NHPP-Based Sofotware Reliability Model with Mixed Gamma Distribution

Hiroyuki Okamura, Takumi Hirata, and Tadashi Dohi□

Department of Information Engineering, Graduate School of Engineering,
Hiroshima University, 1–4–1 Kagamiyama, Higashi-Hiroshima 739-8527, Japan
http://www.rel.hiroshima-u.ac.jp/

Abstract. This paper proposes a software reliability model (SRM) based on a mixed gamma distribution, so-called the mixed gamma SRM. In addition, we develop the parameter estimation method for the mixed gamma SRM. Concretely, the estimation method is based on the Bayesian estimation and the parameter estimation algorithm is described by MCMC (Markov chain Monte Carlo) method with grouped data.

1 Introduction

Software reliability is a significant attribute of software quality. The quantitative software reliability is defined as the probability that no failure occurs during a certain time period. Thus probabilistic models are needed to evaluate quantitative software reliability from field data (failure data). In fact, a vast amount of software reliability models (SRMs) have been proposed and developed from various points of view during the last four decades. Specifically, non-homogeneous Poisson process (NHPP) based SRMs have played a central role to estimate the number of remaining faults as well as the quantitative software reliability [Musa et al.(1987)Musa, Iannino, and Okumoto,Lyu(1996)].

In general, there are two categories of SRMs: parametric and non-parametric SRMs. When evaluating the software reliability using parametric NHPP-based SRMs, we first decide a set of candidates used in the software reliability estimation. The statistical parameter estimation is executed for all the candidates to determine their model parameters fitting to observed software failure data. After estimating model parameters, we choose the best model in the sense of information criteria such as AIC (Akaike information criterion). However, it is not easy to decide a good set of candidates that include the best model describing the software failure occurrence process.

In the non-parametric approach, we do not need to decide the candidates before estimating model parameters. For example, kernel density estimation, which is a typical non-parametric approach, defines a kernel function for each data point. Without any density function, it gives an estimated density function

□ This research was supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), Grant No. 21510167 (2009-2011), Grant No. 23500047 (2011–2013) and Grant No. 23510171 (2011–2013).

based on the set of kernel functions. The simplest example of non-parametric estimation is the empirical distribution. Also, neural network models are also categorized to one of the non-parametric methods. Several papers discussed the applicability of non-parametric approaches to the software reliability evaluation. Generally, the non-parametric estimation gives highly accurate estimators in the case where many samples are observed. However, we rarely observe many failure time or count data in the software reliability evaluation. In addition, the drawback of non-parametric approach is not to estimate the density for truncated area. In general, software failure data behave similar to truncated data, and thus non-parametric approach does not work well.

This paper proposes an intermediate SRM between parametric and non-parametric models, which can fit to any type of fault detection data. Concretely, we present an SRM based on mixed gamma distribution, so-called mixed gamma SRM. For this model, we consider the Bayesian estimation. In general, the Bayesian estimation is commonly used in the parameter estimation of semi-parametric models. Roughly speaking, the maximum likelihood (ML) estimation, which is frequently utilized in the software reliability evaluation, causes overfitting problem due to the fact that the proposed model consists of many model parameters to be estimated. Compared with ML estimation, it is well known that the Bayesian method avoids the overfitting problem.

2 Mixed Gamma SRM

The mixed gamma SRM is defined as the NHPP model where failure occurrence time follows a mixed gamma distribution. The c.d.f. and the p.d.f. of mixed gamma distribution are given by

$$\Phi(\tau;\omega,\theta) = \int_0^{\tau} \phi(\upsilon;\omega,\theta)\,d\upsilon, \tag{1}$$

$$\phi(\tau;\omega,\theta,\eta) = \sum_{k=1}^{m} \omega_k G(\tau;\theta_k,\eta_k), \tag{2}$$

where

$$\omega = \{\omega_1,\ldots,\omega_m\},\ \sum_{k=1}^{m} \omega_k = 1,\ \omega_k \geq 0, \tag{3}$$

$$\theta = \{\theta_1,\ldots,\theta_m\},\ \eta = \{\eta_1,\ldots,\eta_m\}. \tag{4}$$

According to the order statistics model [Langberg and Singpurwalla(1985)], we get the following probability mass function of the cumulative number of software failures $N(\tau)$ before time τ:

$$\Pi(N(\tau) = \kappa) = \frac{(\omega\Phi(\tau;\theta,\eta,\omega))^{\kappa}}{\kappa!} \exp(-\omega\Phi(\tau;\theta,\eta,\omega)). \tag{5}$$

Obviously, the above probability mass function corresponds to the NHPP probability mass function with the mean value function $\omega\Phi(\tau;\theta,\eta,\omega)$, where ω in-dicates the average number of total failures during the software life cycle.

3 Parameter Estimation

As mentioned before, ML estimation for semi-parametric models often causes the overfitting problem. Therefore, this paper considers Bayesian estimation for mixed gamma SRM.

Suppose that software failure occurrences are observed as grouped data. That is, the grouped data collect the number of failure occurrences as a bin. Let τ_j and ξ_j be the cumulative testing time indicating breaks of bins and the number of failures observed in the $\imath$-th bin, where we assume $\tau_0 = 0$.

Consider the Bayesian estimation with the grouped data. Bayesian estimation is the well-established framework for parameter estimation based on prior information. The key idea behind the Bayesian estimation is to regard model parameters as random variables.

Let $\pi(\boldsymbol{\theta})$ and Δ denote the prior information about a parameter set $\boldsymbol{\theta}$ and observed data set, respectively. From Bayes theorem, the posterior information, i.e., the updated information after obtaining Δ is given by

$$\pi(\boldsymbol{\theta} \mid \Delta) = \pi(\Delta \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}) \Big/ \int \pi(\Delta \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})\delta\boldsymbol{\theta}, \tag{6}$$

where $\pi(\Delta \mid \boldsymbol{\theta})$ is the likelihood of Δ on the fixed parameter set $\boldsymbol{\theta}$. Taking account of the normalizing condition of posterior distributions, Eq. (6) can be alternatively expressed without the normalizing constant $\int \pi(\Delta \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})\delta\boldsymbol{\theta}$:

$$\pi(\boldsymbol{\theta} \mid \Delta) \text{ oc } \pi(\Delta \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{7}$$

The computation of normalizing constant causes analytical or numerical integration over the domain of parameter set $\boldsymbol{\theta}$. Only for some specified cases, we can obtain the closed forms of normalizing constants; for example, when the conjugate prior distributions are applied, we have normalizing constants implicitly. In most situations, however, any numerical technique has to be utilized for evaluating posterior distributions. Thus, the concrete estimation algorithm is implemented by the Markov chain Monte Carlo (MCMC).

The MCMC (Markov chain Monte Carlo) is a versatile method to evaluate posterior distributions in Bayesian estimation. The MCMC can be regarded as sampling-based approximation of posterior distributions. Except for the use of conjugate prior distribution, it is difficult to derive the closed forms of posterior distributions. Therefore, we have to apply any specific sampling method to obtain samples from target posterior distributions. The Gibbs sampling and the Metropolis-Hasting method are representative sampling methods in the MCMC.

In the Gibbs sampling, one generates the target joint posterior distribution based on conditional posterior distributions. Let $\pi(\square_1,\ldots, \square_m \mid \Delta)$ be the target joint posterior distribution of parameters$\square_1,\ldots, \square_m$. When one can generate samples from the conditional posterior distribution;

$$\pi(\square_j \mid \square_1,\ldots,\square_{j-1},\square_{j+1},\ldots,\square_m,\Delta), \tag{8}$$

the Gibbs sampler is given by the following scheme:

$$\tilde{\omega}_1 \sim \pi(\omega_1 \mid \tilde{\omega}_2,\ldots,\tilde{\omega}_m,\Delta), \tag{9}$$

$$\tilde{\omega}_2 \sim \pi(\omega_2 \mid \ldots,\tilde{\omega}_m,\Delta), \tag{10}$$

...

$$\tilde{\omega}_i \sim \pi(\omega_i \mid \tilde{\omega}_1,\ldots,\tilde{\omega}_{i-1},\tilde{\omega}_{i+1},\ldots,\tilde{\omega}_m,\Delta), \tag{11}$$

...

$$\tilde{\omega}_m \sim \pi(\omega_i \mid .,\tilde{\omega}_{m-1},\Delta), \tag{12}$$

where $\tilde{\omega} \sim \pi(\cdot)$ indicates that the sample $\tilde{\omega}$ is generated from the probability distribution $\pi(\cdot)$. The above sampling can be regarded as Markov chain with state space $(\omega_1,\ldots,\omega_m)$. In the Gibbs sampling, the stationary distribution of this Markov chain is exactly equal to the joint posterior distribution. Therefore, we obtain the samples from the joint posterior distribution by repeating the above sampling scheme.

4 Conclusions

This paper has proposed the SRM based on the mixed gamma distribution and has discussed the parameter estimation based on Bayes estimation. By applying this model, we do not need to decide the set of candidates from a vast amount of parametric SRMs before estimating their model parameters. This is a great merit from the practical point of view.

References

[Langberg and Singpurwalla(1985)] Langberg, N., Singpurwalla, N.D.: Unification of some software reliability models. SIAM Journal on Scientific Computing 6(3), 781–790 (1985)

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

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