Estimation of Failure Count Data Using Confidence Interval

R. Satya Prasad, V. Goutham, N. Pawan Kumar

Abstract*— A confidence interval gives an estimated range of values which is likely to include an unknown population parameter, the estimated range being calculated from a given set of sample data. Confidence intervals are usually calculated so that this percentage is 95%, Confidence limits are the lower and upper boundaries / values of a confidence interval, that is, the values which define the range of a confidence interval. The upper and lower bounds of a 95% confidence interval are the 95% confidence limits. The method described here is based on Goel Okomoto (GO) model, Confidence Interval (CI) and parameter estimation is maximum likehood (ML). It uses historical sample test data to predict how many residual defects are there in the software system and the estimated range being calculated from a given set of sample data to achieve at least 95% confidence level.

Index Terms*— Confidence interval(CI) , Failure intensity funtion, Goel-Okumoto model(GO), Interval Estimation, Maximum likelihood estimator (MLE), Paramenter estimation,Sof twar Reliability.

I. INTRODUCTION

Software Reliability is the probability that software product will function failure free for a specified period of time in a specified Environment. Software reliability studies the failure process that occur during the execution of program."Software failure" is defined as the departure of the external result of a program from its requirement [5].The term failure refers to state of the program during execution,which means that a program has to be executed inorder to a failure to occur.The term "failure" must be distinguished from the term "fault".Fault is a defect in a program that when executed under special conditions,causes a failure.Software Relaibility models study the behaviour of the failure process by which inferences can be made concerning the quality of the given program[4].Software modeling from are in two ways estimation which is a statistical estimation method applied to failure data collected from the program.This can be done after a program has executed long enough so that failure data are avaliable.Prediction determinate is from properties of a software program.This can be done before any program execution occurs.The uncertainity involved in the determination of a specific form is expressed interms of confidence interval for the value of a parameter. A point estimate is sometimes inadequate in providing an estimate of an unknown parameter, since it rarely coincides with the true value of the parameter.

Manuscript received May, 2013.

Dr. R. Satya Prasad, Associate Professor, Department of Computer Science & Engineering, Acharya Nagarjuna University, Guntur, (A.P.), India.

V. Goutham, Associate Professor, Department of Computer Science and Engineering at St.Mary Group of institutions, Hyderabad, India.

N. Pawan Kumar, Assistance Professor, Department of Computer Science and Engineering at StMary Group of institutions, Hyderabad, India.

An alternative way is to obtain a confidence interval estimation of the form[θL, θu] where θL is the lower bound and θu is upper bound. A confidence interval for a mean specifies a range of values within which the unknown population parameter [3].

II. BACKGROUND

Goel and okumoto[1] described failure detection as a non-homogeneous poission process(NHPP) with an exponentially decaying rate function.It is a simple non-homogeneous poisson process model.

The Goel-Okumoto model has the following

Mean value function $\mu(t)= a(1-e^{-bt})$ ----------(1)

Failure intensity function $\lambda(t)=abe^{-bt}$ ----------(2)

The parameter 'a' is interpreted as the number of initial faults in the software and the parameter 'b' is related to the reliability growth rate of testing process.The software reliability R(x/t) is defined as the probability of a failure free operations of a complete software for a specified time interval(t,t+x) in a specified environment.

$$R(x/t)=Exp\{-[\mu(t+x)- \mu(t)]\}$$ ----------(3)

The most common method for the estimation of parameters is the Maximum Likelihood (ML) method.ML estimation of a broad collection of software reliability for grouped data are discussed in detail [2]

The paramenter 'a' can be estimated using ML method based on the number of failures per interval.Suppose that an observation interval {0,tk} is divided into set of subintervals (0,ti],(t1,t2]……(tk-1,tk] , the number of failures per subinterval is recorded as ni(i=1,2,3..,k) with respect to the number of failures in(ti-1,ti][3].

The Log Likelihood function(LLF)

Log L =

$$\sum_{i=1}^{n}\left(y_i - y_{i-1}\right).\log\left[m(t_i)-m\left(t_{i-1}\right)\right]-m(t_n)$$ ----------(4)

ML (Maximum Likelihood) Parameter Estimation

Parameter estimation is of primary importance in software reliability prediction. Once the analytical solution for $m(t)$ is known for a given model, parameter estimation is achieved by applying a technique of Maximum Likelihood Estimate (MLE). Depending on the format in which test data are available, two different approaches are frequently used. A set of failure data is usually collected in one of two common ways, time domain data and interval domain data. The idea behind maximum likelihood parameter estimation is to determine the parameters that maximize the probability (likelihood) of the sample data. The method of maximum likelihood is considered to be more robust (with some exceptions) and yields estimators with good statistical properties. In other words, MLE methods are versatile and apply to most models and to different types of data. Although the methodology for maximum likelihood estimation is

simple, the implementation is mathematically intense[7]. Assuming that the data are given for the cumulative number of detected errors yi in a given time-interval $(0,t_i)$ where i = 1,2, …, n. and $0 < t_1 < t_2 <\ldots< t_n$ then the log likelihood function (LLF) takes on the following form.

Likely hood function by using λ(t) is:$L = \prod_{i=1}^{n} \lambda(t_i)$

The logarithmic likelihood function for interval domain data [7] is given by:

$$\text{LogL}= \sum_{i=1}^{n}(y_i - y_{i-1}).\log\left[m(t_i) - m(t_{i-1})\right] - m(t_n)$$

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 maximizingΛ, 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 that: $\frac{\partial(\Lambda)}{\partial\theta_j} = 0$, j=1,2,…,k

The parameters 'a' and 'b' are estimated using iterative Newton Raphson Method, which is given as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

ILLUSTRATING THE MLE METHOD

To estimate 'a' and 'b' , for a sample of n units, first obtain the likelihood function:

$$L = \prod_{i=1}^{n} abe^{-bt}$$

Take the natural logarithm on both sides, The Log Likelihood function is given as:

$$\text{Log L} = \log[\prod_{i=1}^{n}\lambda(t_i)] = \log[\prod_{i=1}^{n} abe^{-bt}] = \sum_{i=1}^{n}(y_i - y_{i-1}).\log\left[a\left(1-e^{-bt_i}\right) - a\left(1-e^{-bt_{i-1}}\right)\right] - a\left(1-e^{-bt_n}\right)$$

The parameter 'a' is estimated by taking the partial derivative w.r.t 'a' and equating to '0'.

$$(\text{i.e } \frac{\partial \log L}{\partial a} = 0) \qquad a = \frac{(y_n - y_0)}{\left(1-e^{-b\,tn}\right)}$$

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'. where $g(b) \& g'(b)$ are expressed as follows.

$$g(b) = \frac{\partial \log L}{\partial b} = 0$$

$$g'(b) = \frac{\partial^2 \log L}{\partial b^2} = 0$$

$$g(b) = \sum_{i=1}^{n}(y_i - y_{i-1})\frac{\left(t_i e^{-bt_i} - t_{i-1}e^{-bt_{i-1}}\right)}{e^{-bt_{i-1}} - e^{-bt_i}} - \frac{(y_n - y_0)t_n e^{-bt_n}}{\left(1-e^{-bt_n}\right)}$$

$$g'(b) = \frac{(y_n - y_0)t_n^{\,2}e^{-bt_n}}{\left(1-e^{-bt_n}\right)^2} - \sum_{i=1}^{n}(y_i - y_{i-1})\frac{e^{-bt_i}e^{-bt_{i-1}}\left(t_i - t_{i-1}\right)^2}{\left(e^{-bt_{i-1}} - e^{-bt_i}\right)^2}$$

Distribution of Time between failures

Based on the inter failure data given in Table 1, we can compute the software failures process through Mean Value. We used cumulative time between failures data for software reliability using Goel-Okumoto model[2].

The Proposed Model

The available data from exponential distribution are grouped ,the modified maximum likelihood estimator (MLE) for the mean has been implemented.In this paper we implemented a confidence interval(CI) for the mean based on grouped data.

On lines of Goel and Okumoto (1979)[1], let us specify that the mean value function m(t) is finite valued, non decreasing, non negative and bounded with the boundary conditions

$$m(t) = \begin{cases} 0, & t = 0 \\ a, & t \to \infty \end{cases}$$

Here 'a' represents the expected number of software failures eventually detected. 'b' is a positive constant, serving the purpose of constant of proportional fall in $\lambda(t)$. This relation indicates a decreasing trend for $\lambda(t)$ with increase in $m(t)$ - a characteristic similar to that of LPETM which requires exponential decrease of $\lambda(t)$ with increase in *m(t)*[2]

Confidence interval estimation

Considering the likelihood function say L(a, b) we can get confidence interval for L(a, b).The procedure of interval estimation described above can be adopted by using the exact MLEs of 'a', 'b' and the corresponding estimated variances, covariance of the MLEs with the help of equations(2)

We may recall that the mean value function and intensity function is given by

$$m(t) = a\left(1-e^{-bt}\right) \qquad \text{a>0,b>0,t>=0} \qquad \text{--------- (5)}$$

The constants 'a', 'b' which appear in the mean value function and hence in NHPP, in intensity function (error detection rate) and various other expressions are called parameters of the model. In order to have an assessment of the software reliability 'a',' b' are to be known or they are to be estimated from a software failure data.

Suppose we have 'n' time instants at which the first, second, third..., n^{th} failures of a software are experienced. In other words if S_k is the total time to the k^{th} failure, s_k is an observation of random variable S_k and 'n' such failures are successively recorded.

Accordingly 'a', 'b' would be solutions of the equations

$$\frac{\partial \log L}{\partial a} = 0\,,\frac{\partial \log L}{\partial b} = 0\,.$$

Substituting the expressions for m(t), λ(t) given by equations (5) and (2) in equation (4), taking logarithms, differentiating with respect to 'a', 'b' and equating to zero, after some joint simplification we get

$$a = \frac{(y_n - y_0)}{\left(1 - e^{-b\,tn}\right)} \quad \text{----------(6)}$$

$$g(b) = \sum_{i=1}^{n} (y_i - y_{i-1}) \frac{\left(t_i e^{-bt_i} - t_{i-1} e^{-bt_{i-1}}\right)}{e^{-bt_{i-1}} - e^{-bt_i}} - \frac{(y_n - y_0) t_n e^{-bt_n}}{\left(1 - e^{-bt_n}\right)} \quad \text{-------(7)}$$

In order to get the asymptotic variances and co-variance of the MLEs of 'a', 'b' we needed the elements of the information matrix obtained through the following second order partial derivatives.

$$\frac{d^2(\log L)}{da^2} = \frac{\left(1 - e^{-bt_n}\right)^2}{(y_n - y_0)} \quad \text{----------(8)}$$

$$\frac{d2}{dadb}(\log L) = t_n\, e^{-btn} \quad \text{---------- (9)}$$

$$\frac{d^2}{db^2}(logL) = [\ \frac{e^{-bt_n}\ (y_n - y_0)\ t_n^2}{(1 - e^{-bt_n})^2} \ -- \ \frac{(y_n - y_0)\,(t_i - t_{i-1})^2 . e^{-bt_i} . e^{-bt_{i-1}}}{(e^{-bt_{i-1}} - e^{-bt_i})^2} \ ] \quad \text{-------- (10)}$$

Expected values of negatives of the above three derivatives

would be the following information matrix

$$E\begin{bmatrix} -\frac{\partial^2 \log L}{\partial a^2} & -\frac{\partial^2 \log L}{\partial a \partial b} \\ -\frac{\partial^2 \log L}{\partial a \partial b} & -\frac{\partial^2 \log L}{\partial b^2} \end{bmatrix} \text{----------(11)}$$

Inverse of the above matrix is the asymptotic variance covariance matrix of the MLEs of 'a',' b'. Generally the above partial derivatives evaluated at the MLEs of 'a', 'b' are used to get consistent estimator of the asymptotic variance covariance matrix. However in order to overcome the numerical iterative way of solving the log likelihood equations and to get analytical estimators rather than iterations[6].

To obtain the confidence limits for the parameter a and b ,we can calculate the Fisher information matrix to obtain the asymptotic variance and covariance of the ML estimation of the parameter.The asymptotic variance co-variance matrix of the parameter a and b is obtain by inverting the Fisher information matrix.[7].

Confidence intervals for parameter estimates *a* and b for the log-likelihood function given in given equation

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

we can obtain Fisher information Matrix

$$F=E\begin{bmatrix} \frac{-\partial^2 LogL}{\partial a^2} & \frac{-\partial^2 LogL}{\partial a\,\partial b} \\ \frac{-\partial^2 LogL}{\partial a\,\partial b} & \frac{-\partial^2 LogL}{\partial b^2} \end{bmatrix}$$

The variance matrix, *V*, can be obtained as follows:

$$v=F^{-1} = \begin{bmatrix} Var(a) & Cov(a,b) \\ Cov(a,b) & Var(b) \end{bmatrix}$$

$$= \frac{1}{\frac{\partial^2 LogL}{\partial a^2}\frac{\partial^2 LogL}{\partial b^2} - \left(\frac{\partial^2 LogL}{\partial a \partial b}\right)^2} \begin{bmatrix} \frac{-\partial^2 LogL}{\partial b^2} & \frac{\partial^2 LogL}{\partial a \partial b} \\ \frac{\partial^2 LogL}{\partial a \partial b} & \frac{-\partial^2 LogL}{\partial a^2} \end{bmatrix} \text{----(12)}$$

The Var(a),Var(b),Cov(a,b) can be easily obtained from the above function

$$\text{Var}(m(t_i)) = \left(\frac{dm}{da}\right)^2 var(\hat{a}) + 2\ \frac{dm}{da}\frac{dm}{db}\ \text{cov}(\widehat{ab}) + \left(\frac{dm}{db}\right)^2 var(\hat{b}) \quad \text{----------(13)}$$

Confidence interval for the mean value function m(ti) are given as

$$m_{\,u}(t_i) = \hat{m}\ t_i + z_{\,\alpha}\ \sqrt{var(m(t_i))} \text{--------(13)}$$

$$m_{\,l}(t_i) = \hat{m}\ t_i - z_{\,\alpha}\ \sqrt{var(m(t_i))} \text{--------(14)}$$

where $z_{\,\alpha}$ is the (1-α) quartile of the standard normal distribution[8].

Software Error Data Analysis: The interval methods of estimation are explained by applying the results to the software failure data .The set of software errors analyzed here is borrowed from a Real-time command and control Data(in an 1 hour interval) as published in [7].The data are named as Table 1 test data.

Table 1: Test Data

Week index	Number of Failure	Cum.fault(n_i)
1	27	27
2	16	43
3	11	54
4	10	64
5	11	75
6	7	83
7	2	84
8	5	89
9	3	92
10	1	93
11	4	97
12	7	104
13	2	106
14	5	111
15	5	116
16	6	122
17	0	122
18	5	127
19	1	128
20	1	129
21	2	131
22	1	132
23	2	134
24	1	135
25	1	136

The parameters a and b are estimated.For the dataset in above table the likelihood function is given by

$$\text{Log L} = \sum_{i=1}^{n} (y_i - y_{i-1}).\log\left[m(t_i) - m(t_{i-1})\right] - m(t_n)$$

iate with respect to a anb b ,the first derivation of bove e taken e following Equations will be

$$\frac{d}{da}(\log L) = \frac{\sum_{i=1}^{25} y_i}{a} - \left(1 - e^{-bt_n}\right)$$

$\frac{d}{db}(\log L) = \sum_{i=1}^{25} \frac{(y_n - y_0)(t_i - t_{i-1}) \cdot e^{-bt_i} \cdot e^{-bt_{i-1}}}{(e^{-bt_{i-1}} - e^{-bt_i})}$

$$- \frac{e^{-bt_n} (y_n - y_0) t_n}{1 - e^{-bt_n}}$$

Solve The above equation estimate as follows Therefore the G-O model is given by

$m(ti) = 124.531(1-^{12.ti})$

The second partial derivations of the log-Likehood function are taken and the results given as follows

$$\frac{d^2}{da^2}(logL) = \frac{(1 - e^{-bt_n})^2}{(y_n - y_0)} = 0.007029$$

$\frac{d^2}{db^2}(logL)$ =[$\frac{e^{-bt_n} (y_n - y_0) t_n^2}{(1 - e^{-bt_n})^2}$ -- $\frac{(y_n - y_0)(t_i - t_{i-1})^2 . e^{-bt_i} . e^{-bt_{i-1}}}{(e^{-bt_{i-1}} - e^{-bt_i})^2}$]= 6004.498

$\frac{d^2(\log L)}{dadb} = t_n . e^{-bt_n}$= 3.118

Therefore the variance are given by

$var(\hat{a})$ = 184.857936

$var(\hat{b})$ = 0.000216 and

$var(\hat{a}\hat{b})$ = 0.095991

Taking the derivation of the mean value function m(t25) of the G-O model and get

$\frac{dm}{da} = 1 - e^{-bt25}$ =0.8753

$\frac{dm}{db} = at_{25} e^{-bt25}$ = 388.29

Asymptotic variance can be derived accroding to equation (12)(13)(14)

var(m(t_{25}))= $(1 - e^{-bt}25)^2 var(\hat{a})$ + 2 $(1 - e_{25}^{-bt})$ a.t_{25} $e^{-bt25} . cov(\hat{a}\hat{b})$ + $a.(1 - e^{-bt}25)^2 var(\hat{b})$

=239.492516

Confidence interval for the mean value function m(ti) are given as

$m_u(t_i) = \hat{m}\ t_i + z_\alpha \sqrt{var(m(t_i))}$.= 139.177314

$m_l(t_i) = \hat{m}\ t_i - z_\alpha \sqrt{var(m(t_i))}$ = 78.822686

UpperBoud=139.177314 and LowerBound=78.822686

Total Length=60.3546 (UB-LB)

The Expected number of failure is given by

m(t25) = 124.531(1- $e^{-0.0832.t25}$)

=109.0

The estimated values of 'a' indicate that our model is under estimating the number of faults. The estimated mean value function and the 95% confidence bounds of m(t) for the actual data are given above.

Conclusion:

The data available from an exponential distribution are grouped and the model(Goel –Okumoto) is used to illustrate the parameter estimation problem. An Asymptotic property of a statistic which is used to construct an approximate the confidence interval for the mean. In this paper chart can be developed based on Fisher information matrix.Associated with NHPP software Reliability and the confidence interval for the model are derived.Confidence interval computation is studied in the Goel Okumoto Model.The upper and Lower bounds of the paramenters can be obtained. Further Control chart based on NHPP software reliability model can be proposed and implemented to detect the change in the failure process.

ACKNOWLEDGMENT

I also would like to say thanks to CSE Department of Nagarjuna university and extend my gratitute to my professors ,Finally, thanks to my family for their love, understanding, and encouragement throughout the study.

References

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

[2] Knafl, G.J.(1992),Solving maximum likelihood equations for two-parameter software reliability models using grouped data. Proceedings., Third International Symposium on Software Reliability Engineering (205-213).

[3] M. Kimura, S. Yamada, S. Osaki, "Statistical Software reliability prediction and its applicability based on mean time between failures". Mathematical and Computer Modelling, 1995, Volume 22, Issues 10-12, Pages 149-155.

[4] Lyu M.R(1996),Handbook of software Reliability Engineering,McGraw-Hill,NewYork.

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

[6] M. Ohba, "Software reliability analysis model". IBM J. Res. Develop. 1984, 28, 428-443.

[7] System software reliability, Springer, 2006, H. Pham.

[8] Xie, M and Wohlin, C., 1997. "A Practical Method for the Estimation of Software Reliability Growth in the Early Stage of Testing". IEEE Computer Society

First Author: Dr. R. Satya Prasad received Ph.D. degree in Computer Science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Andhra Pradesh. He received gold medal from Acharya Nagarjuna University for his outstanding performance in Masters Degree. He is currently working as Associate Professor in the Department of Computer Science & Engineering, Acharya Nagarjuna University. His current research is focused on Software Engineering. He has published several papers in National & International Journals.

Second Author: V. Goutham is an Associate Professor in the Department of Computer Science and Engineering at StMary Group of institutions affiliated to J.N.T.U Hyderabad. He received M.Tech from Andhra University and B.Tech from J.N.T.U. He Worked for various MNC Companies in Software Testing and Quality as Senior Test Engineer. His research interests are Software Engineering and Data Mining.

Third Author: N. Pawan Kumar is an Assistence Professor in the Department of Computer Science and Engineering at StMary Group of institutions affiliated to J.N.T.U Hyderabad. He received M.Tech from J.N.T.U hyderabad. His research interests are Software Engineering and Design of Algorithm.