Variational Bayesian Approach for Interval Estimation of NHPP-based Software Reliability Models

Hiroyuki Okamura,[‡] Michael Grottke,[♯] Tadashi Dohi,[‡] and Kishor S. Trivedi[♯]
[‡]Department of Information Engineering
Graduate School of Engineering, Hiroshima University
1–4–1 Kagamiyama, Higashi-Hiroshima, 739–8527, Japan
{okamu, dohi}@rel.hiroshima-u.ac.jp
[♯]Department of Electrical & Computer Engineering
Duke University, Durham, NC 27708-0291, USA
{grottke, kst}@ee.duke.edu

Abstract

In this paper, we present a variational Bayesian (VB) approach to computing the interval estimates for non-homogeneous Poisson process (NHPP) software reliability models. This approach is an approximate method that can produce analytically tractable posterior distributions. We present simple iterative algorithms to compute the approximate posterior distributions for the parameters of the gamma-type NHPP-based software reliability model using either individual failure time data or grouped data. In numerical examples, the accuracy of this VB approach is compared with the interval estimates based on conventional Bayesian approaches, i.e., Laplace approximation, Markov chain Monte Carlo (MCMC) method, and numerical integration. The proposed VB approach provides almost the same accuracy as MCMC, while its computational burden is much lower.

Keywords: Software reliability, non-homogeneous Poisson process, interval estimation, variational Bayes

1. Introduction

Software reliability is one of the most important metrics of software quality. During the last three decades, many software reliability models (SRMs) have been proposed [10]. In particular, SRMs based on the non-homogeneous Poisson process (NHPP) have gained much popularity for describing the stochastic development of the number of failures experienced over time.

Much of the past research in software reliability modeling has focused on the point estimation of the model parameters as well as the reliability itself. How to take into account the uncertainty of the estimates by using interval estimation has not been fully discussed. The most commonly applied interval estimation technique is based on the central limit theorem, assuming the availability of a large number of samples [12]. However, in real-world testing the number of software failures observed is usually not large enough to justify the application of the central limit theorem.

On the other hand, Bayesian approaches can produce interval estimates even in the case of small sample sizes, by utilizing prior knowledge [15]. Since the posterior distribution is derived from both the prior distribution of the parameters and the likelihood of the observed data, Bayesian statistics contains the likelihood-based statistical analysis (such as maximum likelihood estimation) from a mathematical point of view. In the Bayesian framework, interval estimates are derived from the quantiles of the posterior distribution. The calculation is based on the analytical expression for the posterior distribution, if such an expression is feasible. For example, Meinhold and Singpurwalla [11] present the explicit form for the posterior distribution in the Jelinski-Moranda model [6].

However, for almost all SRMs the posterior distribution is mathematically complicated; therefore, it usually needs to be either simulated or approximated. Kuo and Yang [8, 9] propose the application of the Markov chain Monte Carlo (MCMC) approach to compute the posterior distributions in several types of NHPP-based SRMs. This approach can provide very accurate results if a large number of parameter samples is generated. However, the time required for computing such large samples can get very long. Yin and Trivedi [20] use direct numerical integration in the context of Bayesian analysis of the Goel-Okumoto model [5] and the delayed S-shaped model [18]. While direct numerical integration can produce very accurate results, it is vulnerable to round-off and truncation errors. Multiple-precision arithmetic can partially reduce the round-off errors, but the problem of truncation errors remains. To overcome these computational problems, Okamura *et al.* [13] propose a variational Bayesian (VB) approach for the Goel-Okumoto model [5]. While this VB approach leads to a simple algorithm, its underlying assumptions are too restrictive for achieving a good approximation of the posterior distribution.

In this paper, we improve upon the previous work related to the VB approach [13] in three ways: Firstly, we relax the central assumption in the VB approach, which enhances the accuracy of the approximation. While the computation time required by the proposed method is still less than for the MCMC approach, its accuracy is comparable to the one attained by MCMC. An additional advantage over MCMC is that the posterior distribution resulting from our approach is analytically tractable. Secondly, we increase the set of applicable models by presenting a VB algorithm for the gamma-type NHPP-based SRM, which contains the Goel-Okumoto model and the delayed S-shaped model as special cases. Thirdly, we extend the applicability of the VB approach to the grouped data case. While almost all previous results for the Bayesian estimation of SRMs have relied on failure time data, grouped data are easier to collect and thus more widely available. The derivation of methods for the analysis of grouped data is therefore important from a practical point of view.

The rest of this paper is organized as follows: Section 2 describes NHPP-based SRMs of the finite failures category. In Section 3, we discuss the point estimation of NHPP-based SRMs, which is fundamental to the interval estimation. Section 4 is devoted to conventional Bayesian interval estimation. In Section 5, we propose a new VB approach for estimating NHPP-based SRMs. In Section 6, we carry out numerical experiments to compare the application of this VB approach to the estimation of the Goel-Okumoto model with the results obtained by conventional Bayesian approaches; these comparisons are conducted for both failure time data and grouped data. Finally, the paper concludes with a brief summary of the results and an outlook on our future research in Section 7.

2. NHPP-based software reliability models

Consider NHPP-based SRMs of the finite failures category. According to this model class, the number of faults present in the software at the beginning of testing, N, follows a Poisson probability mass function with expected value ω_0:

$$P(N = n) = \frac{\omega_0^n}{n!}\exp(-\omega_0) \qquad \text{for } n = 0, 1, \ldots. \tag{1}$$

Moreover, the failure times $Y_1, Y_2, \ldots, Y_N$ of all N faults are assumed to be independent and identically distributed (i.i.d.), following a general distribution with parameter vector $\boldsymbol{\theta}_0$, $G(\cdot;\boldsymbol{\theta}_0)$. These assumptions imply that $M(t)$, the number of failure occurrences experienced in the time interval $(0, t]$, has a Poisson probability mass function,

$$P(M(t) = m) = \frac{\Lambda(t)^m}{m!}\exp(-\Lambda(t)), \tag{2}$$

with expected value $\Lambda(t) = \omega_0 G(t;\boldsymbol{\theta}_0)$. Hence, the mean value function in the NHPP-based SRMs can be completely characterized by only the failure time distribution $G(t;\boldsymbol{\theta}_0)$. For example, an exponential failure time distribution leads to the Goel-Okumoto model [5], while the delayed S-shaped model [18] is obtained by assuming that $G(t;\boldsymbol{\theta}_0)$ is the distribution function of a 2-stage Erlang distribution.

Software reliability is defined as the probability that no failure occurs in a prefixed time interval. In the NHPP-based SRMs, the software reliability $R(t + u \mid t)$ for time period $(t, t + u]$ is given by

$$\begin{aligned} R(t+u \mid t) &= P(M(t+u) - M(t) = 0) \\ &= \exp\big(-\omega_0 G(t+u;\boldsymbol{\theta}_0) + \omega_0 G(t;\boldsymbol{\theta}_0)\big). \end{aligned} \tag{3}$$

3. Point estimation

Usually, when NHPP-based SRMs are applied, point estimates for the model parameters are determined based on the observed failure data. The most commonly used technique is maximum likelihood estimation. The maximum likelihood estimates (MLEs) of the model parameters are those parameter values for which the likelihood function attains its maximum. Since the likelihood function depends on the data structure, our discussion of the MLEs distinguishes between two kinds of data: failure time data, and grouped data.

Let $\mathcal{D}_T = \{T_1, \ldots, T_{M(t_e)}\}$ be the ordered set of failure times experienced before time t_e; i.e., $0 < T_1 < \cdots < T_{M(t_e)} \le t_e$ are the first $M(t_e)$ order statistics of the failure times $Y_1, \ldots, Y_N$. Given the parameters ω_0 and $\boldsymbol{\theta}_0$, the log-likelihood for the failure time data $\mathcal{D}_T$ is

$$\begin{aligned} \log P(\mathcal{D}_T|\omega_0, \boldsymbol{\theta}_0) = \sum_{i=1}^{M(t_e)} \log g(T_i;\boldsymbol{\theta}_0) + M(t_e)\log\omega_0 \\ - \omega_0 G(t_e;\boldsymbol{\theta}_0), \end{aligned} \tag{4}$$

where $g(t;\boldsymbol{\theta}_0)$ is the probability density function connected to the failure time distribution $G(t;\boldsymbol{\theta}_0)$. We use the probability measure $P(\cdot)$ to indicate the probability density function in the case of a continuous random variable and the probability mass function in the case of a discrete random variable.

Let $\mathcal{D}_G = \{X_1, \ldots, X_k\}$ denote the grouped data for a time sequence $s_0 \equiv 0 < s_1 < \cdots < s_k$, where X_i represents the number of failures experienced during the time interval $(s_{i-1}, s_i]$. For the grouped data $\mathcal{D}_G$, the log-likelihood is given by

$$\begin{aligned}\log P(\mathcal{D}_G|\omega_0, \boldsymbol{\theta}_0) \qquad\qquad (5)\\ = \sum_{i=1}^{k} X_i \log\left(G(s_i; \boldsymbol{\theta}_0) - G(s_{i-1}; \boldsymbol{\theta}_0)\right)\\ + \sum_{i=1}^{k} X_i \log \omega_0 - \sum_{i=1}^{k} \log X_i! - \omega_0 G(s_k; \boldsymbol{\theta}_0).\end{aligned}$$

Based on the observed data $\mathcal{D}_T = \{T_1 = t_1, \ldots, T_{M(t_e)} = t_{m_e}\}$ or $\mathcal{D}_G = \{X_1 = x_1, \ldots, X_k = x_k\}$, we can compute the log-likelihood and find the maximum likelihood estimates $\hat{\omega}_{MLE}$ and $\hat{\boldsymbol{\theta}}_{MLE}$. Since Eqs. (4) and (5) are non-linear, Newton or quasi-Newton method is traditionally applied to derive the MLEs. Recently, Okamura et al. [14] proposed a powerful iteration scheme based on the EM (Expectation-Maximization) algorithm to compute the MLEs for almost all NHPP-based SRMs. This technique is especially suitable for use in automated reliability prediction tools.

The framework of Bayesian statistics produces somewhat different point estimates. Bayesian estimation is employed to make use of prior knowledge. The key idea is to regard the parameters as random variables and to embody the prior knowledge via so-called prior distributions for these parameters. So far, we have used ω_0 and $\boldsymbol{\theta}_0$ to denote the fixed but unknown parameter values. Let ω and $\boldsymbol{\theta}$ denote the corresponding random variables. According to Bayes' theorem, the relationship between the prior density $P(\omega, \boldsymbol{\theta})$, the likelihood $P(\mathcal{D}|\omega, \boldsymbol{\theta})$ and the posterior density $P(\omega, \boldsymbol{\theta}|\mathcal{D})$ is as follows:

$$\begin{aligned}P(\omega, \boldsymbol{\theta}|\mathcal{D}) &= \frac{1}{C} \cdot P(\mathcal{D}|\omega, \boldsymbol{\theta}) P(\omega, \boldsymbol{\theta})\\ &\propto P(\mathcal{D}|\omega, \boldsymbol{\theta}) P(\omega, \boldsymbol{\theta}), \qquad (6)\end{aligned}$$

where C is a normalizing constant ensuring that the total probability is one. This equation shows how the prior knowledge is updated via the information $\mathcal{D}$. Although the Bayesian estimation produces a complete posterior density instead of single parameter estimates, point estimates can easily be derived. For example, the maximum a posterior (MAP) estimates are those parameter values for which the posterior density - or its logarithm - is maximized; i.e.,

$$\begin{aligned}(\hat{\omega}_{MAP}, \hat{\boldsymbol{\theta}}_{MAP}) = \underset{\omega_0, \boldsymbol{\theta}_0}{\text{argmax}} \Big\{ \log P(\mathcal{D}|\omega = \omega_0, \boldsymbol{\theta} = \boldsymbol{\theta}_0)\\ + \log P(\omega = \omega_0, \boldsymbol{\theta} = \boldsymbol{\theta}_0) \Big\}. \qquad (7)\end{aligned}$$

Alternatively, the first moments of the posterior distribution can also be used as point estimates.

4. Bayesian interval estimation

For many software products, only a small number of failure data points are available. In such cases, Bayesian estimation is a more effective method than MLE-based approaches for deriving interval estimates in SRMs. The main challenge in Bayesian estimation is to derive the posterior distribution. If the posterior distribution is explicitly given, the interval estimates are obtained by evaluating the quantile of posterior distribution. However, the posterior density is usually expressed in proportional form, like in Eq. (6). Therefore, except for some specific cases, the calculation of the posterior distribution is quite difficult both analytically and from the computational standpoint.

4.1. Direct methods

The simplest approach is to evaluate Eq. (6) analytically or numerically. Suppose that the failure times $Y_1, \ldots, Y_N$ follow an exponential distribution with probability density function $g(t; \beta) = \beta e^{-\beta t}$. Given the failure time data $\mathcal{D}_T = \{T_1 = t_1, \ldots, T_{M(t_e)} = t_{m_e}\}$, the joint posterior density for the parameters ω and β can be written as

$$\begin{aligned}P(\omega, \beta|\mathcal{D}_T) \propto P(\omega, \beta) \omega^{M(t_e)} \beta^{M(t_e)}\\ \times \exp\left(-\beta \sum_{i=1}^{M(t_e)} T_i - \omega(1 - e^{-\beta t_e}) \right). \qquad (8)\end{aligned}$$

To derive an interval estimate from this expression, we have to determine the normalizing constant in the equation. Yin and Trivedi [20] discuss the interval estimation based on numerical integration. When using this method, the upper and lower limits chosen for the area of integration strongly affect the interval estimates. Choosing too wide a range can cause numerical exceptions like underflows; a too narrow range, on the other hand, leads to an underestimation of the normalizing constant.

4.2. Laplace approximation

The idea behind the Laplace approximation is to approximate the joint posterior distribution of the parameters by a multivariate normal distribution. In general, the MAP estimates and the second derivatives of the posterior distribution evaluated at the MAP estimates are used as the mean vector and the variance-covariance matrix of the approximating multivariate normal distribution, respectively. If a flat prior density (i.e., a constant density over the entire joint parameter domain) is used, the Laplace approximation is reduced to the MLE-based derivation of confidence intervals discussed in [19]. The Laplace approximation does not require complicated computational procedures. However, since the multivariate normal distribution cannot account for skewness, the accuracy of the approximation is low in many cases.

4.3. Markov chain Monte Carlo

The Markov chain Monte Carlo (MCMC) approach is the most popular and versatile method to evaluate the posterior distribution. Instead of an analytical expression for the posterior distribution, the method uses sampling data generated from this posterior distribution. The main idea behind the MCMC approach is to derive samples from the joint posterior distribution of the model parameters by alternatingly applying conditional marginal densities related to the joint posterior distribution.

MCMC methods for NHPP-based SRMs are discussed by Kuo and Yang [8, 9]. For example, suppose that the failure times follow an exponential distribution, and that the prior distribution is the non-informative flat density. Let the random variable $\overline{N} = N - M(t_e)$ denote the residual number of faults at the end of testing. Given the failure time data $\mathcal{D}_T = \{T_1 = t_1, \ldots, T_{M(t_e)} = m_e\}$, Kuo and Yang [8] propose the following Gibbs sampling scheme:

$$\overline{N} \mid \omega, \beta \sim \text{Poisson}(\omega e^{-\beta t_e}), \tag{9}$$

$$\omega \mid \overline{N} \sim \text{Gamma}(m_e + \overline{N}, 1), \tag{10}$$

$$\beta \mid \overline{N} \sim \text{Gamma}(m_e, \textstyle\sum_{i=1}^{m_e} t_i + \overline{N} t_e), \tag{11}$$

where $\sim$ indicates the probability distribution. Also, ‘Poisson(a)’ and ‘Gamma(b, c)’ represent the Poisson distribution with mean a and the Gamma distribution with shape parameter b and scale parameter c. In the MCMC approach, we start from provisional values for ω and β and generate new samples based on the above sampling scheme, using the current parameter values. Repeating this procedure, we obtain samples from the joint posterior distribution of the parameters.

By introducing the residual number of faults, $\bar{N}$, Kuo and Yang are able to reduce the computational cost of sampling. However, to derive interval estimates, we have to determine quantile points from the sampling data. Unlike the computation of moments, the computation of quantiles from samples usually requires a large sample size [2]. From a computational point of view, the derivation of interval estimates via the MCMC approach is therefore not efficient. Moreover, the MCMC methods developed in [8, 9] require failure time data. The application of MCMC to grouped data requires general-purpose sampling methods such as the Metropolis-Hastings algorithm [3], which is even more computationally intensive.

5. Variational Bayesian approach

5.1. Variational posterior for NHPP-based SRMs

The variational Bayesian (VB) approach is based on the variational approximation to the posterior distribution. Unlike MCMC methods, it derives closed analytical forms for the posterior distributions.

Consider the two data sets $\mathcal{D}$ and $\mathcal{U}$, corresponding to observed and unobserved information, respectively. Note that the data sets $\mathcal{D}$, $\mathcal{U}$ together contain the complete information on the ordered software failure times $T_1 < \cdots < T_N$, where N is again the total number of faults. Denoting the set of model parameters by $\boldsymbol{\mu} = \{\omega, \boldsymbol{\theta}\}$, the marginal log-likelihood of $\mathcal{D}$ can be obtained from the complete likelihood $P(\mathcal{D}, \mathcal{U} | \boldsymbol{\mu})$ and the prior density $P(\boldsymbol{\mu})$:

$$\begin{aligned} \log P(\mathcal{D}) &= \log \int\int P(\mathcal{D}, \mathcal{U}, \boldsymbol{\mu}) d\mathcal{U} d\boldsymbol{\mu} \\ &= \log \int\int P(\mathcal{D}, \mathcal{U} | \boldsymbol{\mu}) P(\boldsymbol{\mu}) d\mathcal{U} d\boldsymbol{\mu}. \end{aligned} \tag{12}$$

Here we use $\int P(A, B) dA$ to represent the calculation of the marginal probability measure for the random variable B. That is, if A is a discrete random variable, the corresponding operation is the summation of $P(A, B)$ over all possible values of A. Otherwise, if A is a continuous random variable, the operation reduces to the integral of $P(A, B)$ over the domain of A.

Let $P_v(\cdot)$ be an arbitrary probability measure. From Jensen’s inequality, we have

$$\begin{aligned} \log P(\mathcal{D}) &= \log \int\int P_v(\mathcal{U}, \boldsymbol{\mu}) \frac{P(\mathcal{D}, \mathcal{U}, \boldsymbol{\mu})}{P_v(\mathcal{U}, \boldsymbol{\mu})} d\mathcal{U} d\boldsymbol{\mu} \\ &\geq \int\int P_v(\mathcal{U}, \boldsymbol{\mu}) \log \frac{P(\mathcal{D}, \mathcal{U}, \boldsymbol{\mu})}{P_v(\mathcal{U}, \boldsymbol{\mu})} d\mathcal{U} d\boldsymbol{\mu} \equiv \mathcal{F}[P_v], \end{aligned} \tag{13}$$

where $\mathcal{F}[P_v]$ is a functional of $P_v(\mathcal{U}, \boldsymbol{\mu})$. From Eq. (13), the difference between the marginal log-likelihood of $\mathcal{D}$ and the functional $\mathcal{F}$ can be obtained as follows:

$$\begin{aligned} &\log P(\mathcal{D}) - \mathcal{F}[P_v] \\ &\quad = \int\int P_v(\mathcal{U}, \boldsymbol{\mu}) \log \frac{P_v(\mathcal{U}, \boldsymbol{\mu})}{P(\mathcal{U}, \boldsymbol{\mu} | \mathcal{D})} d\mathcal{U} d\boldsymbol{\mu}. \end{aligned} \tag{14}$$

The right-hand side of Eq. (14) represents the Kullback-Leibler distance between $P_v(\mathcal{U}, \boldsymbol{\mu})$ and the posterior density $P(\mathcal{U}, \boldsymbol{\mu} | \mathcal{D})$. Hence the problem of minimizing the difference $\log P(\mathcal{D}) - \mathcal{F}[P_v]$ with respect to $P_v(\mathcal{U}, \boldsymbol{\mu})$ amounts to approximating the posterior density $P(\mathcal{U}, \boldsymbol{\mu} | \mathcal{D})$ with $P_v(\mathcal{U}, \boldsymbol{\mu})$ as closely as possible. The probability measure P_v is often called a *variational posterior*.

In the most general application of variational Bayes, the following assumption is made about the approximating variational posterior density:

$$P_v(\mathcal{U}, \boldsymbol{\mu}) = P_v(\mathcal{U}) P_v(\boldsymbol{\mu}). \tag{15}$$

In fact, this is also the assumption used in [13]. However, Eq. (15) implies that the unobserved data is independent of the parameters. This assumption imposes a hard restriction on the variational posterior density. The unobserved data $\mathcal{U}$ contains information about the total number of faults N. This information influences the posterior distribution. We therefore refine the assumption in Eq. (15) by separating the unobserved data $\mathcal{U}$ into the total number of faults N and the other information $\mathcal{T}$; i.e., $\mathcal{U} = \{\mathcal{T}, N\}$:

$$P_v(\mathcal{U}, \boldsymbol{\mu}) = P_v(\mathcal{T}, N, \boldsymbol{\mu}) = P_v(\mathcal{T}|N)P_v(\boldsymbol{\mu}|N)P_v(N). \tag{16}$$

This equation represents the conditional independence relationship between $\mathcal{T}$ and the model parameters, given that N is fixed. From this assumption, the optimal variational posterior densities maximizing the functional $\mathcal{F}[P_v]$ are

$$\begin{aligned} P_v(\mathcal{T}|N) &\propto \exp\left(\int P_v(\boldsymbol{\mu}|N) \log P(\mathcal{D}, \mathcal{T}, N|\boldsymbol{\mu}) d\boldsymbol{\mu} \right) \\ &= \exp\Big(\mathrm{E}_{\boldsymbol{\mu}|N}\Big[\log P(\mathcal{D}, \mathcal{T}, N|\boldsymbol{\mu}) \Big] \Big), \end{aligned} \tag{17}$$

$$\begin{aligned} &P_v(\boldsymbol{\mu}|N) \\ &\propto P(\boldsymbol{\mu}) \exp\left(\int P_v(\mathcal{T}|N) \log P(\mathcal{D}, \mathcal{T}, N|\boldsymbol{\mu}) d\mathcal{T} \right) \\ &= P(\boldsymbol{\mu}) \exp\Big(\mathrm{E}_{\mathcal{T}|N}\Big[\log P(\mathcal{D}, \mathcal{T}, N|\boldsymbol{\mu}) \Big] \Big), \end{aligned} \tag{18}$$

and

$$\begin{aligned} P_v(N) &\propto \exp\Bigg(\int\int P_v(\mathcal{T}|N)P_v(\boldsymbol{\mu}|N) \\ &\qquad\qquad \times \log \frac{P(\mathcal{D}, \mathcal{T}, N|\boldsymbol{\mu})P(\boldsymbol{\mu})}{P_v(\mathcal{T}|N)P_v(\boldsymbol{\mu}|N)} d\mathcal{T} d\boldsymbol{\mu} \Bigg) \\ &= \exp\left(\mathrm{E}_{\mathcal{T},\boldsymbol{\mu}|N}\left[\log \frac{P(\mathcal{D}, \mathcal{T}, N|\boldsymbol{\mu})P(\boldsymbol{\mu})}{P_v(\mathcal{T}|N)P_v(\boldsymbol{\mu}|N)} \right] \right) \\ &=: \tilde{P}_v(N), \end{aligned} \tag{19}$$

where $\mathrm{E}_{\boldsymbol{\mu}|N}[\cdot]$ and $\mathrm{E}_{\mathcal{T}|N}[\cdot]$ are the conditional expectations with respect to $\boldsymbol{\mu}$ and $\mathcal{T}$ under the variational posterior densities, provided that N is given. Also, $\mathrm{E}_{\mathcal{T},\boldsymbol{\mu}|N}[\cdot] = \mathrm{E}_{\mathcal{T}|N}[\mathrm{E}_{\boldsymbol{\mu}|N}[\cdot]]$. The variational posterior distribution of the parameter vector becomes the mixture-type distribution of the conditional variational posterior distributions of $\boldsymbol{\mu}$: $P_v(\boldsymbol{\mu}) = \sum_N P_v(\boldsymbol{\mu}|N)P_v(N)$.

Based on this scheme, we propose the following general VB algorithm:

Step 1: Set the range of the total number of faults N to $[m_e, n_{max}]$, where m_e (≥ 0) is the number of previously observed failures, and n_{max} is a sufficiently large number.

Step 2: Compute the conditional variational posterior densities of $\mathcal{T}$ and $\boldsymbol{\mu}$, Eqs. (17) and (18), for each $N \in [m_e, n_{max}]$.

Step 3: Evaluate $\tilde{P}_v(N)$, the unnormalized form of the variational posterior density $P_v(N)$, at each $N \in [m_e, n_{max}]$, and approximate the probability mass function $P_v(N)$ by $\tilde{P}_v(N)/\sum_{i=m_e}^{n_{max}} \tilde{P}_v(i)$.

Step 4: If the probability mass allocated to the value n_{max} by this approximated probability mass function $P_v(N)$ is smaller than the tolerance ε, i.e., if $P_v(n_{max}) < \varepsilon$, go to Step 5. Otherwise, increase n_{max} and go to Step 2.

Step 5: Return $P_v(\boldsymbol{\mu}) = \sum_N P_v(\boldsymbol{\mu}|N)P_v(N)$ as the optimal variational posterior density.

This algorithm approximates the variational posterior density while simultaneously trying to determine an adequate upper bound n_{max} for the value of N. As shown later, $P_v(N)$ is not a closed form of N. Therefore, in Step 3 the appropriateness of n_{max} is checked via the (approximated) probability mass $P_v(N)$. The approach taken here is a heuristic and has the potential for further improvement.

5.2. Computational steps in the VB algorithm

We now assume that the failure times $Y_1, \ldots, Y_N$ are i.i.d., each following a gamma distribution with scale parameter β and fixed shape parameter α_0; i.e., their common marginal probability density function is given by

$$g_{Gam}(t; \alpha_0, \beta) = \frac{\beta^{\alpha_0} t^{\alpha_0 - 1}}{\Gamma(\alpha_0)} e^{-\beta t}, \tag{20}$$

where $\Gamma(\cdot)$ denotes the standard gamma function. This class of gamma-type NHPP-based SRMs contains both the Goel-Okumoto model [5] and the delayed S-shaped model [18]. The model parameters to be estimated are ω and β. To simplify the computations, the independent prior distributions for the parameters ω and β are chosen to be gamma distributions with parameters (m_ω, ϕ_ω) and (m_β, ϕ_β), respectively.

For the complete data $\{\mathcal{D}, \mathcal{T}, N\} = \{T_1 < \cdots < T_N\}$, the complete log-likelihood under the given ω and β is

$$\begin{aligned} &\log P(\mathcal{D}, \mathcal{T}, N \mid \omega, \beta) \\ &\quad = -\omega + N \log \omega + N\alpha_0 \log \beta \\ &\quad\quad + (\alpha_0 - 1) \sum_{i=1}^{N} \log T_i - \beta \sum_{i=1}^{N} T_i - N \log \Gamma(\alpha_0). \end{aligned} \tag{21}$$

From Eq. (18), we have the following variational posterior density for the parameters ω and β:

$$P_v(\omega, \beta \mid N) \propto P(\omega)P(\beta)\omega^N \beta^N e^{-\omega - \zeta_{T|N}}, \tag{22}$$

where $\zeta_{T|N} = \mathrm{E}_{\mathcal{T}|N}\left[\sum_{i=1}^{N} T_i\right]$. Based on the conjugate gamma prior distributions for ω and β, the variational posterior distributions of these parameters are obtained as gamma distributions with parameters $(m_\omega + N, \phi_\omega + 1)$ and $(m_\beta + N\alpha_0, \phi_\beta + \zeta_{T|N})$, respectively. Moreover, from Eq. (17), the general form of the variational posterior density of $\mathcal{T}$ is obtained as

$$P_v(\mathcal{T}|N) \quad \propto \quad \prod_{i=1}^{N} g_{Gam}(T_i; \alpha_0, \xi_{\beta|N}), \qquad (23)$$

where $\xi_{\beta|N} = \mathrm{E}_{\boldsymbol{\mu}|N}[\beta]$. Note that the concrete form of the variational posterior density is determined after $\mathcal{D}$ and $\mathcal{T}$ are specified.

Given the total number of faults N, the computational step for deriving the variational posterior densities mainly consists of the computation of the expected values $\zeta_{T|N}$ and $\xi_{\beta|N}$. Basically, these expected values can be computed from non-linear equations. Since the variational posterior density of $\mathcal{T}$ depends on the data structure, this paper provides the non-linear equations of $\zeta_{T|N}$ and $\xi_{\beta|N}$ for both failure time data and grouped data under the gamma prior distribution for β.

Consider the failure time data $\mathcal{D}_T = \{T_1 = t_1, \ldots, T_{M(t_e)} = t_{m_e}\}$ collected until time t_e. Then $\zeta_{T|N}$ and $\xi_{\beta|N}$ can be calculated via the following equations:

$$\zeta_{T|N} = \sum_{i=1}^{m_e} t_i + \frac{(N - m_e)\alpha_0}{\xi_{\beta|N}} \frac{\overline{G}_{Gam}(t_e; \alpha_0 + 1, \xi_{\beta|N})}{\overline{G}_{Gam}(t_e; \alpha_0, \xi_{\beta|N})}, \qquad (24)$$

and

$$\xi_{\beta|N} = \frac{m_\beta + N}{\phi_\beta + \zeta_{T|N}}. \qquad (25)$$

On the other hand, given the grouped data $\mathcal{D}_G = \{X_1 = x_1, \ldots, X_k = x_k\}$ for the time sequence $s_0 \equiv 0 < s_1 < \cdots < s_k$, the expected values $\zeta_{T|N}$ and $\xi_{\beta|N}$ are given by

$$\zeta_{T|N} = \sum_{i=1}^{k} \frac{x_i\alpha_0}{\xi_{\beta|N}} \frac{\Delta G_{Gam}(s_i, s_{i-1}; \alpha_0 + 1, \xi_{\beta|N})}{\Delta G_{Gam}(s_i, s_{i-1}; \alpha_0, \xi_{\beta|N})} + \frac{(N - \sum_{i=1}^{k} x_i)\alpha_0}{\xi_{\beta|N}} \frac{\overline{G}_{Gam}(s_k; \alpha_0 + 1, \xi_{\beta|N})}{\overline{G}_{Gam}(s_k; \alpha_0, \xi_{\beta|N})}, \qquad (26)$$

and

$$\xi_{\beta|N} = \frac{m_\beta + N}{\phi_\beta + \zeta_{T|N}}, \qquad (27)$$

where $\Delta G_{Gam}(s_i, s_{i-1}; \alpha_0, \xi_{\beta|N})$ is the increment of $G_{Gam}(t; \alpha_0, \xi_{\beta|N})$ for time period $(s_{i-1}, s_i]$.

In the case of the Goel-Okumoto model (i.e., $\alpha_0 = 1$) and the availability of failure time data, the non-linear equation can explicitly be solved. For the other cases, we need to apply numerical techniques to solve the simultaneous equations. The simplest way is successive substitution. Such a method is guaranteed to have the global convergence property [1].

The variational posterior density $P_v(N)$ can also be derived from the complete log-likelihood. Based on the expected operations $\mathrm{E}_{\boldsymbol{\mu}}[\cdot]$ and $\mathrm{E}_{\mathcal{T}|N}[\cdot]$, we obtain

$$P_v(N) \propto \frac{\Gamma(m_\omega + N)\Gamma(m_\beta + N\alpha_0)}{(\phi_\omega + 1)^{m_\omega + N}(\phi_\beta + \zeta_{T|N})^{m_\beta + N\alpha_0}} \times C(N) \cdot \frac{1}{\xi_{\beta|N}^{N\alpha_0} \exp(-\xi_{\beta|N}\zeta_{T|N})}, \qquad (28)$$

where $C(N)$ is the normalizing constant appearing in the denominator of the conditional variational posterior density $P_v(\mathcal{T}|N)$. The form of $C(N)$ depends on the data structure. In the case of failure time data, we get

$$C(N) = \prod_{i=1}^{m_e} g_{Gam}(t_i; \alpha_0, \xi_{\beta|N}) \times \overline{G}_{Gam}(t_e; \alpha_0, \xi_{\beta|N})^{N - m_e}/(N - m_e)!. \qquad (29)$$

For grouped data, the normalizing factor is given by

$$C(N) = \prod_{i=1}^{k} \Delta G_{Gam}(s_i, s_{i-1}; \alpha_0, \xi_{\beta|N})^{x_i} \times \overline{G}_{Gam}(s_k; \alpha_0, \xi_{\beta|N})^{N - \sum_{i=1}^{k} x_i}/(N - \textstyle\sum_{i=1}^{k} x_i)!. \qquad (30)$$

6. Numerical experiments

In this section, we investigate the efficacy of our variational Bayesian approach. To this end, we use the System 17 data collected during the system test phase of a military application [4]. The data is available in two forms:

$\mathcal{D}_T$**:** failure time data consisting of the failure times (measured in wall-clock seconds) for all 38 failures experienced during the system test phase.

$\mathcal{D}_G$**:** grouped data consisting of the number of failure occurrences for each of the 64 working days of the system test phase.

The Goel-Okumoto model is applied to the above data sets. Moreover, we assume the following two scenarios about prior information:

Info: The prior information consists of good guesses of parameters. In the failure time data case, the mean and standard deviation of the prior distributions are (50, 15.8) for ω and (1.0e-5, 3.2e-6) for β. In the grouped data case, the prior distribution of β is different; its mean and standard deviation are given by (3.3e-2, 1.1e-2).

NoInfo: No informative prior information about the parameters is available. Therefore, flat prior densities on the parameters are used. The estimates are thus only based on the likelihood resulting from the observed data.

Table 1. Moments of approximate posterior distributions for $\mathcal{D}_T$ and $\mathcal{D}_G$.

	$\mathcal{D}_T$ and Info				
	E[ω]	E[β]	Var(ω)	Var(β)	Cov(ω, β)
NINT	41.78	1.11E-05	37.69	4.26E-12	-2.13E-06
LAPL	40.69	1.10E-05	36.07	4.20E-12	-1.88E-06
	-2.6%	-1.6%	-4.3%	-1.5%	-11.6%
MCMC	41.82	1.11E-05	37.50	4.27E-12	-2.21E-06
	0.1%	-0.2%	-0.5%	0.3%	3.8%
VB1	41.37	1.14E-05	34.47	2.60E-12	0
	-1.0%	1.8%	-8.5%	-39.0%	100.0%
VB2	41.75	1.12E-05	37.57	4.15E-12	-2.08E-06
	-0.1%	0.2%	-0.3%	-2.5%	-2.3%

	$\mathcal{D}_G$ and Info				
	E[ω]	E[β]	Var(ω)	Var(β)	Cov(ω, β)
NINT	48.63	2.57E-02	65.35	3.51E-05	-2.20E-02
LAPL	47.08	2.50E-02	60.21	3.53E-05	-2.15E-02
	-3.2%	-2.6%	-7.9%	0.4%	-2.5%
MCMC	48.68	2.56E-02	65.49	3.46E-05	-2.18E-02
	0.1%	-0.4%	0.2%	-1.6%	-1.1%
VB1	47.11	2.64E-02	39.26	1.23E-05	0
	-3.1%	2.8%	-39.9%	-64.9%	-100.0%
VB2	48.39	2.59E-02	63.90	3.31E-05	-2.13E-02
	-0.5%	0.8%	-2.2%	-5.9%	-3.1%

	$\mathcal{D}_T$ and NoInfo				
	E[ω]	E[β]	Var(ω)	Var(β)	Cov(ω, β)
NINT	40.66	1.23E-05	44.62	6.83E-12	-3.23E-06
LAPL	39.26	1.22E-05	41.44	6.56E-12	-2.41E-06
	-3.5%	-1.3%	-7.1%	-4.0%	-25.5%
MCMC	39.80	1.18E-05	44.11	6.84E-12	-3.78E-06
	-2.1%	-4.1%	-1.1%	0.2%	17.0%
VB1	39.20	1.22E-05	39.20	3.82E-12	0
	-3.6%	-0.8%	-12.1%	-44.0%	-100.0%
VB2	39.83	1.19E-05	44.61	6.62E-12	-3.56E-06
	-2.0%	-3.7%	0.0%	-3.1%	10.1%

	$\mathcal{D}_G$ and NoInfo				
	E[ω]	E[β]	Var(ω)	Var(β)	Cov(ω, β)
NINT	116.04	1.62E-02	2.65E+04	9.40E-05	-8.86E-01
LAPL	53.48	1.94E-02	2.37E+02	8.31E-05	-1.16E-01
	-53.9%	19.4%	-99.1%	-11.6%	-86.9%
MCMC	1.56E+03	1.03E-02	1.13E+07	9.88E-05	-1.53E+01
	1245.8%	-36.3%	42653.5%	5.1%	1627.8%
VB1	50.82	2.12E-02	5.08E+01	8.86E-06	0
	-56.2%	30.8%	-99.8%	-90.6%	-100.0%
VB2	90.50	1.64E-02	1.09E+04	7.77E-05	-5.15E-01
	-22.0%	1.1%	-59.1%	-17.3%	-41.8%

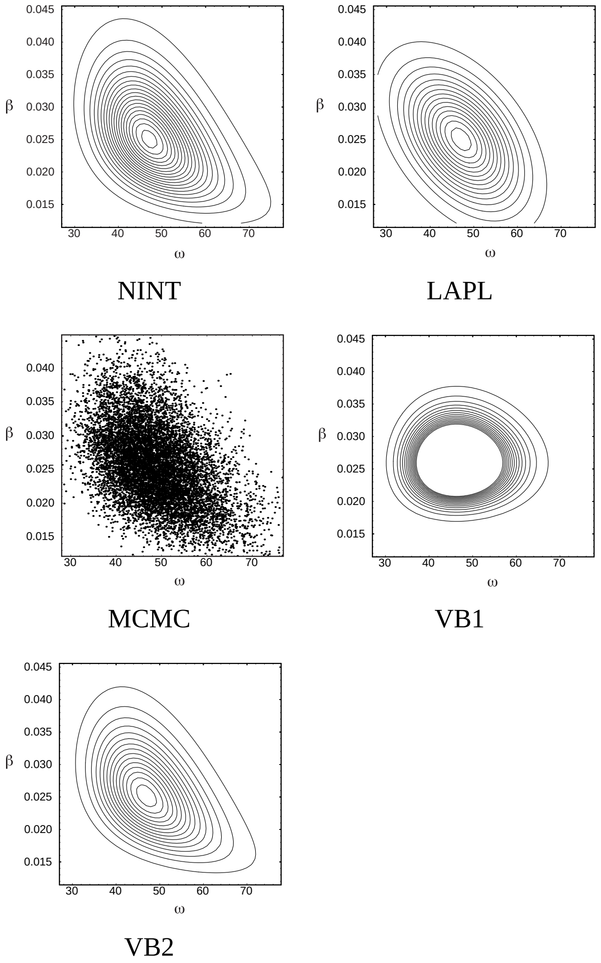
In the following, we compare the inteval estimation results for numerical integration (NINT), Laplace approximation (LAPL), Markov chain Monte Carlo (MCMC), the variational Bayesian approach proposed in [13] (VB1), and the variational Bayesian approach proposed here (VB2).

We implement all of these methods using Mathematica[1]. As noted before, one issue in numerical integration is the adequate choice of the area of integration. In our implementation of NINT, for each parameter the upper and lower limits of integration are determined based on the 0.5%- and 99.5%-quantiles of the respective marginal distribution derived for VB2: While each lower limit is chosen as the corresponding 0.5%-quantile divided by two, the upper limit is chosen as the respective 99.5%-quantile multiplied by 1.5. The numerical integration uses the multiple-precision arithmetic provided by Mathematica.

Under the MCMC method, we generate samples of the posterior distribution from one long-range MCMC series. In order to prevent dependence on the starting values of the parameters, the first 10000 samples (so-called "burn-in samples") are discarded. Moreover, at only every 10th MCMC iteration a sample is collected, to avoid autocorrelation between the samples taken. The quantiles of the posterior distribution are estimated by the corresponding order statistics. For instance, the lower bound of the two-sided 95% confidence interval for ω is estimated by the 500th smallest value of ω in all 20000 samples collected, i.e., by the empirical 2.5%-quantile. The accuracy of these estimates depends on the sample size. In our example, the sample size of 20000 gives us 95% confidence that the empirical 2.5%-quantile lies between the theoretical 2.4%- and 2.6%-quantiles [2].

Analyzing the results for all methods, we first investigate their approximation accuracy from the viewpoint of the entire posterior distribution. Table 1 presents the means, variances and covariances of the approximate joint posterior distributions of ω and β, obtained for failure time data $\mathcal{D}_T$ or grouped data $\mathcal{D}_G$. Although there can be numerical errors in the application of NINT (such as those connected to the truncation of the area of integration), we assume that NINT provides the most accurate approximation. Therefore, for all other methods we show the relative deviations from the results obtained by NINT. Except for the $\mathcal{D}_G$-NoInfo case, the first two moments calculated for NINT, MCMC and VB2 are similar. For higher moments - not shown in Table 1 - the results of NINT, MCMC and VB2 are also almost same; for example, the relative deviations of the third centralized moment $\mathrm{E}[(\omega - \mathrm{E}[\omega])^3]$ in the $\mathcal{D}_T$-Info case are 0.3% for MCMC and -0.9% for VB2. These results suggest that NINT, MCMC and VB2 provide similarly good approximations for the actual posterior distribution. On the other hand, LAPL and VB1 produce considerably worse results. For example, both methods seem to consistently underestimate the mean and the variance of ω. An important reason for the bad performance of VB1 seems to be its inability to model the correlation between ω and β. While LAPL can account for this correlation, its joint posterior density is necessarily symmetric. Why this restriction leads to biased means for ω and β, can be explained with the help

[1] Wolfram Research, Inc. `http://www.wolfram.com/`



NINT

LAPL

MCMC

VB1

VB2

Figure 1. Approximate posterior distribution for $\mathcal{D}_G$ and Info.

of Figure 1. This figure shows the contour plots for the joint posterior densities approximated by NINT, LAPL, VB1 and VB2, as well as the scatter plot using 10000 MCMC samples in the $\mathcal{D}_G$-Info case. The results for NINT, MCMC and VB2 indicate that the marginal posterior densities of ω and β are in fact right-skewed. Therefore, the MAP estimates, around which the LAPL method centers its approximate posterior distribution, tend to be smaller than the true expected parameter values. Like Table 1, the diagrams in Figure 1 suggest that NINT, MCMC and VB2 produce similar results, while LAPL and VB1 cannot approximate the joint posterior distribution as accurately as the other methods.

According to Table 1, the $\mathcal{D}_G$-NoInfo case seems to be special, because NINT, MCMC and VB2 (as well as LAPL and VB1) produce highly different results. As pointed out by several authors [7, 17, 20], in the absence of prior information accurate parameter estimates can only obtained if the failure time data itself contains enough information about the parameters. If this is not the case, the posterior distribution becomes a long-tailed distribution with large variances and covariances. Since the grouped data $\mathcal{D}_G$ of System 17 cannot be fitted by the Goel-Okumoto model as well as the failure time data $\mathcal{D}_T$, it contains less information about the parameters than $\mathcal{D}_T$. Therefore, in the absence of an informative prior distribution, none of the Bayesian methods provides accurate estimates.

Table 2. Two-sided 99% confidence intervals ($\mathcal{D}_T$).

		ω_{lower}	ω_{upper}	β_{lower}	β_{upper}
Info	NINT	27.74	59.45	6.27E-06	1.69E-05
	LAPL	25.22	56.16	5.70E-06	1.63E-05
		-9.1%	-5.5%	-9.1%	-3.7%
	MCMC	27.81	59.28	6.20E-06	1.67E-05
		0.2%	-0.3%	-1.1%	-1.0%
	VB1	27.81	58.05	7.63E-06	1.59E-05
		0.2%	-2.4%	21.7%	-5.6%
	VB2	27.73	59.38	6.41E-06	1.69E-05
		-0.1%	-0.1%	2.2%	0.0%
NoInfo	NINT	25.68	60.35	5.98E-06	1.95E-05
	LAPL	22.68	55.84	5.58E-06	1.88E-05
		-11.7%	-7.5%	-6.6%	-3.8%
	MCMC	24.88	59.59	5.53E-06	1.90E-05
		-3.1%	-1.3%	-7.4%	-2.9%
	VB1	24.95	57.20	6.92E-06	1.56E-05
		-2.8%	-5.2%	15.7%	-20.0%
	VB2	24.95	59.68	5.68E-06	1.90E-05
		-2.8%	-1.1%	-5.0%	-2.7%

The interval estimates for the parameters ω and β obtained by the different approaches confirm our findings. Tables 2 and 3 present the two-sided 99% confidence intervals for $\mathcal{D}_T$ and $\mathcal{D}_G$, respectively. As before, for LAPL, MCMC, VB1 and VB2, we compute the relative deviations from the results obtained for NINT. Again, with the exception of the $\mathcal{D}_G$-NoInfo case, the results calculated by MCMC and VB2 are close to the ones derived by NINT. The fact that the interval estimates calculated by LAPL are usually shifted to the left is explained by the bias in the estimated expected values discussed earlier. On the other hand, because VB1 underestimates the variances of the parameters, it tends to derive interval estimates that are too narrow.

Since none of the methods produces reliable results in the $\mathcal{D}_G$-NoInfo case, we focus our attention on the Info case for the rest of this paper.

We now discuss the statistical inference on the software reliability. Tables 4 and 5 present the point estimates and two-sided 99% confidence intervals of software reliability for $\mathcal{D}_T$ and $\mathcal{D}_G$. The reliability is estimated for the time intervals $(t_e, t_e + u]$, where $u \in \{1000, 10000\}$ for $\mathcal{D}_T$ and

Table 3. Two-sided 99% confidence intervals ($\mathcal{D}_G$).

		ω_{lower}	ω_{upper}	β_{lower}	β_{upper}
Info	NINT	31.20	73.80	1.27E-02	4.29E-02
	LAPL	27.09	67.06	9.72E-03	4.03E-02
		-13.2%	-9.1%	-23.7%	-6.0%
	MCMC	31.15	73.46	1.29E-02	4.23E-02
		-0.2%	-0.5%	1.3%	-1.3%
	VB1	32.54	64.81	9.93E-03	1.84E-02
		4.3%	-12.2%	-22.1%	-57.1%
	VB2	31.10	73.18	1.35E-02	4.28E-02
		-0.3%	-0.8%	5.6%	-0.3%
NoInfo	NINT	31.86	788.14	7.59E-04	4.25E-02
	LAPL	13.84	93.12	<-4.11E-03>	4.29E-02
		-56.6%	-88.2%	-641.9%	0.9%
	MCMC	32.51	1.85E+04	3.06E-05	3.83E-02
		2.1%	2242.6%	-96.0%	-9.8%
	VB1	34.33	71.05	1.43E-02	2.97E-02
		7.8%	-91.0%	1788.1%	-30.1%
	VB2	31.11	583.35	1.06E-03	4.09E-02
		-2.4%	-26.0%	39.3%	-3.7%

Table 4. Interval estimation for software reliability ($\mathcal{D}_T$ and Info).

		reliability	lower bound	upper bound
$u = 1000$	NINT	0.9791	0.9483	0.9946
	LAPL	0.9802	0.9580	<1.0024>
	MCMC	0.9790	0.9474	0.9945
	VB1	0.9806	0.9607	0.9933
	VB2	0.9792	0.9492	0.9946
$u = 10000$	NINT	0.8200	0.5974	0.9513
	LAPL	0.8268	0.6448	<1.0087>
	MCMC	0.8192	0.5919	0.9502
	VB1	0.8314	0.6795	0.9391
	VB2	0.8210	0.6029	0.9513

Table 5. Interval estimation for software reliability ($\mathcal{D}_G$ and Info).

		reliability	lower bound	upper bound
$u = 1$	NINT	0.7907	0.6618	0.9015
	LAPL	0.7678	0.6281	0.9075
	MCMC	0.7901	0.6629	0.8998
	VB1	0.7987	0.7202	0.8688
	VB2	0.7923	0.6637	0.9015
$u = 5$	NINT	0.3382	0.1353	0.6198
	LAPL	0.2829	0.0283	0.5374
	MCMC	0.3369	0.1359	0.6149
	VB1	0.3480	0.2080	0.5173
	VB2	0.3413	0.1374	0.6197

$u \in \{1, 5\}$ for $\mathcal{D}_G$.

For NINT, the point estimate $\hat{R}(t_e + u|t_e)$ and the p-quantile $\hat{R}_p^{-1}(t_e+u|t_e)$ of software reliability are computed via the following equations:

$$\hat{R}(t_e + u|t_e) = \int_0^\infty \int_0^\infty \exp\{-\omega(e^{-\beta t_e} - e^{-\beta(t_e+u)})\} P(\omega, \beta|\mathcal{D}) d\omega d\beta, \tag{31}$$

$$\int_0^\infty \int_{\frac{-\log \hat{R}_p^{-1}(t_e+u|t_e)}{e^{-\beta t_e} - e^{-\beta(t_e+u)}}}^\infty P(\omega, \beta|\mathcal{D}) d\omega_0 d\beta = p. \tag{32}$$

Since Eq. (32) is non-linear, we use the bisection method to derive $\hat{R}_p^{-1}(t_e + u|t_e)$. For VB1 and VB2, $P(\omega, \beta|\mathcal{D})$ is replaced by the respective variational posterior density. The point estimate in LAPL is obtained by directly plugging $\hat{\omega}_{MAP}$ and $\hat{\beta}_{MAP}$ into Eq. (3), and the confidence interval is computed from the first derivatives of software reliability with respect to ω and β. For MCMC, we calculate the software reliability for all 20000 parameter samples and derive the point estimate and the interval estimate as the sample mean and the respective order statistics of all reliability values computed.

The results for NINT, MCMC and VB2 shown in Tables 4 and 5 are almost the same for both point estimation and interval estimation. Thus the accuracies of the software reliability estimates can be considered high. The other two methods produce considerably worse results. Especially, since VB1 underestimates the variance of the model parameters, the estimated intervals of software reliability derived by this method tend to be too narrow.

Finally, we discuss computational speed. Since the computation of NINT relies on VB2 to detect the area of integration, we focus only on MCMC and VB2. The computational speed of MCMC and VB2 directly depends on the number of samples and the truncation point n_{max}, respectively. Tables 6 and 7 show the computation times for the MCMC and VB2 algorithms we implemented using Mathematica. In MCMC, we measure the computation time to collect 20000 samples; taking into account the burn-in samples and the fact that a sample is only collected at every tenth iteration, the total number of Poisson and gamma random variates to be generated is 630000 (=3·(10000+10·20000)). In the grouped data case, in which we additionally use data augmentation [16] at each stage, to generate the 38 failure times, the total number of Poisson, gamma and exponential random variates to be computed is 8610000 (=(3+38)·(10000+10·20000)). The computation time for VB2 is measured for the truncation points $n_{max} \in \{100, 200, 500, 1000\}$. Here we use successive substitution to solve the non-linear equations (24)–(27). Even for $n_{max} = 1000$, the computation time of VB2 is considerably smaller than the one for MCMC. However, such a large value of n_{max} is not necessary, as can be seen from the respective probability masses $P_v(n_{max})$ shown in Table 7. For example, given a tolerance of ε =5e-15, the criterion $P_v(n_{max}) < \varepsilon$ is already fulfilled for $n_{max} = 200$. The values in Table 7 suggest that the computation time of

Table 6. Computation time for MCMC.

Data	random variates	time (sec)
$\mathcal{D}_T$ and Info	630000	541.97
$\mathcal{D}_G$ and Info	8610000	4036.38

Table 7. Computation time for VB2.

Data	n_{max}	$P_v(n_{max})$	time (sec)
$\mathcal{D}_T$ and Info	100	2.35e-11	0.56
	200	4.48e-21	1.44
	500	3.67e-46	6.59
	1000	1.94e-86	23.22
$\mathcal{D}_G$ and Info	100	1.49e-06	13.28
	200	2.66e-15	58.32
	500	6.56e-40	369.53
	1000	4.85e-80	1429.41

our current VB2 algorithm increases disproportionally with n_{max}. This is because the complexity of the problem to solve the non-linear equations becomes larger as n_{max} increases. If we use fast convergence methods like the Newton method to solve the non-linear equations, then the computation time can be expected to be proportional to n_{max}.

7. Conclusions

This paper proposes a new VB approach to approximate the posterior distribution in the Bayesian estimation of NHPP-based SRMs. In particular, we have presented a concrete numerical procedure to compute the variational posterior distributions for gamma-type NHPP-based SRMs, for both failure time data and grouped data. In numerical experiments, we have compared our VB approach with conventional approximate methods. Our results indicate that the proposed method provides almost the same accuracy as MCMC, while its computational burden is much lower. For the future, we are planning to develop methods for the computation of confidence intervals using analytical expansion techniques.

Acknowledgments

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (B), Grant No. 1631011600 (2004-2006) and Scientific Research (C), Grant No. 18510138 (2006-2008).

References

[1] H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proc. 15th Conf. on Uncertainty in Artificial Intelligence*, pages 21–30, 1999.

[2] E. J. Chen and W. D. Kelton. Simulation-based estimation of quantiles. In *Proc. of the 1999 Winter Simulation Conference*, pages 428–434, 1999.

[3] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *American Statistician*, 49(4):327–335, 1995.

[4] Data & Analysis Center for Software. The software reliability dataset. http://www.dacs.dtic.mil/databases/sled/swrel.shtml.

[5] A. L. Goel and K. Okumoto. Time-dependent error-detection rate model for software reliability and other performance measures. *IEEE Trans. Reliab.*, R-28:206–211, 1979.

[6] Z. Jelinski and P. B. Moranda. Software reliability research. In W. Freiberger, editor, *Statistical Computer Performance Evaluation*, pages 465–484. Academic Press, New York, 1972.

[7] H. Joe. Statistical inference for general-order-statistics and nonhomogeneous-Poisson-process software reliability models. *IEEE Trans. Software Eng.*, SE-15:1485–1490, 1989.

[8] L. Kuo and T. Y. Yang. Bayesian computation of software reliability. *J. Comput. Graphical Statist.*, 4:65–82, 1995.

[9] L. Kuo and T. Y. Yang. Bayesian computation for nonhomogeneous Poisson processes in software reliability. *J. Amer. Statist. Assoc.*, 91:763–773, 1996.

[10] M. R. Lyu, editor. *Handbook of Software Reliability Engineering*. McGraw-Hill, New York, 1996.

[11] R. J. Meinhold and N. D. Singpurwalla. Bayesian analysis of commonly used model for describing software failures. *The Statistician*, 32:168–173, 1983.

[12] J. D. Musa, A. Iannino, and K. Okumoto. *Software Reliability - Measurement, Prediction, Application*. McGraw-Hill, New York, 1987.

[13] H. Okamura, T. Sakoh, and T. Dohi. Variational Bayesian approach for exponential software reliability model. In *Proc. 10th IASTED Int'l Conf. on Software Eng. and Applications*, pages 82–87, 2006.

[14] H. Okamura, Y. Watanabe, and T. Dohi. An iterative scheme for maximum likelihood estimation in software reliability modeling. In *Proc. 14th Int'l Symp. on Software Reliab. Eng.*, pages 246–256, 2003.

[15] N. D. Singpurwalla and S. P. Wilson. *Statistical Methods in Software Engineering*. Springer-Verlag, New York, 1997.

[16] M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *J. Amer. Statist. Assoc.*, 82:528–540, 1987.

[17] M. C. van Pul. Asymptotic properties of a class of statistical models in software reliability. *Scand. J. Statist.*, 19:235–253, 1992.

[18] S. Yamada, M. Ohba, and S. Osaki. S-shaped reliability growth modeling for software error detection. *IEEE Trans. Reliab.*, R-32:475–478, 1983.

[19] S. Yamada and S. Osaki. Software reliability growth modeling: Models and applications. *IEEE Trans. Software Eng.*, SE-11:1431–1437, 1985.

[20] L. Yin and K. S. Trivedi. Confidence interval estimation of NHPP-based software reliability models. In *Proc. 10th Int'l Symp. Software Reliab. Eng.*, pages 6–11, 1999.