An Estimation About Software Reliability Based On A New Model

Jinjiang Yao
Department of Mathematics, Linyi Normal University, Linyi Shandong 276005
lysyaojinjiang@yahoo.com.cn

Abstract — Because the data has properties of small sample, and data random deleted, and the method of parameterization estimate at present dealing with this kind of problem is not always successful. In this paper, we propose a new model of the whole reliability of software, and discuss the whole reliability of software. Further, we give a method of parameterization estimate under condition of data random deleted, and prove that the value estimated is a nonlinear equation.

Index Terms——reliability, software, rate of failure of the first defect

I. INTRODUCTION

The reliability of soft is more and more emphasized with the widely applied and complicated. JM model proposed by Jelinski and Moranda[1] is much important in all kinds of models of reliability. The result considered could deviate, if a statistics and analysis is carried out without the information in the data deleted. In addition, we naturally use the method of parameterization estimate, when appropriate model or distribution can impersonate data or suppose the data from some whole of distribution. And in reality incomplete data often appear. At this time, EM algorithm is used to solve the reliability of soft, and a method of parameterization estimate under the condition of random deleting data is given. At last, we prove the quantity of estimate is a nonlinear equation.

II. NEW MODEL OF THE WHOLE RELIABILITY OF SOFTWARE

In paper [2,3], the JM model was improved because the conditions of JM model is powerful. Then in paper[4], a new model of the whole reliability of software was proposed and a reliable scale of point estimate was given because in [2,3] the number of unknown parameters are too much for the number of that observed. But we find in [4] that only the software individual was discussed. So we give a study on the many samples on software:

Suppose the model of reliability of software as follow: There are n samples, and τ1, τ2, ..., τi1, τi2, ..., (i = 1, 2, ..., n) is the moment that defect number of ith sample appears. \( S_{ji} = \tau_{ji} - \tau_{j-1,i} \) is the time interval of the ith sample appears. Suppose \( N_i(t) \) is the defect number of ith sample detected at the moment t. That is \( \forall t \geq 0 \).

\[
N_i(t) = \text{SUP}\{k : \tau_{ki} \leq t, 1 \leq k \} \quad (i = 1, 2, \ldots, n)
\]

Supposed \( \text{SUP}\emptyset = 0 \). We consider that the defect will be deleted at once if it is found during the testing. Let \( \lambda > 0 \) be unknown parameter, \( S_{ji} \) obeys exponential distribution, and the rate of failures is \( \lambda \).

If \( \tau_{1i} \) is given, then \( S_{2i} \) obeys exponential distribution, and rate of failures is

\[
\lambda \exp \left( -\frac{1}{\lambda} \left( \frac{1}{\tau_{1i}} + \frac{1}{\tau_{2i}} \right) \right).
\]

If \( \tau_{1i} \), \( \tau_{2i} \) is given, then \( S_{3i} \) obeys exponential distribution, and rate of failures is

\[
\lambda \exp \left( -\frac{1}{\lambda} \left( \frac{1}{\tau_{1i}} + \frac{1}{\tau_{2i}} + \frac{1}{\tau_{3i}} \right) \right), \quad \text{in general, if } \tau_{1i}, \tau_{2i}, \ldots, \tau_{k-1,i}
\]

is given, then \( S_{ki} \) obeys exponential distribution, and rate of failures is

\[
\lambda \exp \left( -\frac{1}{\lambda} \left( \frac{1}{\tau_{1i}} + \frac{1}{\tau_{2i}} + \ldots + \frac{1}{\tau_{k,i}} \right) \right), \quad (i = 1, 2, \ldots, n)
\]

(2)

We can prove that \( \{ N_i(t), t \geq 0 \} \) is a course of point of self stimulating, and its strength is

\[
\lambda(t, N_i(t), \tau_{1i}, \ldots, \tau_{N_i(t)})) = \lambda \exp \left( -\frac{d_{N_i(t)}}{\lambda} \right).
\]

\( d_{ki} \) satisfies: when \( k > 1 \),

\[
d_{ki} = \frac{1}{\tau_{1i}} + \ldots + \frac{1}{\tau_{N_{ki}}} ; \text{when } k = 1, \ d_{ki} = 0.
\]
We denote \( t_{i_1}, t_{i_2}, \cdots \) as the values observed of \( \tau_{i_1}, \tau_{i_2}, \cdots \), and \( s_{i_1}, s_{i_2}, \cdots \) as the values observed of \( S_{i_1}, S_{i_2}, \cdots \).

The values observed of \( N_i(t) \) is \( n_i(t) \), we can get values as follow:

\[
n_1(t) = t_{i_1}, t_{i_2}, \cdots, t_{n_1};
\]

\[
n_2(t) = s_{i_1}, s_{i_2}, \cdots, s_{n_2};
\]

\[
(i = 1, 2, \cdots, n)
\]

In order to change samples of capacitance \( n \) into sample of capacitance \( 1 \) to use the conclusion in existing paper, we suppose the test with \( n \) samples as a test with \( 1 \) sample,

\[
\sum_{i=1}^{n} \tau_{i_k} = \frac{k}{n} \quad (k = 1, 2, \cdots, n)
\]

The moment that the defects of the software sample appear in return. If \( \tau_{i_k} \) does not be observed, we replace it with \( t \).

\[
S_k = \tau_{i_k} - \tau_{k-1} \quad (k = 1, 2, \cdots, n)
\]

is the integral number of defects of software. And

\[
N(t) = \left[ \sum_{i=1}^{m} N_i(t) \right] / n
\]

(\( m \) is the integral number) is the defect number appear at the moment \( t \). So we provide new model: let \( \lambda > 0 \) be unknown parameter, and \( S \) obey

\[
\lambda \exp \left\{ \left( -\frac{1}{\lambda} \right) \left( \frac{1}{\tau_1} + \frac{1}{\tau_2} + \cdots + \frac{1}{\tau_{k-1}} \right) \right\}
\]

So we obtain the data detected of the new model from (3):

\[
n(t); t_1, t_2, \cdots, t_{N(t)}; S_1, S_2, \cdots, S_{N(t)}, \quad (4)
\]

They are the values of \( N(t); t_1, t_2, \cdots, t_{N(t)}; S_1, S_2, \cdots, S_{N(t)} \), especially.

We can explain this model from view of engineering: supposing the whole rate of failure is \( \lambda \), when software is begun to test. The rate of failure comes to reduce, when the first defect is found and deleted. The measure reduced is \( \hat{\lambda}(1) \) which is the rate of failure of the first defect. At this time, the rate of failure is \( \lambda - \hat{\lambda}(1) \). The rate of failure comes to reduce, when the first defect is found and deleted. The measure reduced is \( \hat{\lambda}(1) \) which is the rate of failure of the first defect. At this time, the rate of failure is \( \lambda - \hat{\lambda}(1) \). The rate of failure comes to reduce again, when the second defect is found and deleted, the measure reduced is \( \hat{\lambda}(2) \). At this time the rate of failure is \( \lambda - \hat{\lambda}(1) - \hat{\lambda}(2) \); In general, the rate of failure is

\[
\lambda - \hat{\lambda}(1) - \hat{\lambda}(2) - \cdots - \hat{\lambda}(k-1)
\]

When the \( k \)-th defect is found and deleted.

But \( \hat{\lambda}(1), \hat{\lambda}(2), \cdots, \hat{\lambda}(k-1) \) are unknown random variables, so we estimate to \( \hat{\lambda}(k) \) \( (k = 1, 2, \cdots, n) \) at first. We consider \( \frac{1}{\tau_k} \) as the measure of estimate. So when the \( k-1 \)-th defect is found and deleted, the rate of failure is about

\[
\hat{\lambda} = \frac{1}{\tau_1} - \cdots - \frac{1}{\tau_{k-1}}
\]

But \( \hat{\lambda} = \frac{1}{\tau_1} - \cdots - \frac{1}{\tau_{k-1}} \) is possibly to be negative.

So we use formula

\[
\text{exponential distribution, rate of failures is } \lambda \text{.}
\]

Suppose \( \tau_1 \) given, \( S_2 \) obey exponential distribution, rate of failures is \( \lambda \exp \left\{ \left( -\frac{1}{\lambda} \right) \left( \frac{1}{\tau_1} + \frac{1}{\tau_2} \right) \right\} \), \( \tau_1, \tau_2 \) given,

\( \lambda \exp \left( \frac{1}{\lambda} + \frac{1}{\tau_2} \right) \), ..., \( \lambda \exp \left( \frac{1}{\lambda} + \frac{1}{\tau_1} \right) \), especially.

In general, \( \tau_1, \tau_2, \cdots, \tau_{k-1} \) is given, \( S_k \) obey exponential distribution, rate of failures is

\[
= \lambda \exp \left( \frac{1}{\lambda} + \frac{1}{\tau_1} + \cdots + \frac{1}{\tau_{k-1}} \right)
\]

We consider the value of estimate (3) by replacing (4) with (5). Thus at this time we can obtain the new model when we suppose the time interval obey exponential distribution. This model is adaptable, when the problem we care for is the compendia that the software submitted after the certain time.

Supposed \( Y(T) = T_{N(t)+1} - T \), it means the moment that the next defect appear, and the reliability to estimate is

\[
R(\lambda, x, N(t), \tau_1, \tau_2, \cdots, \tau_{N(t)})
\]

\[
= P(Y(t) > x | N(t), \tau_1, \tau_2, \cdots, \tau_{N(t)})
\]

\[
= \exp \left( -x \lambda \exp \left( \frac{d_{N(t)}}{\lambda} \right) \right)
\]

Supposed \( d_k = \frac{1}{\tau_1} + \cdots + \frac{1}{\tau_{k-1}} \), it is enough for us to estimate only the value \( \lambda \), because

\( R(\lambda, x, N(t), \tau_1, \tau_2, \cdots, \tau_{N(t)}) \) is a monotone decreasing function of \( \lambda \).
III. AN ESTIMATION ABOUT SOFTWARE RELIABILITY BASED ON A NEW MODEL

Supposed the normal working time of the software is $y$, when $k$ defects are found, then

$$f(y) = \lambda \exp\left[-\frac{d_y}{\lambda}\right] \exp\left[-\lambda \exp\left(-\frac{d_y}{\lambda}\right)\right]$$

Suppose that we have a test on the real data $Y = (y_1, y_2, \cdots, y_k, y_{k+1}, \cdots, y_n)$. But $y$ is not observed because of some reason. And only the function $Z = (z_1, \cdots, z_k, z_{k+1}, \cdots, z_n)$ of $y$ is found, $z_{k+1}^+, \cdots, z_n^+$ denote that there are some deletions of these data. And there are relationship between $y$ and $z$

$$y_j = z_j, j = 1, \cdots, k$$
$$y_j \geq z_j, j = k + 1, \cdots, n$$

We need to estimate parameter $\lambda$ by the EM algorithm, when we can not get the complete data $Z$.

Theorem 1: If the random data $Z = (z_1, \cdots, z_k, z_{k+1}, \cdots, z_n)$ obey the exponential distribution, then parameter estimation value $\hat{\lambda}$ obtained from EM algorithm satisfy

$$\lambda = \sum_{j=1}^{n} d_y \left[ \exp\left(-\frac{d_y}{\lambda}\right) - 1 \right] - n$$

(10)

Proof: From EM algorithm[6], Supposed $\lambda^i$ is the estimation value of the $i + 1$ th iteration, then the $i + 1$ th steps

E step:

$$Q(\theta | \theta^i) = n \ln \lambda - \sum_{j=1}^{n} \exp\left(-\frac{d_y}{\lambda}\right) E(y_j | z, \lambda^i) - \sum_{j=1}^{n} \lambda \exp\left(-\frac{d_y}{\lambda}\right) E(y_j | z, \lambda^i)$$

(11)

M step:

$$\frac{\partial Q}{\partial \lambda} = \frac{n}{\lambda} + \sum_{j=1}^{n} \exp\left(-\frac{d_y}{\lambda}\right) E(y_j | z, \lambda^i) - \sum_{j=1}^{n} \lambda \exp\left(-\frac{d_y}{\lambda}\right) d_y E(y_j | z, \lambda^i) = 0$$

(12)

In fact, the solution of equation (12) is the parameter estimate value of $i + 1$ th iteration. Furthermore

$$\sum_{j=1}^{n} \exp\left(-\frac{d_y}{\lambda}\right) E(y_j | z, \lambda^i) = \sum_{j=1}^{n} \exp\left(-\frac{d_y}{\lambda}\right) z_j$$

(13)

By the same way

$$\sum_{j=1}^{n} \exp\left(-\frac{d_y}{\lambda}\right) E(y_j | z, \lambda^i)$$

$$= \sum_{j=1}^{n} \exp\left(-\frac{d_y}{\lambda}\right) z_j$$

(14)

We take the formula above into (12), then we can have

$$\frac{n}{\lambda^i} + \sum_{j=1}^{n} \frac{d_y}{\lambda^i} = \sum_{j=1}^{n} \exp\left(-\frac{d_y}{\lambda}\right) z_j$$

(15)

(16)
\[ z_j \lambda \exp \left( -\frac{d_j}{\lambda} \right) + 1 - \frac{\lambda^2}{\lambda^2} \exp \left[ -\lambda \exp \left( -\frac{d_j}{\lambda} \right) z_j + \frac{2d_j}{\lambda} \right] \]

Then (14) can be simplified:

\[
\sum_{j=1}^{n} d_j + \sum_{j=1}^{n} [z_j \lambda \exp \left( -\frac{d_j}{\lambda} \right) + 1] [\exp \left( \frac{d_j}{\lambda} \right) + d_j] = \sum_{j=1}^{n} d_j \lambda [\exp \left( -\frac{d_j}{\lambda} \right) - 1] - n
\]

Thus we can obtain the parameter estimation value of the model of software by solve nonlinear equation (10) under this condition.

REFERENCES