ML estimates for four different data sets published in (Wood 1996) are given in Table1.

Table1: Different real time data sets

	DS1		DS2		DS3		DS4	
Test Week	CPU Hours	Defects found	CPU Hours	Defects found	CPU Hours	Defects found	CPU Hours	Defects found
1	519	16	384	13	162	6	254	1
2	968	24	1186	18	499	9	788	3
3	1430	27	1471	26	715	13	1054	8
4	1893	33	2236	34	1137	20	1393	9
5	2490	41	2772	40	1799	28	2216	11
6	3058	49	2967	48	2438	40	2880	16
7	3625	54	3812	61	2818	48	3593	19
8	4422	58	4880	75	3574	54	4281	25
9	5218	69	6104	84	4234	57	5180	27
10	5823	75	6634	89	4680	59	6003	29
11	6539	81	7229	95	4955	60	7621	32
12	7083	86	8072	100	5053	61	8783	32
13	7487	90	8484	104	-----	----	9604	36
14	7846	93	8847	110	-----	----	10064	38
15	8205	96	9253	112	-----	----	10560	39
16	8564	98	9712	114	-----	----	11008	39
17	8923	99	10083	117	-----	----	11237	41
18	9282	100	10174	118	-----	----	11243	42
19	9641	100	10272	120	-----	----	11305	42
20	10000	100	----	-----	-----	----	----	----

Solving Equations by Newton Raphson method for DS1, DS2, DS3 and DS4, the iterative solutions for MLEs of' a' and 'b' are presented in Table 2.

Table 2: Estimated parameters of the proposed model			
Dataset	Number of samples	Estimated Parameters	
		a	**b**
DS 1	20	107.0339	27.1805
DS 2	19	156.6663	96.2521
DS 3	12	68.21619	16.1002
DS 4	19	47.7392	46.2382

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 performance evaluation of software reliability growth model is generally measured with sum of square errors (SSE) and correlation index of regression curve equation (R-square). Among them, the model performance is better when SSE is smaller and R-square is close to 1.

SSE is used to describe the distance between actual and estimated number of faults detected totally, which is defined as

$$SSE=\sum_{i=1}^{n}\left(y_i - m(t_i)\right)^2 \tag{10}$$

Where n denotes the number of failure samples in failure data set, y_i denotes the number of faults observed to the moment t_i, and $m(t_i)$ denotes the estimated number of faults detected to the time t_i according to the proposed model. The model can provide a better goodness-of-fit when the value of SSE is smaller.

The equation of calculating the value R-square is written as:

$$R-square=\frac{\sum_{i=1}^{n}\left(\bar{y}-m(t_i)\right)^2}{\sum_{i=1}^{n}\left(\bar{y}-y_i\right)^2} \tag{11}$$

Where $\bar{y}$ denotes the mean value of faults detected. The model can provide a better goodness-of-fit when the value of R-square is close to 1. The reliabilities and performance of the different data sets are presented in Table 3.

From the Table -3 it can be seen that the value of SSE is smaller and the value of R-square is more close to 1. The results indicate that our NHPP inverse Rayleigh distribution model based on fault detection rate fits the data in the given datasets, best and predicts the number of residual faults in software most accurately.

Table 3: The results on different datasets			
Data Set	Reliability (t_n+50)	SSE	R-Square
DS 1	0.747890	40147.9804	1.602093
DS 2	0.352202	104678.4843	3.866083
DS 3	0.744917	11246.8730	1.728165
DS 4	0.290814	3803.7800	1.832837

6. Conclusion

Software reliability growth model can estimate the optimal software release time and the cost of testing efforts. And SRGM can help project managers to determine the testing resources and manpower needed to achieve desired reliability requirements. So more accurate model is needed to decrease the testing cost and increase the profit of releasing software. In this paper the fault detection rate is calculated with the number of faults remaining in the software. Considering the two factors jointly the fault detection rate is more realistic and accurate. Moreover, we have discussed the performances of 4 datasets by using our new inverse Rayleigh SRGM. The experiment result shows that the data set DS4 can provide a better goodness-of-fit compared with other datasets are given in Table 3.The reliability of the model over DS1 is high among the data sets which were considered.

7. References

[1] Anu Gupta., Kapur.P.K and Omphal Singh., On discrete software reliability growth model and categorization of faults. Operational Research Society of India, 42(4): pp: 340–354, (2005).

[2] Bardhan.A.K., Kapur.P.K. and Shatnawi.O. Why software reliability growth modelling should define errors of different severity, Journal of Indian Statistical Association, 40(2), pp:119–142, (2002).

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

[4] Hoang Pham., A generalized logistic software reliability growth model. Operational Research Society of India (OPSEARCH), 42(4):322–331, (2005).

[5] Kuo.S.Y. Huang.C.Y. and Lyu.M.R. Effort-index based software reliability growth models and performance assessment. In Proceedings of 24th IEEE Annual International Computer Software and Applications Conference, pages 454–459, (2000).

[6] Kantam.R.R.L., and Subba Rao.R. Pareto distribution – a software reliability growth model, International Journal of Performability Engineering, 5(3):275–281, (2009).

[7] Ohtera.H. Yamada.S. and Narihisa.R. Software reliability growth models with testing effort. IEEE Transactions on Reliability, R-35:19–23, (1986).

[8] Pham, H., Software Reliabilit,. Springer-Verlag, (2000).

[9] Pham.H., and Zhang.X. Non homogeneous poisson process software reliability and cost models with testing coverage. European Journal of Operational Research, (145):443–454, (2003).

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

[11] Sridevi G., Satya Prasad R and Rama devi B., Considering Residual faults of Burr type XII Software Reliability Growth Model, International Journal of Electronics and Computer Science Engineering, Vol. 4, No. 2, pp. 123-129, (2015).

[12] Sridevi G and Satya Prasad R., Pareto Type IV Software Reliability Growth Model, Elixir Comp. Sci. & Engg. 64,pp: 19124-19129, (2013).

[13] Satya Prasad R., Half logistic distribution for software reliability growth model, Ph.D thesis, (2007).

[14] Tamura.Y., Yamada.S. and Kimura.M. A software reliability growth model for a distributed development environment, Electronic and Communications in Japan, 83(3):1446–1453, (2003).

[15] Vara Prasad Rao. B, Gangadhara Rao. K and Srinivasa Rao. B, Inverse Rayleigh Software Reliability Growth model", International Journal of Computer applications, Vol. 75, No. 6, (2013).

[16] Wood.A.P., Predicting software reliability. IEEE Computer, 11:69–77, (1996).