Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 60 (2013) 16341-16344

ISSN: 2229-712X

A two step approach for parameter estimation of software reliability

R.Satya Prasad[1,*], S. Murali Mohan[2] and G.Krishna Mohan[3]

[1]Department of CS &Engineering, Acharya Nagrjuna University, Nagarjuna Nagar.

[2]Vikrama Simhapuri University, Nellore.

[3]Department of Computer Science, P.B.Siddhartha college, Vijayawada.

ARTICLE INFO

Article history:

Received: 16 May 2013;

Received in revised form:

24 June 2013;

Accepted: 9 July 2013;

Keywords

Exponential,

MLE,

LSE,

SRGM,

Goodness of fit,

Regression.

ABSTRACT

Software Reliability Growth Model (SRGM) is a mathematical model of how the software reliability improves as faults are detected and repaired. The performance of SRGM is judged by its ability to fit the software failure data. How good does a mathematical model fit to the data and reliability of software is presented in the current paper. The model under consideration is the, G-O model. A two step approach is used to estimate the model parameters by combination of Maximum Likelihood Estimation (MLE) and Least Square Estimation (LSE) methods. To assess the performance of the considered SRGM, we have carried out the parameter estimation on the real software failure data sets.

Introduction

Software reliability is defined as the probability of failure-free software operation for a specified period of time in a specified environment (Lyu, 1996) (Musa et al., 1987). SRGM is a mathematical model of how the software reliability improves as faults are detected and repaired. (Quadri and Ahmad, 2010). Among all SRGMs developed so far a large family of stochastic reliability models based on a non-homogeneous poisson process known as NHPP reliability models, has been widely used. Some of them depict exponential growth while others show S-shaped growth depending on nature of growth phenomenon during testing. The success of mathematical modelling approach to reliability evaluation depends heavily upon quality of failure data collected.

However, a problem is the model validation and selection. If the selected model does not fit the collected software testing data relatively well, we would expect a low prediction ability of this model and the decision makings based on the analysis of this model would be far from what is considered to be optimal decision (Xie et al., 2001) . The present paper presents a method for model validation.

Literature Survey

NHPP Models

The NHPP group of models provides an analytical framework for describing the software failure phenomenon during testing. They are proved to be quite successful in practical software reliability engineering (Musa, 1987). They have been built upon various assumptions. If ‘t’ is a continuous random variable with probability density function: $f(t,\theta_1,\theta_2,\ldots,\theta_k)$, and cumulative distribution function: $F(t)$.where $\theta_1,\theta_2,\ldots,\theta_k$ are k unknown constant parameters. The mathematical relationship between the pdf and cdf is given as: $f(t)=F'(t)$.

Let $N(t)$ be the cumulative number of software failures by time ‘t’. A non-negative integer-valued stochastic process $N(t)$ is called a counting process, if $N(t)$ represents the total number of occurrences of an event in the time interval [0, t] and satisfies these two properties:

1. If $t_1 < t_2$, then $N(t_1) \le N(t_2)$
2. If $t_1 < t_2$, then $N(t_2)-N(t_1)$ is the number of occurrences of the event in the interval $[t_1,t_2]$.

One of the most important counting processes is the Poisson process. A counting process, $N(t)$, is said to be a Poisson process with intensity λ if

1. The initial condition is $N(0) = 0$
2. The failure process, $N(t)$, has independent increments
3. The number of failures in any time interval of length s has a Poisson distribution with mean λs, that is,

$$P\{N(t+s)-N(t)=n\}=\frac{e^{-\lambda s}(\lambda s)^n}{n!}$$

Describing uncertainty about an infinite collection of random variables one for each value of ‘t’ is called a stochastic counting process denoted by $[N(t),t\ge 0]$. The process $\{N(t),t\ge 0\}$ is assumed to follow a Poisson distribution with characteristic MVF (Mean Value Function) $m(t)$, representing the expected number of software failures by time ‘t’. Different

Tele:

E-mail addresses: muraliradhikamohan@gmail.com

models can be obtained by using different non decreasing $m(t)$. The derivative of $m(t)$ is called the failure intensity function $\lambda(t)$.

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

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

Where, $m(t)$ is a 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. The mean value function $m(t)$ is the characteristic of the NHPP model.

The NHPP models are further classified into Finite and Infinite failure models. 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 $\lambda(t)$ is given by: $\lambda(t)=aF'(t)$ (Pham, 2006).

SRGM

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 behaviour 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.

Model description: G-O model

One simple class of finite failure NHPP model is the Goel and Okumoto model (Goel and Okumoto, 1979), which has an exponential growth of the cumulative number of failures experienced. It is an NHPP based SRGM, assuming that the failure intensity is proportional to the number of faults remaining in the software describing an exponential failure curve. It has two unknown parameters. Where, 'a' is the expected total number of faults in the code and 'b' is the shape parameter defined as, the rate at which the failure rate decreases. The cumulative distribution function of the model is: $F(t)=1-e^{-bt}$. The corresponding probability density function has the form: $f(t)=be^{-bt}$. The expected number of faults at time't' is the Mean Value Function and is denoted by $m(t)=a\left(1-e^{-bt}\right)$. The corresponding failure intensity function is given by $\lambda(t)=abe^{-bt}$. Where 't' can be calendar time (Krishna Mohan et al., 2012). Software reliability is defined as the probability of failure-free software operation for specified period of time 't' in a specified environment, $R(t)=e^{-(m(t_i)-m(t_{i-1}))}$.

Two step approach for parameter estimation

The main issue in the NHPP model is to determine an appropriate mean value function to denote the expected number of failures experienced up to a certain time point. Method of least squares (LSE) or maximum likelihood (MLE) has been suggested and widely used for estimation of parameters of mathematical models (Kapur et al., 2008). Non-linear regression is a method of finding a nonlinear model of the relationship between the dependent variable and a set of independent variables. Unlike traditional linear regression, which is restricted to estimating linear models, nonlinear regression can estimate models with arbitrary relationships between independent and dependent variables. The model proposed in this paper is a non-linear and it is difficult to find solution for nonlinear models using simple Least Square method. Therefore, the model has been transformed from non linear to linear. MLE and LSE techniques are used to estimate the model parameters (Lyu, 1996) (Musa et al., 1987). Sometimes, the likelihood equations are difficult to solve explicitly. In such cases, the parameters estimated with some numerical iterative methods (Newton Raphson method). On the other hand, LSE, like MLE, applied for small sample sizes and may provide better estimates (Huang and Kuo, 2002).

Algorithm for a 2-step approach of parameter estimation and data as best fit.

- Consider the Cumulative distribution function $F(t)$ and equate to p_i, i.e $F(t)=p_i$,

where $p_i=\frac{i}{n+1}$

- Express the equated equation $F(t)=p_i$ as a linear form, $y=mx+b$.
- Find model parameters of mean value function $m(t)$.

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

  - The initial number of faults $\hat{a}$ is estimated through MLE method. Since, it forms a closed solution.
  - The remaining parameters are estimated through LSE regression approach.
- Find the failure intensity function $\lambda(t)=aF'(t)$
- Find likelihood function L
- Find the Log likelihood function log L. (Which comes to be –ve value.)
- The distribution model with the highest –ve value is the best fit.

ML (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, $t_1,t_2,\ldots,t_N$. The likelihood function (Pham, 2003) may be given by the following product:

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

Likelihood function by using λ(t) is:

$$L = \prod_{i=1}^{n} \lambda(t_i)$$

Log Likelihood function for ungrouped data (Pham, 2006),

$$\log L = \log\left(\prod_{i=1}^{n} \lambda(t_i)\right)$$

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

The maximum likelihood estimators (MLE) of $\theta_1,\theta_2,\ldots,\theta_k$ are obtained by maximizing L or Λ, where Λ is ln L . By maximizing Λ, which is much easier to work with than L, the maximum likelihood estimators (MLE) of $\theta_1,\theta_2,\ldots,\theta_k$ are the simultaneous solutions of k equations such as: $\frac{\partial(\Lambda)}{\partial\theta_j} = 0$, j=1,2,…,k.The parameters 'a' and 'b' are estimated as follows. The parameter 'b' is estimated by iterative Newton Raphson Method using $b_{n+1} = b_n - \frac{g(b_n)}{g'(b_n)}$, which is substituted in finding 'a'.

LS (Least Square) estimation

LSE is a popular technique and widely used in many fields for function fit and parameter estimation (Liu, 2011). 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*.

The Least Squares Regression Line

Linear regression finds the straight line, called the least squares regression line that best represents observations in a bivariate data set. Given a random sample of observations, the population regression line is estimated by: $\hat{y} = bx + a$. 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. Which amounts to minimizing the expression: $E = \sum_i \left(Y_i - \hat{Y}_i\right)^2$.

Taking the derivative of E with respect to 'a' and 'b' and equating them to zero gives the following set of equations (called the normal equations):

$$\frac{\partial E}{\partial a} = 2Na + 2b\sum X_i - 2\sum Y_i = 0$$, and

$$\frac{\partial E}{\partial b} = 2b\sum X_i^2 + 2a\sum X_i - 2\sum Y_i X_i = 0$$

The solutions of 'a' and 'b' are obtained by solving the above equations. Where, $a = \bar{Y} - b\bar{X}$ and $b = \frac{\sum\left(Y_i - \bar{Y}\right)\left(X_i - \bar{X}\right)}{\sum\left(X_i - \bar{X}\right)^2}$.

Illustration

ML Estimation

The likelihood function of G-O model is given as, $L = \prod_{i=1}^{N} abe^{-(bt)}$

Taking the natural logarithm on both sides, The Log Likelihood function is given as: $\log L = \sum_{i=1}^{n} \log(abe^{-(bt_i)}) - a[1-e^{-(bt_n)}]$.

Taking the Partial derivative with respect to 'a' and equating to '0'. (i.e $\frac{\partial \log L}{\partial a} = 0$).

$$a = \frac{n}{\left[1-e^{-(bt_n)}\right]}$$

LS Estimation for parameter 'b' using regression approach

The cumulative distribution function of G-O model is, $F(t) = 1 - e^{-\left(x_i/\sigma\right)}$. The c.d.f is equated to p_i. Where, $p_i = \frac{i}{n+1}$.

The equation $F(t) = p_i$ is expressed as a linear form, $Y_i = CX_i + D$. Where, $Y_i = \log\left(-\log\left(1-p_i\right)\right)$; $X_i = \log\left(x_i\right)$; $D = -\log\sigma$.

The parameter D is estimated as, $\hat{D} = \bar{Y} - \bar{X}$ and therefore, $\hat{\sigma} = e^{-\hat{D}}$. Where, '$1/\sigma$' is nothing but the parameter 'b' estimated through regression approach.

Table 2: estimate of parameters and log likelihood for, Xie data

n=	30
$\hat{D}$ =	-2.780344
σ =	16.124569
$\hat{b}$ =	0.062017
$\hat{a}$ =	31.899246
$m(t_n)$ =	31.698171
$Log\,L$ =	-15.595950

Table 3: Parameters estimated through MLE and LSE

Data Set	Number of samples	Estimated Parameters		
		a (using MLE)	b (using MLE)	b (using LSE)
Xie	30	31.899246	0.003819	0.062017
AT&T	22	23.582254	0.003973	0.054681
IBM	15	17.608791	0.006451	0.033428
NTDS	26	30.168926	0.007917	0.122495
SONATA	30	68.291239	0.000316	0.010700

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 (for example a linear regression model) fit to the data?". Inorder to validate the model under study and to assess its performance, experiments on a set of actual software failure data have been performed. 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.

Table 4: Log likelihood on different data sets

Data Set (No)	Log L (MLE)	Log L (Two step approach)	Reliability t_n+50 (MLE)	Reliability t_n+50 (Two step approach)
Xie	-120.617374	-15.606746	0.718799	0.999422
NTDS	-82.959757	-1.175846	0.255979	0.999566
AT&T	-94.518262	-20.830877	0.751991	0.00000000006
SONATA	-153.527000	-56.783483	0.548818	0.000101
IBM	-59.554687	-22.381117	0.487115	0.023403

Conclusion

To validate the proposed approach, we have carried out the parameter estimation on the data sets collected from (Xie et al.,2002; Pham, 2006; Ashoka, 2010). Parameters of the model are estimated by MLE and the linear regression least squares method using cumulative failure data against time. Out of the data sets that were collected, the model under consideration best fits the data of SONATA using both MLE and Two step approach. Since, it is having the highest negative value for the log likelihood. The reliability of the model using both MLE method and Two step method is given in table 4.

References

[1] Ashoka. M.,(2010), "Sonata Software Limited" Data Set, Bangalore.

[2] Goel, A. L. and Okumoto, K., 1979, 'Time-dependent error-detection rate model for software reliability and other performance measures', IEEE Transactions on Reliability, vol. 28, pp. 206-211.

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

[4] 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.

[5] Kapur, P.K., Sunil kumar, K., Prashant, J. and Ompal, S. (2009). "Incorporating concept of two types of imperfect debugging for developing flexible software reliability growth model in distributed development environment", Journal of Technology and Engineering sciences, Vol.1, No.1, Jan-Jun.

[6] Krishna Mohan, G., Srinivasa Rao, B. and Satya Prasad, R. (2012). "A Comparative study of Software Reliability models using SPC on ungrouped data", International Journal of Advanced Research in Computer Science and Software Engineering. Volume 2, Issue 2, February.

[7] Liu, J., (2011). "Function based Nonlinear Least Squares and application to Jelinski-Moranda Software Reliability Model", stat. ME, 25th August.

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

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

[10] Pham. H., 2003. "Handbook Of Reliability Engineering", Springer.

[11] Pham. H., 2006. "System software reliability", Springer.

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

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

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

[15] Xie, M., Goh. T.N., Ranjan.P., (2002). "Some effective control chart procedures for reliability monitoring" -Reliability engineering and System Safety 77 143 -150.

Dr. R Satya Prasad received Ph.D. degree in Computer Science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He have a satisfactory consistent academic track of record and received gold medal from Acharya Nagarjuna University for his outstanding performance in a first rank in Masters Degree. He is currently working as Associate Professor in the Department of Computer Science & Engineering, Acharya Nagarjuna University. He has occupied various academic responsibilities like practical examiner, project adjudicator, external member of board of examiners for various Universities and Colleges in and around in Andhra Pradesh. His current research is focused on Software Engineering, Image Processing & Database Management System. He has published several papers in National & International Journals.

MURALI MOHAN. S, working as controller of examinations in Vikrama Simhapuri University. He worked as a controller of examinations in Dravidian University for three and half years. He worked as a Principal of an affiliated college of Andhra University in Visakhapatnam (A.P). He is presently a research scholar in the Department of Computer Science of Acharya Nagarjuna University.

Mr. G. Krishna Mohan, working as a Reader in the Department of Computer Science, P.B.Siddhartha College, Vijayawada. He obtained his M.C.A degree from Acharya Nagarjuna University, M.Tech from JNTU, Kakinada, M.Phil from Madurai Kamaraj University and pursuing Ph.D from Acharya Nagarjuna University. He qualified AP State Level Eligibility Test. His research interests lies in Data Mining and Software Engineering. He published 14 research papers in various National and International journals.