Software Reliability with Logistic Growth Model

R. Jeromia Muthuraj and A. Loganathan
Department of Statistics
Manonmaniam Sundaranar University
Tirunelveli, Tamilnadu – 627 012.

ABSTRACT: Many software reliability growth models have been proposed for measuring the growth of reliability of the software. This paper describes Logistic Growth Model (LGM) to study software reliability. Software Reliability Growth Model (SRGM) is a mathematical model of how the reliability of the software improves as faults are detected and repaired. The performance of SRGM is judged by its ability to fit the software failure data. The model which is used for the analysis Logistic Software Reliability Growth Model. An approach is used to estimate the model parameters by Maximum Likelihood Estimation (MLE) and least square Estimation (LSE). To assess the performance of the considered SRGM, It's carried out the parameter estimation on the real software failure data sets.

Keywords: Logistic Growth Model, Software Reliability, Maximum likelihood Estimation, Least square Estimation.

1. INTRODUCTION

Software reliability is the probability that a software will function without failure under a given environmental condition during a specified period of time. Software Reliability Modeling plays a vital role in developing software systems and enhancing computer software. Software reliability theory deals with probabilistic methods applied to the analysis of random occurrence of failures in a given software system. A software is said to contain a fault if, for input data, the output result is incorrect. Fault is always an inevitable part in software codes. Therefore, the process of software debugging is a fundamental task of the life cycle of a software system. During this period, the software program is tested many times to detect the faults contained. When a failure is observed, the code is inspected to find the fault which caused the software failure. The fault is usually removed by correcting the software codes. As a result, the software reliability will be increase during the testing phase as more and more faults are removed. The reliability improvement phenomenon is called reliability growths. SRGM are useful for estimating how software reliability improves as the faults are detected and repaired. It can be used to predict the ultimate level of reliability and also helps to stop testing if the given reliability level is archived. SRGMs help in decision making in many software development activities such as number of initial faults, failure intensity, reliability within a specified interval of time period, number of remaining faults, cost analysis and release time etc.

2. LITERATURE SURVEY

There are many hardware reliability approaches but Software Reliability Modeling (SRM) work started in the early '70s, with the inventive works of Jelinski and Moranda (1972), Shooman and Coutinho. After that many works were done related to software reliability. Many software reliability models were constructed in parametric and non-parametric approaches. Some parametric models are Jelinski and Moranda De-Eutrophication Model (1972), Schick and Wolver ton Model, Goel and Okumoto Imperfect Debugging Model, Littlewood - Verrall Bayesian Model (1973), Goel-Okumoto Non-homogeneous Poisson Process Model, Shooman Exponential Model, and etc. Some Non Parametric models are A Non-Parametric Order Statistics Software Reliability Model (1998), State Transition Model for Predicting Software Reliability (2007), and etc. The experts say that there are more than 225 software reliability models. But there is not even a single model that can be used in all situations. A model may work well for a set of certain software, but it may be completely off track for other kinds of problems. The NHPP type's software reliability models provides an analytical framework for describing the software failure occurrence during testing. They are proved to be quite successful in practical software reliability engineering (Musa et al., 1987).SRGMs are a statistical interpolation of defect detection data by mathematical functions (Wood, 1996). They have been grouped into two classes of models- Concave and S shaped. The only way to verify and validate the software is by testing. This involves running the software and checking for unexpected behavior of the software output (Kapur et al., 2009). SRGMs are used to estimate the reliability of a software product. In literature, we have several SRGMs developed to monitor the reliability growth during the testing phase of the software development.

3. MODEL DESCRIPTION

A Poisson process model for describing about the number of software failures in a given time (0, t) is given by the probability equation.

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

Where, $m(t)$ is a failure intensity function of $\lambda(t)$, finite valued non negative and non-decreasing function of 't' called the mean value function. Such a probability model for $N(t)$ is said to be an NHPP model.

Let 'a' denote the expected number of faults that would be detected given infinite testing time in case of finite failure NHPP models. Then, the mean value function of the finite failure NHPP models can be written as:

$$m(t)=aF(t)$$

The failure intensity function

$$\lambda(t)=aF'(t)$$

One simple class of finite failure NHPP model is the LGM, assuming that the failure intensity is proportional to the number of faults remaining in the software describing an exponential failure curve. It has two parameters. Where, 'a' is the expected total number of faults in the code and 'b' is the shape factor defined as, the rate at which the failure rate decreases. The cumulative distribution function of the model is:

$$F(t)=\frac{1-e^{-bt}}{1+e^{-at}}.$$

The expected number of faults at time 't' is denoted by

$$m(t)=\frac{a(1-e^{-bt})}{(1+e^{-at})},\ a>0,b>0,t\geq 0.$$

The corresponding failure intensity function is given

$$\lambda(t) = \frac{2abe^{-bt}}{1+e^{-bt}}.$$

The Reliability function is

$$R(t) = e^{-(m(t_i)-m(t_{i-1})}.$$

4. PARAMETER ESTIMATION

4.1 maximum likelihood estimation

The idea behind maximum likelihood parameter estimation is to determine the parameters that maximize the probability of the sample data. The method of maximum likelihood is considered to be more robust and yields estimators with good statistical properties. In other words, MLE methods are versatile and apply to many models and to different types of data. Although the methodology for MLE is simple, the implementation is mathematically intense. Using today's computer power, however, mathematical complexity is not a big obstacle. If we conduct an experiment and obtain N independent observations, $t_1, t_2, ..., t_N$. The likelihood function may be given by

$$L(t_1, t_2, ..., t_N \mid \theta_1, \theta_2, ..., \theta_k) = \prod_{i=1}^{N} f(t_i; \theta_1, \theta_2, ..., \theta_k)$$

Likelihood function by using $\lambda(t)$ is

$$L = e^{-m(t)} \prod_{i=1}^{N} \lambda(t_i)$$

Log Likelihood function is

$$\log L = \sum_{i=1}^{n} \log[\lambda(t_i)] - m(t_n)$$

Procedure for find Parameter 'a' using MLE

The Likelihood function is given by

$$L = e^{-\frac{a(1-e^{bt})}{(1+e^{-bt})}} \prod_{i=1}^{N} \frac{2abe^{-bt}}{(1+e^{-bt})}$$

Log likelihood function of the above equation is

$$\log L = n\log 2 + n\log a + n\log b - b\sum_{i=1}^{n} t_i - 2\sum_{i=1}^{n} \log(1+e^{-bt_i}) - a\left(\frac{1-e^{bt_n}}{1+e^{bt_n}}\right)$$

Taking the Partial derivative with respect to ‘a’ and equating to ‘0’.

$$a = n\left[\frac{1-e^{-bt}}{1+e^{-at}}\right]$$

Taking the Partial derivative with respect to 'b' and equating to'0'.

$$g(b) = \sum_{i=1}^{n} t_i - \frac{n}{b} - 2\sum_{i=1}^{n} \frac{t_i e^{-bt_i}}{(1+e^{-bt_i})} - \frac{2t_i e^{-bt_n}}{(1+e^{-bt_n})}\left[1 - \frac{n}{(1-e^{-bt_i})}\right]$$

4.2 Least square Estimation

Least square Estimation is a popular technique and extensively used in many fields for function fit and parameter estimation .The least squares method finds values of the parameters such that the sum of the squares of the difference between the fitting function and the experimental data is minimized. Least squares linear regression is a method for predicting the value of a dependent variable Y, based on the value of an independent variable X. Linear regression finds the straight line, called the least squares regression line that best represents observations in a bi-variate data set. Given a random sample of observations, the population regression line is estimated by

$$\hat{y} = a + bx$$

Where ,'a' is a constant, 'b' is the regression coefficient and 'x' is the value of the independent variable, and '$\hat{y}$' is the predicted value of the dependent variable. The least square method defines the estimate of these parameters as the values which minimize the sum of the squares between the measurements and the model.

$$E = \sum_i (Y_i - \hat{Y}_i)^2$$

The solutions of 'a' and 'b' are obtained by

$$a = \bar{Y} - b\bar{X}$$

$$b = \frac{\sum (Y_i - \bar{Y}_i)(X_i - \bar{X}_i)}{\sum (X_i - \bar{X}_i)}$$

Procedure to find parameter 'b' using regression approach

$$F(t) = \frac{1-e^{-bt}}{1+e^{-at}}$$

This c.d.f is equated to p_i Where $p_i = \dfrac{i}{n+1}$

The equation $F(t) = p_i$ is expressed as a linear form, $V_i = \sigma U_i$

Where, $V_i = X_i$

$$U_i = -\log\left(\frac{1-p_i}{1+p_i}\right)$$

$$\sigma = \frac{\sum V_i U_i - n\bar{V}\bar{U}}{\sum U_i^{\,2} - n\bar{U}^{\,2}}$$

Where $1/\sigma$ is the parameter b estimated through LSE approach.

5. DATA DESCRIPTION

To validate the model, the parameter estimation is carried out from the failure data sets from various reputed software testing centers from India. Five data sets were used here to compare the performance of the model.

6. RESULTS

Estimation of Parameters of the data sets

Data set	***1***	***2***	***3***	***4***	***5***
n	40	32	45	75	35
σ	42.7501	36.5810	51.5503	53.2140	35.2420
$\hat{b}$	0.02339	0.02734	0.01940	0.01879	0.02838
$\hat{a}$	39.2414	37.5614	56.4551	49.7711	36.0221
$m(t_n)$	38.2621	36.2463	61.3501	54.4521	36.5405
$\log L$	-45.2210	-39.0124	-76.2541	-69.5265	-49.5874

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 it to the data?”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 data sets collected from real world failure data is given as below.

Data Set	Log L	Reliability t_n+50
1	-45.221	0.5920
2	-39.0124	0.8421
3	-76.2541	0.6425
4	-69.5265	0.9210
5	-49.5874	0.3412

7. CONCLUSION

To validate the proposed approach, the parameter estimation is carried out on the data sets collected from 5 software testing centers. Parameters of the model are estimated by MLE and least squares method using cumulative failure data against time. The reliability of the model over Data set 4 data after 50 units of time is high among the datasets which were considered for the analysis.

REFERENCES

Huang, C.Y and Kuo, S.Y., (2002). "Analysis of incorporating logistic testing effort function into software reliability modeling", *IEEE Transactions on Reliability*, Vol.51, No. 3, pp. 261-270.

Kapur, P.K., Gupta, D., Gupta, A. And Jha, P.C., (2008). "Effect of Introduction of Fault and Imperfect Debugging on Release Time", *Ratio Mathematica*, 18, pp. 62-90.

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

Musa, J.D., Iannino, A., and Okumoto, K. (1987). "*Software Reliability: Measurement, Prediction, and Application*", New York: McGraw-Hill.

Pham. H., (2003). "*Handbook of Reliability Engineering*", Springer.

Pham. H., (2006). "*System software reliability*", Springer.

Prasad, S.R., Shaheen., and G. Krishna Mohan, (2013). Two Step Approach for Software Reliability: HLSRGM. *International Journal of Computer Trends and Technology*, 4, 3575-3579.

Wood, A., (1996). "Software Reliability Growth Models", Tandem *Computers, Technical report* 96.1.

Xie, M., Yang, B. and Gaudoin, O. (2001). "Regression goodness of-fit Test for Software Reliability Model Validation", ISSRE and Chillarege Corp.