Software Reliability Growth Model with Efficient Debugging Involving Time Dependent Fault Content Function

Suneet Saxena, Ph.D
Assistant Professor
Deptt. of Mathematics
SRMSCET, Bareilly

Digvijay Choudhary, Ph.D
Assistant Professor
Deptt. of Mathematics
JPIET, Meerut

Ajay Gupta, Ph.D
Assistant Director
BIT,
Muzaffarnagar

ABSTARCT

The paper presents NHPP software reliability growth model exhibiting efficient debugging phenomenon. In efficient debugging, cumulative faults corrected more than the faults responsible for software failures. Proposed model incorporates time dependent exponential fault content function and I/D pattern of learning process. Parameters are estimated by LSE and goodness of fit is performed. Determination of MSE and R^2 indicates model is fit better to given dataset. Finally software reliability has been analyzed.

Keywords

Software Reliability, Efficient Debugging, NHPP Model, Exponential Function.

1. INTRODUCTION

Reliability of software depends on fault detection and correction process. Software reliability growth models (SRGM) are mathematical models which measures reliability of software. SRGM provides mathematical relationship between cumulative faults detected and time during testing phase. These models are based on either homogenous poisson process (HPP) or non homogenous poisson process (NHPP). HPP involves constant failure rate whereas in NHPP failure rate is variable. NHPP SRGMs [3-10] are used extensively for reliability estimation of software.

Debugging is the process of fault detection and correction. There are two type of debugging process, perfect and imperfect. In perfect debugging faults corrected with certainty which are responsible for software failures and no new faults are introduced. The phenomenon of imperfect debugging occurs due to introduction of new faults during the correction process and also faults corrected are less than the faults responsible for failures. Using imperfect debugging various SRGMs [3-7] have been developed. Subburaj, Gopal and Kapur[4] proposed efficient debugging phenomenon in which faults corrected are more than the faults responsible for software failures. This phenomenon occurs when learning skills of testing team improves during repair process of existing faults. Based on learning skills, the team repair those faults in advance which yet to be detected. We have developed model which exhibits efficient debugging by considering time dependent fault content function and I/D pattern of learning process. The paper is organized as follows: Section 2 describes notation and assumption. Section 3 presents model development. Section 4 evaluates model by parameter estimation and goodness of fit. Section 5 discussed reliability analysis and conclusion.

2. NOTATIONS AND ASSUMPTIONS

i. Model follows NHPP Process.
ii. $\mathrm{m(t)}$ is mean value function of expected number of faults detected in time (0,t).
iii. $\mathrm{f(t)}$ is fault content function depends on time. We have considered $\mathrm{f(t) = ae^{\alpha t}}$, where a is initial faults and α fault introduction rate.
iv. b represents fault detection rate.
v. c is debugging index. For perfect, imperfect and efficient debugging $c =, < and > 1$ respectively.

3. MODEL DEVELOPMENT

The rate of faults detected at any time is given by

$$\frac{dm}{dt} = b(f(t) - m)\ . \qquad (1)$$

Introducing exponential fault content function and debugging index c, we get

$$\frac{dm}{dt} = b(ae^{\alpha t} - cm) \qquad (2)$$

When $c = 1$ debugging is perfect, whereas $c < 1$ and $c > 1$ represent imperfect and efficient debugging respectively.

Using condition m(0)=a and solving equation, we get solution

$$m(t) = \frac{ba}{\alpha + cb}\left(e^{\alpha t} - e^{-bct}\right).\ . \qquad (3)$$

In order to introduce learning phenomenon i.e. increasing/decreasing pattern (I/D) of failure intensity function, above solution modified to

$$m(t) = \frac{ba}{\alpha + cb}\left(e^{\alpha t} - e^{-bct}\right)^{d}. \qquad (4)$$

I/D pattern given by $d > 1$.

4. MODEL EVALUATION

4.1 Parameter Estimation

We have used Musa's [2] SS3 and SS4 failure dataset. Parameters of model are estimated by Least Square Error (LSE) method. Table 1 represents parameters evaluated by Least Square Error method.

4.2 Goodness of Fit (GoF)

Some of the measures used to determine GoF are given below:

i. Sum of Square Error (SSE)

ii. Mean Square Error (MSE)
iii. Root Mean Square Prediction Error (RMSPE)
iv. Coefficient of Determination (R^2)

4.2.1 Coefficient of Determination (R^2)

Coefficient of Determination is also known as multiple correlation coefficient. It measures the correlation between the dependent and independent variables. Value of R^2 vary from 0 to 1. The conditions are given below :

i. $R^2 = 1$, perfect fitting.
ii. $R^2 = 0$, no fitting.
iii. R^2 close to 1, good fitting.

We have evaluated MSE and R^2. Table 2 represents the values of both measures. Model fits better to given dataset if value of MSE is low and R^2 close to 1.

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(y_j - E_J)^2$$

$$R^2 = \frac{\sum_{j=1}^{n}(E_J - \bar{y})^2}{\sum_{j=1}^{n}(y_j - \bar{y})^2}$$

$$\bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j \quad \text{where}$$

n : number of data points.
y_j : observed cumulative faults at time j .
E_J: estimated cumulative faults at time j.

Table 1: Estimated Parameters.

	α	a	b	c	d
SS3	0.0046	75	0.007	1.02	1.001
SS4	0.0069	50	0.026	1.065	1.1

Table 2: Goodness of Fit.

	MSE	R^2
SS3	26.90	0.98
SS4	22.87	0.86

Figure 1 and 2 represent cumulative faults detected with respect to time for dataset SS3 and SS4 respectively.

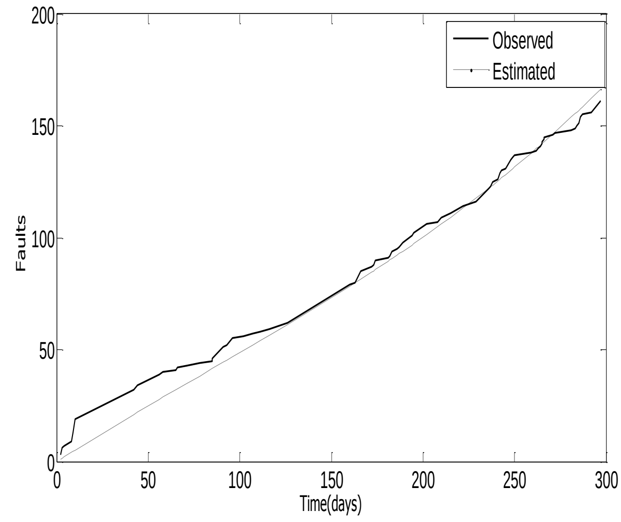

Fig 1: SS3 data set

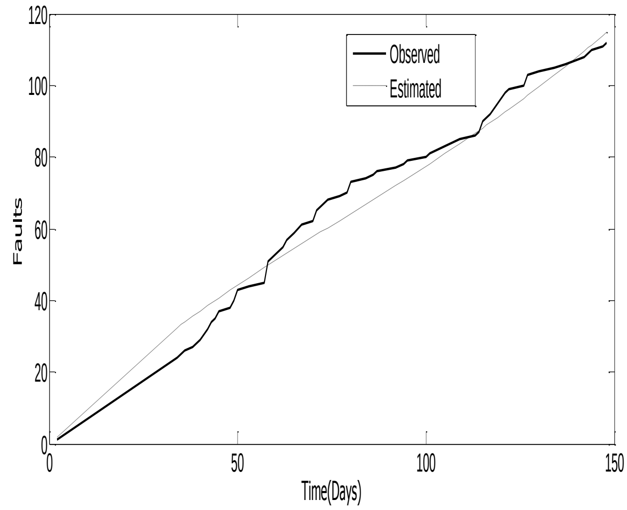

Fig 2: SS4 data set

5. ANALYSIS

5.1 Reliability Analysis

Software reliability is defined as the probability that software perform its operation without failure in time interval (t, t + Δt) under specified conditions given by

$$R\left(\frac{\Delta t}{t}\right) = exp[-\{m(t+\Delta t) - m(t)\}]$$

Using Δt = 1 day , reliability has been analyzed with respect to testing time (Figure 3). $m(t+\Delta t)$ and $m(t)$ evaluated using parameters estimated by SS3 and SS4 data set.

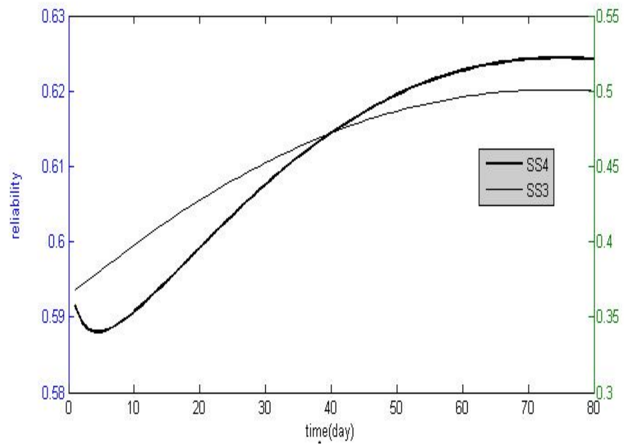

Fig 3: Software reliability with respect to time.

5.2 Conclusion

This paper presents software reliability growth model incorporating learning phenomenon, efficient debugging and exponential fault content function. Corresponding to the dataset SS3 and SS4, values of MSE are low for proposed model. Also R^2 close to 1 (table 2). These values of GoF measures indicate proposed model fits better.

6. REFERENCES

[1] Lyu, M. R. 1996. Handbook of Software Reliability Engineering, McGraw- Hill and IEEE Computer Society, pp. 27-164, New York.

[2] Musa. 2005 DACS software reliability dataset, Data & Analysis Center for Software, J.D. 980. www.dacs.dtic.mil/databases/sled/swrel.shtml.

[3] Pham, H., Nordmann, L., and Zhang, X. 1999. A general imperfect-software debugging model with S-shaped fault-detection rate, IEEE Transactions on Reliability, Vol.48, No.2, pp. 169-175.

[4] Subburaj, R., Gopal, G., and Kapur, P.K. 2007. A software reliability growth model for vital quality metrics, South African Journal of Industrial Engineering, vol 18(2), pp 93-108.

[5] Subburaj, R., and Gopal, G. 2006. Generalized Exponential Poisson Model for software reliability growth, International Journal of Performability Engineering, Vol.2, No.3, pp. 291-301.

[6] Xie, M. 2003. A study of the effect of imperfect debugging on software development cost, IEEE Transactions on Software Engineering, Vol.29, No.5, pp. 471-473.

[7] Zeephongsekul, P., Xia, G., and Kumar, S. 1994. Software-reliability growth model: Primary-failures generate secondary-faults under imperfect debugging, IEEE Transactions on Reliability, Vol. 43(3), pp. 408-413.

[8] Yamada, S., and Osaki, S. 1985. Software reliability growth modeling: Models and applications, IEEE Transactions on Software Engineering, Vol. SE-11(12), pp.1431-1437.

[9] Zhang, X., Teng, X., and Pham, H. 2003. Considering fault removal efficiency in software reliability assessment, IEEE Transactions on Systems, Man and Cybernetics – part A: Systems and Humans, vol. 33 (1), pp.114-120.

[10] Zhang, X. M, Teng, X. L., and Pham, H. 2003. Considering fault removal efficiency in software reliability assessment, IEEE Trans. Systems, Man, and Cybernetics−Part A: System and Humans, vol. 33, no. 1, pp. 114-120.