Interval Estimation in Software Reliability Analysis

G.Y. Hong, M. Xie,

National University of Singapore, Singapore

M. Zhao and C.Wohlin

Linkoping University, Sweden

Abstract

Software reliability is traditionally estimated by analyzing failure data collected during software testing. There are many software reliability models available, but the estimation of model parameters usually requires a large number of failure data which might not be available. Hence the estimated parameters are not accurate and frequent revisions are needed as more failure data become available. In this paper, we study the use of interval estimation in software reliability prediction. For a commonly used software reliability model, we present the interval estimates of the parameters and their uses for a better planning during reliability testing. Confidence limits for the reliability and failure intensity function are presented. The results are useful, for example, in the determination of software release time which is a difficult problem in practice.

Key Words: *Software Reliability, Interval estimation, Parameter estimation, failure intensity function, the Goel-Okumoto model*

1. Introduction

One basic problem in software development is that software systems have become more complex and larger than ever. This is due to the complexity of the tasks which the software must perform. It is important to produce reliable software systems since software failures may cause the breakdown of computer system that can result in tremendous loss to the society. Several software reliability models have been proposed to estimate and predict software reliability during the testing phase, see Xie(1991) and Lyu(1996).

In the recent years, probabilistic models are employed to predict the reliability level achieved in the software so that the developers can predict the amount of testing needed and estimate the release time at which the reliability target is met. Software reliability is traditionally evaluated using a point estimate which is a single numerical value that results from the analysis of the failure data collected during software testing. However, most of the studies assume that the parameters of the underlying software reliability models are known accurately. This may not be true since the estimation of the model parameters are based on the number of failures discovered during the testing, and the estimation of model parameters are usually different according to different estimation methods. Furthermore, an accurate parameter estimation requires a large number of failure data which might not be available. Hence the estimated parameters are not accurate and they may change frequently when more failure data are collected.

To solve this problem, we present in this paper interval estimates of software reliability model parameters. This result is useful for a better planning during reliability testing. In Section 2, the Goel-Okumoto model and the estimation problems are discussed. In Section 3, the interval estimation methods of the model parameters are proposed. Confidence limits for the reliability and failure intensity function are presented. An example is given to illustrate the method in Section 4.

2. The Goel-Okumoto Model and the Estimation Problem

A software reliability growth model characterizes how the reliability of that software varies with execution time. These models have gained considerable acceptance in software reliability analysis, but the accuracy of each model varies and no single model is superior above all the other models(Azem 1995). For illustration, Goel-Okumoto model is used here in the analysis. The Goel-Okumoto model is a simple non-homogeneous Poisson process (NHPP) model. Similar analysis can be carried out using other reliability models.

The Goel-Okumoto model has the following mean value function $\mu(t)$ and failure intensity function $\lambda(t)$:

$$\mu(t) = a(1 - e^{-bt}). \tag{1}$$

$$\lambda(t) = abe^{-bt} \tag{2}$$

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 the testing process. The software reliability $R(x|t)$ is defined as the probability of a failure-free operation of a computer software for a specified time interval $(t,t+x]$ in a specified environment. We have

$$R(x|t) = \exp\{-[\mu(t+x) - \mu(t)]\} \tag{3}$$

The most common method for the estimation of the parameters is the maximum likelihood (ML) method. Detailed discussion about ML method can already be found in the literature. For example, in Knafl (1992), ML estimates of a broad collection of software reliability models for grouped data are discussed in detail. In Knafl (1996), for ungrouped data set, the ML estimates are summarized. Because of the random nature of testing data, there are some problems with the stability especially for grouped software failure data.

The parameters of (1) can be estimated using the maximum likelihood method based on numbers of failures per interval. Suppose that an observation interval $(0, t_k]$ is divided into a set of subintervals $(0, t_1], (t_1, t_2], \ldots, (t_{k-1}, t_k]$, the number of failures per subinterval is recorded as $n_i (i = 1,2,\ldots,k)$ with respect to the number of failures in $(t_{i-1}, t_i]$. The likelihood function is

$$L(n_1, \ldots n_k) = \prod_{i=1}^{k} \frac{\{\mu(t_i) - \mu(t_{i-1})\}^{n_i}}{n_i\,!} \exp\{-[\mu(t_i) - \mu(t_{i-1})]\} \tag{4}$$

To solve equation (4), we take the logarithm of both sides:

$$\begin{aligned} \ln L &= \sum_{i=1}^{k} \ln\{\frac{[\mu(t_i) - \mu(t_{i-1})]^{n_i}}{n_i\,!} \exp[-(\mu(t_i) - \mu(t_{i-1}))]\} \\ &= \sum_{i=1}^{k} \{n_i \ln[\mu(t_i) - \mu(t_{i-1})] - [\mu(t_i) - \mu(t_{i-1})] - \ln n_i\,!\} \end{aligned} \tag{5}$$

For the Goel-Okumoto model, in order to estimate the parameter a and b, we can take the derivative of lnL with respect to a and b. To find the location of the maximum, we compute

$$\begin{cases} \dfrac{\partial \ln L}{\partial a} = \dfrac{\sum_{j=1}^{k} n_i}{a} + e^{-bt_k} - e^{-bt_0} \\ \dfrac{\partial \ln L}{\partial b} = \sum_{i=1}^{k} (\dfrac{n_i}{e^{-bt_{i-1}} - e^{-bt_i}} - a)(t_i e^{-bt_i} - t_{i-1} e^{-bt_{i-1}}) \end{cases} \tag{6}$$

Equating these derivatives to zero and solving the resulting equations for a and b, we find the estimates as follows:

$$\begin{cases} \hat{a} = \dfrac{\sum_{i=1}^{k} n_i}{1-e^{-\hat{b}t_k}} \\ \sum_{i=1}^{k}\left(\dfrac{n_i}{e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_i}} - \dfrac{\sum_{i=1}^{k} n_i}{1-e^{-\hat{b}t_k}}\right)(t_i e^{-\hat{b}t_i} - t_{i-1}e^{-\hat{b}t_{i-1}}) = 0 \end{cases} \tag{7}$$

The value of parameter a can be obtained using the first equation of (7) after getting the estimate of parameter b. Since the second equation of (7) for estimating parameter b is nonlinear, we can not find an analytic solution and must obtain it numerically.

In practice, we need a large number of failure data for the estimates to be accurate. In an early stage of testing, the estimates are usually not accurate at all and hence the estimated release time may not be accurate either. As this is the case, a confidence interval estimation has more advantages than a point estimation only.

3. Interval Estimation of the Parameters

In this section, we discuss the interval estimation of the parameters of the Goel-Okumoto model. In order to obtain the confidence limits for parameter a and b, we can calculate the Fisher information matrix to obtain the asymptotic variances and covariance of the ML estimates of the parameter. The Fisher information matrix is symmetric. It uses the matrix of negative second partial derivatives of the log likelihood function which is

$$F = \begin{bmatrix} -\partial^2 \ln L / \partial^2 a & -\partial^2 \ln L / \partial a \partial b \\ -\partial^2 \ln L / \partial a \partial b & -\partial^2 \ln L / \partial^2 b \end{bmatrix}. \tag{8}$$

The asymptotic covariance matrix V of the ML estimators $\hat{a}$ and $\hat{b}$ is the inverse of the Fisher information matrix:

$$V = F^{-1} = \begin{bmatrix} Var(\hat{a}) & Cov(\hat{a},\hat{b}) \\ Cov(\hat{a},\hat{b}) & Var(\hat{b}) \end{bmatrix}$$
$$= \frac{1}{\frac{\partial^2 \ln L}{\partial^2 a}\frac{\partial^2 \ln L}{\partial^2 b} - (\frac{\partial^2 \ln L}{\partial a \partial b})^2} \begin{bmatrix} -\partial^2 \ln L / \partial^2 b & \partial^2 \ln L / \partial a \partial b \\ \partial^2 \ln L / \partial a \partial b & -\partial^2 \ln L / \partial^2 a \end{bmatrix} \tag{9}$$

Here the negative second partial derivatives of the log likelihood function can be derived using the ML estimator $\hat{a}$ and $\hat{b}$ as follows:

$$\begin{cases} (-\partial^2 \ln L / \partial^2 a)_{a=\hat{a}} = \sum_{j=1}^{k} n_i / \hat{a}^2 \\ (-\partial^2 \ln L / \partial^2 b)_{b=\hat{b}} = \sum_{i=1}^{k} n_i \frac{(t_i - t_{i-1})^2 e^{-\hat{b}(t_i + t_{i-1})}}{(e^{-\hat{b}t_{i-1}} - e^{-\hat{b}t_i})^2} - \hat{a}t_k^2 e^{-\hat{b}t_k} + \hat{a}t_0^2 e^{-\hat{b}t_0} . \\ (-\partial^2 \ln L / \partial a \partial b)_{a=\hat{a}, b=\hat{b}} = t_k e^{-\hat{b}t_k} - t_0 e^{-\hat{b}t_0} \end{cases} \tag{10}$$

Since $\lambda(t)$ is a continuous function of parameter a and b, its maximum likelihood estimate is $\hat{\lambda}(t) = \hat{a}\hat{b}e^{-\hat{b}t}$ which is called the invariance property of ML estimators. The estimate of $Var(\hat{\lambda})$ is obtained by

$$
\begin{aligned}
Var(\hat{\lambda}) &= (\partial\lambda / \partial a)^2{}_{a=\hat{a}}\, Var(\hat{a}) + (\partial\lambda / \partial b)^2{}_{b=\hat{b}}\, Var(\hat{b}) \\
&\quad + 2(\partial\lambda / \partial a)(\partial\lambda / \partial b)_{a=\hat{a},b=\hat{b}}\, Cov(\hat{a},\hat{b}) \\
&= \hat{b}^2 e^{-2\hat{b}t} Var(\hat{a}) + \hat{a}^2 e^{-2\hat{b}t}(1-\hat{b}t)^2 Var(\hat{b}) + 2\hat{a}\hat{b}e^{-2\hat{b}t}(1-\hat{b}t)Cov(\hat{a},\hat{b})
\end{aligned}
\tag{11}
$$

Similarly, the maximum likelihood estimate of the reliability function *R(x|t)* is $\hat{R}(x|t) = \exp\{\hat{a}(e^{-\hat{b}(t+x)} - e^{-\hat{b}t})]\}$. The asymptotic variance for $\hat{R}(x|t)$ is

$$
\begin{aligned}
Var(\hat{R}) &= (\partial R / \partial a)^2_{a=\hat{a}} Var(\hat{a}) + (\partial R / \partial b)^2_{b=\hat{b}} Var(\hat{b}) \\
&\quad + 2(\partial R / \partial a)(\partial R \partial b)_{a=\hat{a},b=\hat{b}}\, Cov(\hat{a},\hat{b}) \\
&= e^{2\hat{a}e^{-\hat{b}t_i}(e^{-\hat{b}x\tau}-1)} \{e^{-2\hat{b}t_i}(e^{-\hat{b}x}-1)^2 Var(\hat{a}) + \hat{a}^2[(t_i+x)e^{-\hat{b}(t_i+x)} - t_i e^{-\hat{b}t_i}]^2 Var(\hat{b}) \\
&\quad + 2\hat{a}e^{-\hat{b}t_i}(e^{-\hat{b}x}-1)[t_i e^{-\hat{b}t_i} - (t_i+x)e^{-\hat{b}(t_i+x)})Cov(\hat{a},\hat{b})\}
\end{aligned}
\tag{12}
$$

The two sided approximate $100\alpha\%$ confidence limits for the parameter *a* and *b* are

$$
\begin{aligned}
a_{upper} &= \hat{a} + Z_\alpha [Var(\hat{a})]^{1/2} \qquad a_{lower} = \hat{a} - Z_\alpha [Var(\hat{a})]^{1/2} \\
b_{upper} &= \hat{b} + Z_\alpha [Var(\hat{b})]^{1/2} \qquad b_{lower} = \hat{b} - Z_\alpha [Var(\hat{b})]^{1/2}
\end{aligned}
\tag{13}
$$

where Z_α is the $(1-\alpha)$ quartile of the standard normal distribution.

We employ the large-sample normal distribution of $\lambda(t)$ and *R(x|t)* and get the two sided approximate $100\alpha\%$ confidence limits for the true values:

$$\lambda_{upper} = \hat{\lambda} + Z_\alpha [Var(\hat{\lambda})]^{1/2} \qquad \lambda_{lower} = \hat{\lambda} - Z_\alpha [Var(\hat{\lambda})]^{1/2} \tag{14}$$

$$R_{upper} = \hat{R} + Z_\alpha [Var(\hat{R})]^{1/2} \qquad R_{lower} = \hat{R} - Z_\alpha [Var(\hat{R})]^{1/2} \tag{15}$$

where Z_α is the $(1-\alpha)$ quartile of the standard normal distribution.

4. An Example of Application

An example is given in this section to illustrate the applicability of the interval estimation. A software was developed and then tested for 28 weeks. The complete failure data were recorded and given in Table 1.

Table 1. Number of failures per month

Month	Failures	Month	Failures	Month	Failures	Month	Failures
1	3	8	32	15	7	22	3
2	3	9	8	16	0	23	4
3	38	10	8	17	2	24	1
4	19	11	11	18	3	25	2
5	12	12	14	19	2	26	1
6	13	13	7	20	5	27	0
7	26	14	7	21	2	28	1

In order to predict the reliability behavior of the software system, we apply the Goel-Okumoto model. To estimate parameter a and b of equation (1), we use the maximum likelihood method discussed in Section 2. The results of the last few weeks are listed in Table 2. The 95% confidence intervals for parameter a and b are listed in Table 3. The 95% confidence intervals for the software failure intensity and reliability predicted in the following months are listed in Table 4.

Table 2. The estimation of parameter a and b using ML methods

Month	Failures	*CMF*	a	b
20	5	220	258.9	0.09472
21	2	222	256.3	0.09578
22	3	225	256.3	0.095597
23	4	229	268.5	0.083328
24	1	230	255.8	0.095575
25	2	232	255.3	0.095751
26	1	233	256.3	0.092254
27	0	233	250.2	0.099146
28	1	234	249.2	0.099855

Table 3. 95% confidence intervals for parameter a and b

Month	a	a_U	a_L	b	b_U	b_L
20	258.9	224.7	293.2	0.09472	0.07397	0.1155
21	256.3	222.6	290.0	0.09578	0.07519	0.1164
22	256.3	222.8	289.8	0.09560	0.07544	0.1157
23	268.5	233.7	303.3	0.08333	0.06548	0.1012
24	255.8	222.7	288.9	0.09558	0.07626	0.1149

25	255.3	222.5	288.2	0.09575	0.07682	0.1147
26	256.3	223.4	289.2	0.09225	0.07407	0.1104
27	250.2	218.1	282.3	0.09915	0.08058	0.1177
28	249.2	217.3	281.1	0.09986	0.08159	0.1181

Table 4. 95% confidence intervals for the failure intensity and software reliability predicted for the following months

month	λ	λ_U	λ_L	R	R_U	R_L
28	1.52	2.11	0.93	0.74	0.83	0.65
29	1.38	1.92	0.83	0.76	0.84	0.68
30	1.24	1.75	0.74	0.78	0.86	0.70
31	1.13	1.59	0.66	0.80	0.87	0.73

5. Discussion

In this paper, we have discussed the interval estimation of the software reliability model parameters. The Goel-Okumoto model is used to illustrate the parameter estimation problems. The confidence limits for the reliability prediction is obtained. The results are useful in practice because we can provide an interval estimates with confidence limits rather than just a point estimate. The approach in our study can be extended to many other models.

References

Azem, A. (1995), *Software Reliability Determination for Conventional and Logic Programming*, Walter de Gruyter, Berlin.

Knafl, G. J. (1992), Solving Maximum Likelihood Equations for 2-Parameter Models Using Grouped Data, *Proceedings of International Symposium on Software Reliability Engineering*, 205-213.

Knafl, G. J. and Morgan, J. (1996), Solving ML Equations for 2-Parameter Poisson-Process Models for Ungrouped Software-Failure Data, *IEEE Transactions on Reliability*, Vol. 45, No. 1, pp. 42-53.

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

Xie, M.(1991), *Software Reliability Modelling*, World Scientific, Singapore.