UNIT-IV

Software NHPP Models

Although some basic and advanced Markov models are presented in the previous sections, some NHPP models are mentioned here due to their significant impact on the software reliability analysis. Such a model simply models the failure occurrence rate as a function of time (see e.g., Section 2.4). Hopefully this occurrence rate is decreasing when faults are removed as an effect of debussing. Note that after the release, the failure occurrence rate should be a constant unless the debugging is continued (Yang & Xie, 2000).

4.5.1. The Goel-Okumoto (GO) model

In 1979, Goel and Okumoto presented a simple model for the description of software failure process by assuming that the cumulative failure process is NHPP with a simple mean value function. Although NHPP models have been studied before, see e.g. Schneidewind (1975), the GO-model is the basic NHPP model that later has had a strong influence on the software reliability modeling history.

Model description

The general assumptions of the GO-model are

1) The cumulative number of faults detected at time t follows a Poisson distribution.
2) All faults are independent and have the same chance of being detected.
3) All detected faults are removed immediately and no new faults are introduced.

Specifically, the GO-model assumes that the failure process is modeled by an NHPP model with mean value function $m(t)$ given by

$$m(t) = a[1 - \exp(-bt)], \quad a > 0, b > 0 \qquad (4.46)$$

The failure intensity function can be derived by

$$\lambda(t) = \frac{d}{dt}m(t) = ab\exp(-bt) \tag{4.47}$$

where *a* and *b* are positive constant. Note that $m(\infty) = a$. The physical meaning of parameter *a* can be explained as the expected number of faults which are eventually detected. The quantity *b* can be interpreted as the failure occurrence rate per fault.

The expected number of remaining faults at time *t* can be calculated as

$$E[N(\infty) - N(t)] = m(\infty) - m(t) = a\exp(-bt)$$

The GO-model has a simple but interesting interpretation based on a model for fault detection process. Suppose that the expected number of faults detected in a time interval $[t, t + \Delta t)$ is proportional to the number of remaining faults, we have that

$$m(t + \Delta t) = b[a - m(t)]\Delta t$$

where *b* is a constant of proportionality.

The above difference equation can be transformed into a differential equation. Divide both sides by Δt and take limits by letting Δt tend to zero, we get the following equation,

$$m'(t) = a \cdot b - b \cdot m(t)$$

It can be shown that the solution of this differential equation, together with the initial condition $m(0) = 0$, lead to the mean value function of the GO-model.

Note that both the GO-model and JM-model give the exponentially decreasing number of remaining faults. It can be shown that these two models cannot be distinguished using only one realization from each model. However, the models are different because the JM-model assumes a discrete change of the failure intensity at the time of the removal of a fault while the GO-model assumes a continuous failure intensity function over the whole time domain.

Parameter estimation

Denoted by n_i the number of faults detected in time interval $[t_{i-1}, t_i)$, where $0 = t_0 < t_1 < \cdots < t_k$ and t_i are the running times since the software testing begins. The estimation of model parameters *a* and *b* can be carried out by maximizing the likelihood function, see e.g. Goel & Okumoto (1979). The likelihood function can be reduced to

$$\sum_{i=1}^{k} \frac{n_i[t_i \exp(-bt_i) - t_{i-1}\exp(-bt_{i-1})]}{\exp(-bt_{i-1}) - \exp(-bt_i)} = \frac{t_k \exp(-bt_k) \cdot \sum_{i=1}^{k} n_i}{1 - \exp(-bt_k)} \qquad (4.48)$$

Solving this equation to calculate the estimate of *b*, and then *a* can be estimated as

$$a = \frac{\sum_{i=1}^{k} n_i}{1 - \exp(-bt_k)} \qquad (4.49)$$

Usually, the above two equations has to be solved numerically. It can also be shown that the estimates are asymptotically normal and a confidence region can easily be established. A numerical example is illustrated below.

Example 4.5. Suppose a software product is being tested by a group. Each time when detecting the failure, it is removed and the time for repair is not computed in the test time. The 30 test data of time to failures are recorded in Table 4.4.

Solving the likelihood equations, we get *b* = 0.0008 and *a* = 57. The failure intensity function and the mean value function for this GO model are

$$\lambda(t) = 0.0456 \exp(-0.0008t)$$

and

$$m(t) = 57[1 - \exp(-0.0008t)]$$

Table 4.4. A set of time to failure data.

Failure number	Time to failures	Failure number	Time to failures	Failure number	Time to failures
1	1.33	11	288.43	21	547.21
2	3.43	12	288.84	22	554.65
3	24.87	13	303.02	23	629.93
4	58.15	14	330.30	24	741.44
5	85.78	15	375.16	25	773.25
6	145.84	16	414.85	26	789.56
7	203.84	17	417.96	27	815.74
8	205.82	18	434.56	28	874.62
9	219.97	19	517.28	29	888.81
10	244.09	20	543.72	30	924.94

■

4.5.2. S-shaped NHPP models

The mean value function of the GO-model is exponential-shaped. Based on the experience, it is observed that the curve of the cumulative number of faults is often S-shaped as shown by Fig. 4.8, see e.g. Yamada *et al.* (1984).

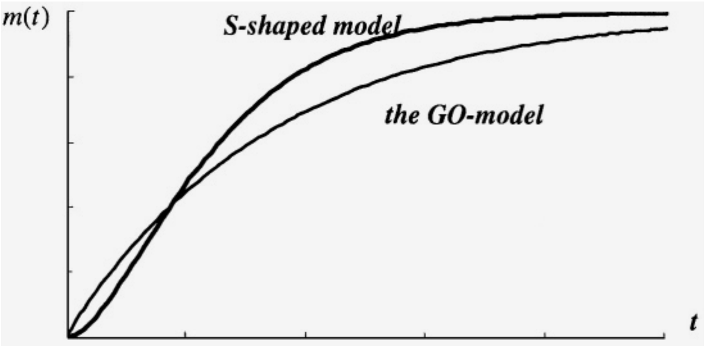

Fig. 4.8. The S-shaped mean value function.

This can be explained by the fact that at the beginning of the testing, some faults might be "covered" by other faults. Removing a detected fault at the beginning does not reduce the failure intensity very much since the same test data will still lead to a failure caused by other faults. Another reason of the S-shaped behavior is the learning effect as indicated in Yamada *et al*. (1984).

Delayed S-shaped NHPP model

The mean value function of the *delayed S-shaped* NHPP model is

$$m(t) = a[1-(1+bt)\exp(-bt)]\,;\quad b>0, \qquad (4.50)$$

This is a two-parameter S-shaped curve with parameter a denoting the number of faults to be detected and b corresponding to a fault detection rate. The corresponding failure intensity function of this delayed S-shaped NHPP model is

$$\lambda(t) = \frac{dm(t)}{dt} = ab(1+bt)\exp(-bt) - ab\exp(-bt) = ab^2 t\exp(-bt)$$

The expected number of remaining faults at time t is then

$$m(\infty) - m(t) = a(1+bt)\exp(-bt)$$

Inflected S-shaped NHPP model

The mean value function of the *inflected S-shaped* NHPP model is

$$m(t) = \frac{a[1-\exp(-bt)]}{1+c\exp(-bt)}\,;\quad b>0, c>0$$

In the above a is again the total number of faults to be detected while b and c are called the fault detection rate and the inflection factor, respectively. The intensity function of this inflected S-shaped NHPP model can easily be derived as

$$\lambda(t)=\frac{dm(t)}{dt}=\frac{ab(1+c)\cdot\exp(-bt)}{[1+c\exp(-bt)]^2}$$

Given a set of failure data, for both delayed and inflated S-shaped NHPP models, numerical methods have to be used to solve the likelihood equation so that estimates of the parameters can be obtained.

4.5.3. Some other NHPP models

Besides the S-shaped models, there are many other NHPP models that extend the GO-model for different specific conditions.

Duane model

The Duane model assumes that the mean value function satisfies

$$m(t)=\left(\frac{t}{\alpha}\right)^{\beta},\quad \alpha>0,\ \beta>0 \tag{4.51}$$

In the above, α and β are parameters which can be estimated by using collected failure data. The mean value functions with $\alpha=100$ and different $\beta=\{0.5,1,2\}$ are depicted by the Fig. 4.9.

It can be noted that when $\beta=1$, the Duane NHPP model is reduced to a Poisson process whose mean value function is a straight line. In such a case, there is no reliability growth. In fact, the Duane model can be used to model both reliability growth $(\beta<1)$ and reliability deterioration $(\beta>1)$ which is common in hardware systems.

The failure intensity function, $\lambda(t)$, is

$$\lambda(t)=\frac{d}{dt}m(t)=\frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1},\quad \alpha>0,\ \beta>0$$

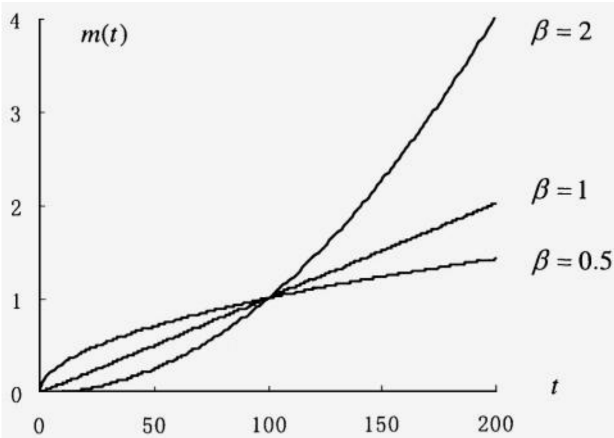

Fig. 4.9. Mean value functions of Duane NHPP models

One of the most important advantages of the Duane model is that if we plot the cumulative number of failure versus the cumulative testing time on a log-log-scale, the plotted points tends to be close to a straight line if the model is valid. This can be seen from the fact that the relation between $m(t)$ and t can be rewritten as

$$\ln m(t) = -\beta \ln \alpha + \beta \ln t = a + b \ln t$$

where $a = -\beta \ln \alpha$ and $b = \beta$. Hence, $\ln m(t)$ is a linear function of $\ln t$ and due to this linear relation, the parameters α and β may be estimated graphically and the model validity can easily be verified. In fact, this is called first-model-validation-then-parameter-estimation approach (Xie & Zhao, 1993).

The Duane model gives an infinite failure intensity at time zero. Littlewood (1984) proposed a *modified Duane model* with the mean value function

$$m(t) = k\left[1 - \left(\frac{\alpha}{\alpha + t}\right)^{\beta}\right], \quad \alpha > 0, \ \beta > 0, \ k > 0$$

The parameter k can be interpreted as the number of faults eventually to be detected.

Log-power model

Xie & Zhao (1993) presented a log-power model. The mean value function of this model can be written as

$$m(t) = a\ln^{b}(1+t)\,;\ \ a,b>0\,,\ \ t\ge 0 \qquad (4.52)$$

This model has shown to be useful for software reliability analysis as it is a pure reliability growth model. It is also easy to use due to its graphical interpretation. The plot of the cumulative number of failures at time t against t+1 will tend to be a straight line on a log-double-log scale if the failures follow the log-power model. This can be seen from the following relationship

$$\ln m(t) = \ln a + b\ln\ln(1+t)$$

The slope of the fitted line gives an estimation of b and its intercept on the vertical axis gives an estimation of ln$a$.

The failure intensity function of the log-power model can be obtained as

$$\lambda(t) = \frac{ab\ln^{b-1}(1+t)}{1+t}\,,\ \ t\ge 0 \qquad (4.53)$$

The failure intensity function is interesting from a practical point of view. The log-power model is able to analyze both the case of strictly decreasing failure intensity and the case of increasing-then-decreasing failure intensity function. For example, if $b\le 1$, then $\lambda(t)$ of the above equation is a monotonic decreasing function of t; Otherwise given $b>1$, $\lambda(t)$ is increasing if $0\le t<\exp(b-1)$ and decreasing if $t\ge \exp(b-1)$.

The estimation of the parameters a and b is also simple. Suppose total n failures are detected during the a testing period $(0,T]$ and the times to failures are ordered by $0 < t_1 < t_2 < \cdots < t_n \le T$. The maximum likelihood estimation of *a* and *b* is then given by:

$$\hat{b} = \frac{n}{n \ln \ln(1+T) - \sum_{i=1}^{n} \ln \ln(1+t_i)}$$

and

$$\hat{a} = \frac{n}{\ln^{\hat{b}}(1+T)}$$

They can be simply calculated without numerical procedures.

Musa-Okumoto model

Musa and Okumoto (1984) is another model for infinite failures. This NHPP model is also called the logarithmic Poisson model. The mean value function is

$$m(t) = a \ln(1+bt),\ t > 0 \tag{4.54}$$

The failure intensity function is derived as

$$\lambda(t) = \frac{ab}{1+bt}$$

Given a set of failure time data $\{t_i, i = 1,2,\ldots,n\}$, the maximum likelihood estimates of the parameters are the solutions of the following equations:

$$\begin{cases} \hat{a} = \dfrac{n}{\ln(1+\hat{b}t_n)} \\ \dfrac{1}{\hat{b}} \sum_{i=1}^{n} \dfrac{1}{1+\hat{b}t_i} - \dfrac{nt_n}{(1+\hat{b}t_n)\ln(1+\hat{b}t_n)} = 0 \end{cases} \tag{4.55}$$

These equations have to be solved numerically.