Software Reliability Analysis Based on Hierarchical Dynamic Models and Bayesian Estimations using Machine Learning

Toru Kaise  
Graduate School of Business, University of Hyogo  
8-2-1 Gakuen-Nishimachi, Nishiku, Kobe 651-2197, Japan  
E-mail: kaise@mba.u-hyogo.ac.jp

Abstract

A Bayesian dynamic method for analysis of software debugging process data is handled. It is addressed to predict states of software reliability. In the Bayesian analysis, hierarchical prior models are structured with the Boltzmann machine, and empirical and expert knowledge priors are supposed. These priors play roles recognized as representations for complex situations. The empirical prior based on observed data is used for the representation of uncertainty corrections. The prior of success probability for the tests is assumed based on expert knowledge of engineers. The reliability is estimated based on the posterior mean of the failure states. The Bayesian inferences are derived based on the computational simulation methods, and the information criterion EIC is used to choose appropriate models.

1 Introduction

Debugging processes are handled in software developments. The processes consist of testing and correcting for the activities. However tests and corrections of the debugging processes have uncertainty performances. It is possible to assume that the tests are not perfect operations to find errors involved in the software, and some of the corrections also might be not complete treatments for the defections. The reliability of the software means dependency of the amount number of remaining errors. In software developments, predictions of the reliability are derived, and then releases are decided in the case of high reliability situations. These mean that remaining errors are critical estimates. Therefore, methodologies for the evaluation of the software reliability are significant. In particular, the models are necessary to be considered as the situation of the uncertain debugging performances.

The software reliability models and estimation methods have been developed, and the models are classified into two types which are dynamic and static [1], [2]. As the dynamic models, nonhomogeneous Poisson processes (NHPP) and Jelinski-Moranda (JM) methods play the important roles on inferences of the software reliability [3], [4]. The NHPP method is applied to the debugging numbers as discrete variables, and it is assumed that the observed numbers of debugging are decreasing according to repetitions of testing. Therefore the NHPP method represents the reliability growth based on the observations of debugging numbers. The JM method deals with interval time between bugs found at tests, and it is assumed that the hazard function as the continuous variable is decreasing according to advanced defect processes. These methodologies are exceeded for complex phenomena of the debugging processes based on Bayesian methodologies. In recent years, computational estimation methods using simulation techniques are applied to the Bayesian dynamic models [5], [6]. As the static models, the hypergeometric distribution and others are used based on the maximum likelihood estimators. These methods make the possible to estimate amount error numbers after the debugging operations [1].

In this paper, a dynamical modeling method for interval time between debugs is handled [8]. The Weibull distribution is used for the hierarchical dynamical models, and the gamma distributions are used for the scale and shape parameters with the Boltzmann machine. In particular, prior organized knowledge is proposed to install the hype parameters. Bayesian computational inferences are used for the Bayesian dynamical models based on Markov chain Monte Carlo methods[7]. In particular, approximate marginal computations are applied to the Bayesian computational methods with maximizations of marginal likelihoods [9]. The information criterion EIC is applied to the Bayesian computational methodologies, and the model and priors fitted to a set of data are chosen [10]. It is provided that the reliability prediction method based on the Bayesian computation means a machine learning, and moreover the methodology is effective for software developments.

2 Dynamical Bayesian Model

Let $i$ be a debugging treatment, and we suppose that $i = 1, 2, 3, \cdots, n$. It is denoted that $m_i$ is the num-
ber of ith remained errors which is unobserved, $t_i$ is the interval time between $i$ and $i+1$, and $k_i$ is the number of found errors at the ith debugging performance. The data of the observations are denoted by $t_i^o = \{t_1, t_2, \cdots, t_n\}^T$, and $k_i = \{k_1, k_2, \cdots, k_n\}^T$. The reliability of the software is depended on the $m_i, i = 1, 2, 3, \cdots, n$. However the debugging performance at stage $i$ have uncertainty of the increasing or decreasing reliability, and then it is not true that $m_{i+1}$ equals $m_i - k_i$. Therefore the software release is decided by the prediction of the reliability based on $t_i^o$ and $k_i^o$ using the Bayesian modeling and the machine learning inferences. The description of the debugging processes is given by Fig. 1.

$$
\begin{array}{c|c|c|c}
  m_{i-1} & m_i & m_{i+1} \\
  k_{i-1} & k_i & k_{i+1} \\
  \lambda_{i-1} & \lambda_i & \lambda_{i+1} \\
\end{array}
$$

where $\lambda_i > 0$ means the scale of stochastic property for $i$, and it has the probability density function $g(\lambda_i | a_i, b_i)$ of the gamma distribution, which is given by

$$
g(\lambda_i | a_i, b_i) = \frac{b_i^a}{\Gamma(a)} \lambda_i^{a-1} \exp\left[-b_i \lambda_i\right], \quad (2)
$$

where $a > 0$ is the shape parameter, and $b_i > 0$ means the scale of stochastic behavior of $\lambda_i$. In addition, we assume that $b_i = \Phi(z_i; p, W) = \exp[- \sum_{j=1}^r p_j z_{ij} - \sum_{k=1}^r \sum_{j=1}^r w_{jk} z_{ik}]$, where conditions $z_i = [z_{i1}, z_{i2}, \cdots, z_{ir}]$ and parameters $p = [p_1, p_2, \cdots, p_r]$, $W = [w_{jk}], j = 1, 2, \cdots, r, k = 1, 2, \cdots, r$. It is possible that the the function based on the Boltzmann machine $\Phi(z_i; p, W)$ is more comlexity due to the additional deep structures. The forms (1) and (2) represent the observation equation of $t_i$ and the state equation of $\lambda_i$, respectively. The state space modeling is a dynamical system based on the hierarchical Bayesian method. The prior distribution of $\beta$ in the form (1) is given by

$$
k(\beta) = \frac{h^c}{\Gamma(c)} \beta^{c-1} \exp[-h \beta], \quad (3)
$$

where $h > 0$ and $c > 0$ mean the hyper parameters, and $k(\beta)$ is the probability density function of the gamma distribution. The prior distribution of $a$ in the form (2) is given by

$$
\pi(a) = \frac{1}{\gamma}, \quad (4)
$$

where the form means the probability density function of the uniform distribution with the hyper parameter $\gamma > 0$. It is proposed that the hyper parameters $h, c$ and $\gamma$ are provided from experience of common knowledge in past developmental activities, and the Bayesian handling of knowledge is effective for the complex reliability analysis.

We suppose that $k_i, i = 1, 2, \cdots, n$ are fixed as 1 values respectively. In actual, it is desired that software debugging processes are evaluated by the amount of testing or correcting observations for each stage at $i$. We might consider the definition of $t_i$ in detail, and it might be useful that the simple situation is handled at the present moment.

3 Posterior Distributions

According to the Bayesian methodology, the conditional posterior distributions are derived based on forms (1), (2), (3), (4).

1) the case of unified prior distribution for $\beta$

Using the forms (1) and (2), the posterior distribution $p(\lambda_i | t_i, \beta)$ is derived as follows

$$
p(\lambda_i | t_i, \beta) = \frac{f(t_i | \beta, \lambda_i) g(\lambda_i | a_i, b_i)}{\int_0^{\infty} f(t_i | \beta, \lambda_i) g(\lambda_i | a_i, b_i) d\lambda_i} = \frac{\lambda_i^a e^{-(\lambda_i t_i^o + b_i)}}{\Gamma(a + 1)} \exp[-\lambda_i (t_i^o + b_i)], \quad (5)
$$

According to the forms (1) and (3), the posterior distribution $p(\beta | t_i, \lambda_i)$ is given by

$$
p(\beta | t_i, \lambda_i) = \frac{\int_0^{\infty} f(t_i | \beta, \lambda_i) k(\beta)}{\int_0^{\infty} f(t_i | \beta, \lambda_i) k(\beta) d\beta}, \quad (6)
$$

where the numerator of the form (6) is given by

$$
f(t_i | \beta, \lambda_i)k(\beta) = \frac{\lambda_i h^c}{\Gamma(c)} \beta^{c-1} e^{-(\lambda_i t_i^o + h \beta)}. \quad (7)
$$

Therefore, the denominator is derived as follows

$$
\int_0^{\infty} f(t_i | \beta, \lambda_i)k(\beta) d\beta = \frac{\lambda_i h^c}{\Gamma(c)} I_\beta(c, t_i). \quad (8)
$$

Substituting the forms (7) and (8) for the form (6), $p(\beta | t_i, \lambda_i)$ is given by

$$
p(\beta | t_i, \lambda_i) = \frac{t_i^o x^c \exp(-\lambda_i t_i^o - h \beta)}{I_\beta(c, t_i)}. \quad (9)
$$
here note that
\[
I_{\beta}(c, t_i) = \int_0^{\infty} \beta^c t_i^{a-1} \exp \left[ -\lambda_i t_i^c - h \beta \right] d\beta.
\]

The posterior means of the posterior distributions (6) and (9) are given by
\[
E\{\lambda_i|t_i, \beta\} = \int_0^{\infty} \lambda_i p(\lambda_i|t_i, \beta) d\lambda_i = \frac{a + 1}{t_i^c + b_i}
\]
and
\[
E\{\beta|t_i, \lambda_i\} = \int_0^{\infty} \beta p(\beta|t_i, \lambda_i) d\beta = \frac{I_{\beta}(c + 1, t_i)}{I_{\beta}(c, t_i)},
\]
respectively.

2) the case of mutualized prior distribution for \( \beta \) and \( \lambda_i \)

The mutual prior distribution of \( \lambda_i \) and \( \beta \) is denoted by \( \pi(\lambda_i, \beta) \), and we assume that the form \( \pi(\lambda_i, \beta) \) is given by forms (2) and (3), that is,
\[
\pi(\lambda_i, \beta) = g(\lambda_i|a, b_i)k(\beta) = \frac{b_i^a \lambda_i^{c-1} \beta^{-1} \exp [-b_i \lambda_i - h \beta]}{\Gamma(a) \Gamma(c)}.
\]

According to the forms (1) and (10), the posterior distribution \( p(\lambda_i, \beta|t_i) \) is derived as follows
\[
p(\lambda_i, \beta|t_i) = \frac{f(t_i|\lambda_i, \beta)\pi(\lambda_i, \beta)}{\int_0^{\infty} f(t_i|\lambda_i, \beta)\pi(\lambda_i, \beta) d\lambda_i d\beta} = \frac{\lambda_i^c t_i^{a-1} \exp \left[ -\lambda_i (t_i^{c+1} + b_i^c) - h \beta \right]}{\Gamma(a + 1) J_{\beta}(c, a + 1, t_i)},
\]
where \( J_{\beta}(c, a + 1, t_i) \) means that
\[
J_{\beta}(c, a + 1, t_i) = \int_0^{\infty} \beta^c t_i^{a-1} \exp [-h \beta] d\beta.
\]

Using the form (11), the posterior means for \( \lambda_i \) and \( \beta \) are derived as follows
\[
E\{\lambda_i|t_i\} = \int_0^{\infty} \int_0^{\infty} \lambda_i p(\lambda_i, \beta|t_i) d\beta d\lambda_i = \frac{(a + 1)J_{\beta}(c, a + 2, t_i)}{J_{\beta}(c, a + 1, t_i)}
\]
and
\[
E\{\beta|t_i\} = \int_0^{\infty} \int_0^{\infty} \beta p(\lambda_i, \beta|t_i) d\lambda_i d\beta = \frac{J_{\beta}(c + 1, a + 1, t_i)}{J_{\beta}(c, a + 1, t_i)}.
\]

3) Marginal likelihood

The posterior distribution of \( \beta \) with conditional of \( t^n \) is represented by \( p(\beta|t^n) \), and it is given by
\[
p(\beta|t^n) = \frac{l(\lambda^n, \beta|t^n)k(\beta)}{\int_0^{\infty} l(\lambda^n, \beta|t^n)k(\beta) d\beta} = \prod_{i=1}^{n} \frac{\beta^{-b_i} t_i^{a-1} \exp [-h \beta]}{J_{\beta}(c, a + 1, t_i)},
\]
where
\[
l(\lambda^n, \beta|t^n) = \prod_{i=1}^{n} \int_0^{\infty} f(t_i|\lambda_i, \beta)g(\lambda_i|a, b_i) d\lambda_i = \prod_{i=1}^{n} \frac{\beta^{-b_i} t_i^{a-1} \exp [-h \beta]}{J_{\beta}(c, a + 1, t_i)}.
\]

According to the maximization of form (15), it makes possibility to solve the likelihood estimation. In this case, the parameters for the estimation mean for \( p, W \) and \( a \). However the parameter \( a \) has the prior distribution, and an estimation of \( \beta \) is derived based on the expectation of the form (13), which is
\[
E\{\beta|t^n\} = \prod_{i=1}^{n} \frac{J_{\beta}(c + 1, a + 1, t_i)}{J_{\beta}(c, a + 1, t_i)}.
\]

However, the form (12) is involved in the form (15), and the numerical solution for the maximum likelihood is complicated. In this paper, we propose using the Laplace’s method to approximate the marginal likelihood which is given by the form (15). Let the form (15) to be represented by \( l_{m}(\lambda^n|t^n) \), and the Laplace’s approximation is given by
\[
l_{m}(\lambda^n|t^n) \approx l(\lambda^n, \beta_{\text{BMO}}|t^n)k(\beta_{\text{BMO}}) \times \frac{(2\pi)^{\frac{1}{2}}}{n^\frac{1}{2} |\mathbf{i}(\beta_{\text{BMO}})|^\frac{1}{2}},
\]
where \( \beta_{\text{BMO}} \) means the posterior mode which is derived by argmax_\beta l(\lambda^n, \beta|t^n)k(\beta), and \( \mathbf{i}(\beta_{\text{BMO}}) \) is given by
\[
\mathbf{i}(\beta_{\text{BMO}}) = -\frac{\partial^2 l(\lambda^n, \beta|t^n)k(\beta)}{\partial \beta^2} |_{\beta=\beta_{\text{BMO}}},
\]
4 Computational Bayesian Inference

4.1 MCMC

The Gibbs sampling is used to estimate of the posterior distribution. Let a parameter vector \( \theta \) be divided as \( \theta = [\theta_1, \theta_2, \ldots, \theta_k] \), and the posterior distribution is denoted by \( p(\theta_1, \theta_2, \ldots, \theta_k | y) \), the conditional posterior distributions could be given as \( p(\theta_i | \theta_{-i}) \), \( i = 1, 2, \ldots, k \).

The notation \( \theta_{-i} \) means the parameter vector excluding \( \theta_i \), i.e., \( \theta_{-i} = [\theta_1, \theta_2, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_k] \).

The Gibbs sampling algorithm is performed as the following repeats.

\[
\begin{align*}
j & = 1, 2, \ldots, M, \\
\theta_{1(j)} & \sim p(\theta_1 | \theta_{-1(j)}), \\
\theta_{2(j)} & \sim p(\theta_2 | \theta_{-2(j)}, \theta_{1(j)}), \\
\theta_{3(j)} & \sim p(\theta_3 | \theta_{-3(j)}, \theta_{1(j)}, \theta_{2(j)}), \\
& \vdots \\
\theta_{k(j)} & \sim p(\theta_k | \theta_{-k(j)}, \theta_{1(j)}, \ldots, \theta_{k-1(j)}). 
\end{align*}
\]  

(17)

It is considered that the samples \( \theta_{i(j)} \), \( i = 1, 2, \ldots, k \) are distributed by the posterior distribution \( p(\theta_1, \theta_2, \ldots, \theta_k | y) \) for the enormous number of \( j \). In actual, the burn-in period \( B \) is assumed and the samples for \( j = B + 1, B + 2, \ldots, M \) are used for the estimation of the posterior distribution.

The Gibbs sampling is performed by the Monte Carlo simulation based on the conditional posterior distributions. However, it is possible that the simulation is not permitted due to the form type of the conditional distribution. The Metropolis-Hasting (MH) method is used to simulate random variables based on any distribution. Let \( \theta = \theta^{(i)} \) be given, and \( \theta' \) is assigned as \( \theta^{(i+1)} \) which is proposed by \( q(\theta, \theta') \). The algorithm is handled as following, that is

\[
\alpha(\theta, \theta') = \begin{cases} 
\min \left( \frac{q(\theta')q(\theta)}{q(\theta)q(\theta')}, 1 \right), & \text{if } p(\theta)q(\theta', \theta) > 0, \\
1, & \text{if } p(\theta)q(\theta', \theta) = 0,
\end{cases}
\]

and \( u \) is generated by the normal distribution \( N(0, 1) \). If \( \alpha(\theta, \theta') < u \) then \( \theta' \) is accepted, not if then it is rejected. Using the substitute distribution \( q(\theta, \theta') \), the value is generated according to any conditional distribution \( p(\theta') \), if the \( \alpha(\theta, \theta') \) is accepted. We use the MH method in the algorithm of Gibbs sampling.

4.2 Computational Method for the Bayesian Weibull Model

The hierarchical Bayesian model, which is described by the forms (1),(2),(3),(4), has the parameters \( p, \beta, a, \) and \( \lambda^r \). It is needed that the parameters are estimated based on the data \( t^* \). It is proposed that the MCMC is addressed for the estimations applying to the conditional posterior distributions (5) or (11), (13) in particular, we use the maximization of the marginal likelihood (16) for the MCMC procedure. The computational procedure for the estimations is as follows,

1. hyper parameters \( c, h, \gamma \) are given,
2. Initial values of \( \alpha \sim \pi(\alpha) \), \( p, W \) are given,
3. \( \hat{\beta}_{BMO} \sim \arg\max \beta (\lambda^*, \beta | t^* ) k(\beta) \)
4. \( p, W, a \sim \arg\max \beta p, W, a l_m(\lambda^* | t^*) \)
5. \( \beta \sim p(\beta | t^*) \)
6. \( \lambda_i \sim p(\lambda_i | t_i, \beta), i = 1, 2, \ldots, n \)
7. the repetitions from [3] to [6] are carried out.

In [6], it is possible to use \( p(\lambda_i, \beta | t_i) \) which is given by the form (11), substituting the form (5).

5 Information Criterion

The Kullback-Leibler (KL) information is denoted by \( KL(g : f) \), and the random variables \( N, U \) have the same distribution as the probability model for \( n^K \). Then \( KL(g : f) \) is given by

\[
KL(g : f) = \int_0^\infty \frac{g(x|\theta_o)}{f(x|\theta)}g(x|\theta_o)dx
\]

\[
= \int_0^\infty \log f(x|\theta)g(x|\theta_o)dx
\]

\[
- \int_0^\infty \log f(x|\theta)g(x|\theta_o)dx,
\]

where \( g(x|\theta_o) \) is the probability density function for the true model, and \( f(x|\theta) \) is the probability density function for the fitted model. When \( KL(g : f) \geq 0 \) is close to 0, the fitted model means to be close to the true model.

The positive term of \( KL(g : f) \) is unknown, but the negative term means \( K \) times the expected mean log likelihood which is denoted by ELL and given by

\[
ELL = EN \left\{ E_U \left\{ \log l(\hat{\theta}(n^K) | u^K) \right\} \right\}
\]

\[
= KE_N \left\{ \int_0^\infty \log f(u|\hat{\theta}(n^K))g(u|\theta_o)du \right\}.
\]

However \( \log l(\hat{\theta}(n^K) | n^K) \) is an estimate of ELL, then the bias correction is necessary to the estimation of ELL. EIC consists of the bias correction based on the data \( n^K \) and the bootstrap samples, therefore the bias \( C^* \) is given by

\[
C^* = EN \left\{ \log l_1(\hat{\theta}_A(n^K) | n^K) - l(\hat{\theta}_A(n^K) | n^K) \right\}
\]

\[
+ EN \left\{ E_{U^*} \left[ \log l(\hat{\theta}_A(n^K) | u^K) \right] \right\}
\]

\[
- E_{U^*} \left[ \log l(\hat{\theta}_A(n^K) | u^K) \right],
\]

where \( X^*, U^* \) are bootstrap samples and \( \hat{\theta}_A(n^K) \) is any estimator based on the data \( n^K \). In this paper, the
bootstrap samples are driven by the fitted model and estimated parameters under \( n^K \). EIC is defined as
\[
EIC = -2 \times \log(l(\hat{\theta}(n^K)|n^K)) + 2 \times C^*.
\]
In the Bayesian methodology of this paper, the values of EIC are calculated based on the marginal likelihood functions derived by (14), where \( \hat{\theta}_A = \hat{\theta}_{MCMC} \) and \( \log(l(\hat{\theta}_{MCMC}(t^n)|t^n)) \) is given. Note that \( \hat{\theta}_{MCMC}(t^n) \) means the estimate based on the MCMC of the the Bayesian methodology.

6 Concluding Remarks

In this paper, the Bayesian dynamic method for analysis of software debugging process data has been handled with the Boltzmann machine, and the aim is to estimate the prediction of the reliability for each debugging stage. In particular, the hierarchical prior models have been structured, and empirical expert knowledge priors have been supposed using the hype parameters. The estimation method for the hierarchical model have been represented based on Bayesian principles. The Bayesian inferences have been derived based on the computational simulation methods. The reliability has been estimated based on the posterior mean \( E\{\lambda_i|t_i\} \) or \( E\{\lambda_i|t_i,\beta\} \), \( i=1,2,\ldots,n \). It has been shown that the information criterion EIC is effective to choose an appropriate model involving the prior information.

References

[1] Xie, M., *Software Reliability Modelling*, World Scientific, 1991.

[2] Singpurwalla, N. D. and Wilson, S. P., *Statistical Methods in Software Engineering*, Springer, 1999.

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

[4] Jelinski, Z. and Moranda, P. B., Software reliability research, in Statistical Computer Performance Evaluation, eds: Freiberger, W., Academic Press, 465-497, 1972.

[5] Kuo, L., Software reliability, in Bayesian Thinking: Modeling and Computation, Handbook of Statistics Vol. 25, eds: Dey, D. K. and Rao, C. R., Elsevier, Amsterdam, Chapter 33, 929-963, 2005.

[6] Kaise, T., Reliability analysis of software debugging processes based on hierarchical Bayesian methodologies for remaining errors. Proceedings of the International Symposium on Stochastic Systems Theory and Its Applications, 44, 168-172, 2012.

[7] Chen, M. H., Shao, Q. M. and Ibrahim, J. G., *Monte Carlo Methods in Bayesian Computation*, Springer, 2000.

[8] Kaise, T., Bayesian modeling and estimations for reliability predictions of software debugging processes based on the Weibull distribution. Proceedings of the International Symposium on Stochastic Systems Theory and Its Applications, 48, 141-145, 2016.

[9] Tierney, L. and Kadane, J. B., Accurate approximations for posterior moments and marginal densities, Journal of the American Statistical Association, Vol. 81, 81-86, 1986.

[10] Konishi, S. and Kitagawa, G., *Information Criteria and Statistical Modeling*, Springer, 2008.