# A Bayesian analysis based on beta-mixtures for software reliability models

Nam, Seungmin*, Kim, Kiwoong*, Cho, Sinsup*, and Yeo, Inkwon**

*Dept. of Statistics, Seoul National University
**Division of Mathematics and Statistical Informatics, Chonbuk National University

**Abstract**

Nonhomogeneous Poisson Process is often used to model failure times which occurred in software reliability and hardware reliability models. It can be characterized by its intensity functions or mean value functions. Many parametric intensity models have been proposed to account for the failure mechanism in real situation. In this paper, we propose a Bayesian semiparametric approach based on beta-mixtures. Two real datasets are analyzed.

## 1. Introduction

Computers have become an indispensable tool in various fields, such as industry, education, research and business. As the importance of computers are highly appreciated, software, one of the main components of computers, not only plays an important role but becomes more complicated. As a consequence, the errors inside software which are not detected can do a seriously harm.

Theoretically, it is possible to make software to be error-free, but finding software faults is not only a difficult but also an expensive work. Software reliability is different from hardware reliability in the sense that software does not wear out or burn out. Software itself does not fail; rather flaws within the software's design and implementation by humans can possibly cause a failure in its dependent system, Pham (2003). The evaluation of software reliability is essential to produce software of a good quality, quickly and efficiently. Many statistical models have been developed to measure software reliability and utilized to estimate the failure intensity of software and predict the quality of software. With a good software reliability model, programmers can determine when to release their software package more easily and rationally.

In this paper, we propose an estimation method using the beta-mixture for software reliability; Bayesian approach using the Monte Carlo Markov Chain algorithm(MCMC).

This paper is organized as follows. Section 2 introduce the incomplete beta-mixture model and its motivation. We illustrate the real data analysis in two different cases in section 3. Conclusion follows in section 4.

## 2. Incomplete Beta-Mixture model

Let N(t) be the cumulative number of failures of the software that are observed during time (0,t]. N(t) can be modeled by Nonhomogeneous Poisson process (NHPP) with mean value function m(t). N(t) can also be specified by the intensity function $\mu(t)$, which is the derivative of m(t). m(t) or $\mu(t)$ specifies the NHPP, with distribution,

$$P(N(t)=n) = \frac{m(t)^n}{n!}e^{-m(t)}, \quad (1)$$

where n=0,1,2,...

The NHPP can be classified into two classes by the property of the mean value function. If $\lim_{t\to\infty} m(t)$ is finite, then the process can be denoted by NHPP-I. Otherwise ($\lim_{t\to\infty} m(t)$ is infinite), the process is denoted by NHPP-II. Many mean value functions or intensity functions are proposed in the literature. It is noted that Goel and Okumoto(1979), Goel(1983), and Ohba-Yamada(1982) processes belong to NHPP-I and Musa-Okumoto(1984), Duane(1964), and Cox and Lewis(1966) processes belong to NHPP-II. In this paper we deal with the Incomplete beta-mixture type mean value function which includes both types, NHPP-I and NHPP-II.

The mixture model approach models an unknown c.d.f. using a dense class of mixtures of standard distributions. That is, we enrich the class of standard parametric models for a given setting by wandering nonparametrically near the standard class. It relies on standard densities

$f(x|\theta)$ as a functional basis, approximating the true density $g(x)$ of the sample as follows:

$$g(x) \simeq \hat{g}(x) = \sum_{i=1}^{r} \omega_i \, f(x|\theta_i), \qquad (2)$$

where $\sum_{i=1}^{r} \omega_i = 1$ and r is large. Mixture of distributions can model quite exotic distributions with few parameters and a high degree of accuracy.

Discrete mixtures of Beta densities provide a dense class of models for densities on [0,1], Diaconis and Ylvisaker(1985).

$$T(g(\theta)) = \sum_{l=1}^{L} \omega_l IB\{g_{0(t)}, \alpha_l, \beta_l\}, \qquad (3)$$

where $g_0$ is a centering function for $g$. Under suitable assumptions, Mallick and Gelfand(1994) show that $ET(g(\theta)) \approx T(g_0(\theta))$.

Mallick and Gelfand(1994) model link function in generalized linear model using beta-mixture model. In survival analysis, Gelfand and Mallick (1995) apply beta-mixture model to the proportional hazard model and Mallick and Walker (2003) to the frailty model. Kim et al. (2003) firstly apply beta-mixture model to reliability.

Let IB(u;c,d) be the incomplete beta function associated with the beta density with parameters c and d evaluated at u. Kim et al. (2003) propose the mean value function based on the beta-mixture

$$m(t) = G^{-1}[\sum_{l=1}^{L} \omega_l IB\{F_0(t), \alpha_l, \beta_l\}, \lambda], \quad (4)$$

where L is the number of mixands and $\omega_l$'s are the mixing weights such that $\omega_l \geq 0$ and $\sum \omega_l = 1$. The function $F_0$ denotes a centering function with probability density function $f_0$ supported in $R^+$ and $G^{-1}$ is the inverse function of a continuous distribution function $G$ indexed by unknown parameters or given constants $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_p)$, usually $p=1$ or 2, and supported in $R^+$.

If we set L=1, $\alpha_1 = \beta_1 = 1$ and G as a uniform$(0, \lambda)$, then $m(t) = \lambda F_0(t)$. Hence it belongs to NHPP-I. If L=1, $\alpha_1 = \beta_1 = 1$ and G as an Exponential(1), $m(t) = -\log(1 - F_0(t))$ and it belongs to NHPP-II, see Kim et al.(2003)

We could assume either the mixture weight to be random or the parameters of the beta densities to be random. Mathematically it is much simpler to work with the mixture weight, Gelfand and Mallick(1995). For that reason, we would fix beta density parameters and assume that mixture weight be random. We also fix the $\alpha$ and $\beta$ to provide a set of beta densities which blanket [0,1], e.g.,

$$\alpha_l = \sigma l \text{ and } \beta_l = \sigma(L - l + 1), \quad l=1,2,\ldots,L. \quad (5)$$

In this paper, we present the Bayesian inference using Gibbs sampling as a estimation method for the model using this mean value function.

$$m(t) = G^{-1}[\sum_{l=1}^{L} \omega_l IB\{F_0(t), \sigma l, \sigma(L - l + 1)\}, \lambda] \qquad (6)$$

## 3. Applications

We apply the two proposed models, the error count model and the failure truncated model, to the real data sets.

### 3.1 Error Count Model

The software failure data set for testing this model was introduced by Tohma et al. (1991). The data are recorded day by day, and have 111 observations. They are real test data of the program for monitoring and real-time control. The program consists of about 200 modules and the modules have, on average, 1000 lines of a high-level language like Fortran.

Let $D = \{(x_i, t_i), i=1, \ldots, n\}$ be the observed data, where $t_i$ is the ith observed time which is equally spaced and $x_i = N(t_{i-1}, t_i)$ is the number of failure times between $t_{i-1}$ and $t_i$ .

Fig. 1 shows the cumulative number of failure times of the testing data and estimated mean value function.

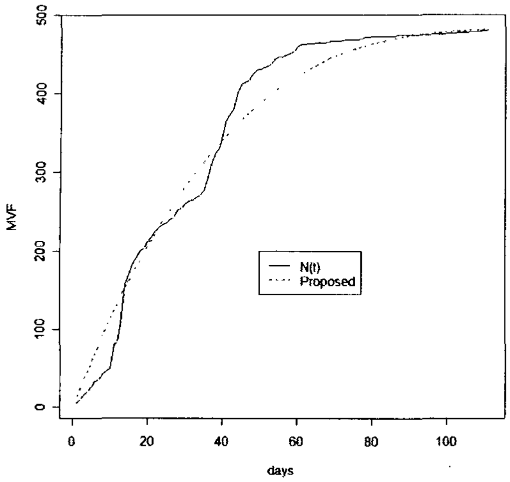


Figure 1. Cumulative number of failure times and proposed mean value function

In our model, the centering function $F_0$ is assumed to be uniform (0,115) and the number of mixand L is 3. We set $\sigma=1$ in (5). Assume that $G(\cdot)$ be the uniform (0, $\lambda$) distribution. Then, the corresponding mean value function is

$$m_\theta(t) = \lambda A_{\underline{\omega}}(t) \tag{7}$$

where

$$A_{\underline{\omega}}(t) = 3\omega_1 F_0(t)(1-F_0(t)) + 3\omega_2 F_0^2(t)(1-F_0(t)) + F_0^3(t) \tag{8}$$

For the simple expression, let

$$\begin{aligned} m_{\underline{\theta}}(t_{i-1}, t_i) &= m_{\underline{\theta}}(t_i) - m_{\underline{\theta}}(t_{i-1}) \\ &= \lambda(A_{\underline{\omega}}(t_i) - \lambda(A_{\underline{\omega}}(t_{i-1})) \\ &= \lambda B_{\underline{\omega}}(t_{i-1}, t_i) \end{aligned}$$

and

$$B_{\underline{\omega}}(t_{i-1}, t_i) = A_{\underline{\omega}}(t_i) - A_{\underline{\omega}}(t_{i-1}).$$

From (1), the likelihood function of the model is,

$$\begin{aligned} L(\underline{\theta}|D) &= \left[\prod_{i=1}^{n} \frac{[m_{\underline{\theta}}(t_{i-1}, t_i)]^{x_i}}{x_i!}\right] \exp(-m_{\underline{\theta}}(t_n)) \\ &= \left[\prod_{i=1}^{n} \frac{\lambda B_{\underline{\omega}}(t_{i-1}, t_i)]^{x_i}}{x_i!}\right] \exp(-\lambda A_{\underline{\omega}}(t_n)) \\ &= \lambda^{\Sigma x_i}\left[\prod_{i=1}^{n} \frac{[B_{\underline{\omega}}(t_{i-1}, t_i)]^{x_i}}{x_i!}\right] \exp(-\lambda A_{\underline{\omega}}(t_n)) \end{aligned} \tag{9}$$

We assume the following prior densities for $\lambda$ and $\underline{\omega}$:

$$\begin{aligned} p(\lambda) &\sim Gamma(a, b) \\ p(\underline{\omega}) &\sim Dirichlet(\alpha_1, \alpha_2, \alpha_3) \end{aligned} \tag{10}$$

It is noted that $\lambda$ and $\underline{\omega}$ are independent, and gamma distribution is semi-conjugate for $\lambda$. Hyper parameters that we use $a=1, b=0.0001$ are very popular in conventional setting and diffuse prior $\alpha_1=\alpha_2=\alpha_3=1$ makes Dirichlet to be vague prior, see Mallick and Walker(2003). Then, the full posterior density is,

$$\begin{aligned} p(\underline{\omega}, \lambda|D) &\propto Likelihood \times Prior \\ &\propto \lambda^{\Sigma x_i} \prod_{i=1}^{n} [B_{\underline{\omega}}(t_{i-1}, t_i)]^{x_i} \\ &\times \exp(-\lambda A_{\underline{\omega}}(t_n)) p(\underline{\omega}) p(\lambda) \end{aligned} \tag{11}$$

The conditional posterior densities for the Gibbs algorithm are given by,

$$p(\lambda|\underline{\omega}, D) = Gamma(n+a, A_{\underline{\omega}}(t_n) + b)$$

$$\begin{aligned} p(\underline{\omega}|\lambda, D) \propto &\left[\prod_{i=1}^{n} [B_{\underline{\omega}}(t_{i-1}, t_i)]^{x_i}\right] \\ &\times \exp(-\lambda A_{\underline{\omega}}(t_n)) p(\underline{\omega}) \end{aligned} \tag{12}$$

The sampling algorithm can be summarized as follows :

- Step 0 . Let $\underline{\omega}^0$ be initial values. Set i=1.
- Step 1 . Generate $\lambda^{(i)}$ from $Gamma(n+a, A_{\underline{\omega}^{(i)}}(t_n) + b)$
- Step 2 . Generate $\underline{\omega}^{(i)}$ using Metropolis - Hasting's Algorithm.

For each initial values for $\underline{\omega}$ (9 cases), we iterate 35000 times and drop first 5000 iterations. We consider 5010th, 5020th, ... iteration, which yields a sample of size 3000 for each case. In Table 1, we have the posterior summaries of parameters.

We monitored the convergence of the Gibbs samplers using the Gelman and Rubin(1992) method that uses the analysis of variance technique.

Table 1. Posterior summaries of parameters

| | Mean | Median | S.D. |
|---|---|---|---|
| $\lambda$ | 482.399 | 482.065 | 21.975 |
| $\omega_1$ | 0.972 | 0.975 | 0.015 |
| $\omega_2$ | 0.013 | 0.010 | 0.013 |

The model adequacy is verified as follows. The conditional predictive density for the number of future failure $X_{i+1}$ given $(x_1,\ldots,x_i)$ can be computed by

$$p(X_{i+1}|D_{x_i}) = \int p(X_{i+1}|\theta, D_{x_i})p(\theta|D_{x_i})d\theta$$
$$= \int \lambda^{X_{i+1}} \frac{[B_{\omega}(t_i, t_{i+1})]^{X_{i+1}}}{X_{i+1}!} \times \exp[\lambda B_{\omega}(t_i, t_{i+1})]p(\theta|D_{x_i})d\theta \quad (13)$$

Therefore, we can construct a $1-\alpha$ level predictive interval for $X_{i+1}$ for the Gibbs sample of $X_{i+1}$ using (13). The expression in the integral is Poisson density with mean $\lambda B_{\omega}(t_i, t_{i+1})$, so we can simulate $X_{i+1}$ with parameters $\lambda^{(r,s)}$ and $\underline{\omega}^{(r,s)}$.

Let $\lambda^{(r,s)}$, $\underline{\omega}^{(r,s)}$ denote the variate for $\lambda, \underline{\omega}$ drawn in the $r^{th}$ iteration and the $s^{th}$ replication where R and S are the total number of iterations and simulations of the Gibbs sampler, respectively.

The 95% predictive interval for $X_{i+1}$ can be constructed from the 2.5% and 97.5% quantiles of the sample $\{x_{i+1}^{(r,s)}\}_{r=1}^{R}{}_{,\,s=1}^{S}$. If 95% of the interval contain the observed $x_i$ we would conclude that the model is adequate. If we ignore the early results which are highly influenced by the prior, we can see that the predictiave interval cover most of the observed $x_i$ from Fig. 2 and we conclude that the model is adequate.

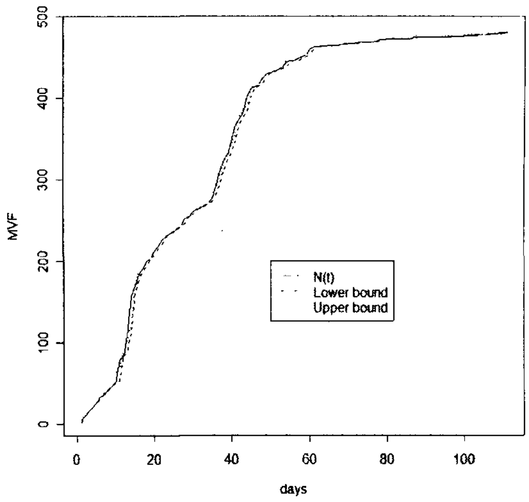


Figure 2. Cumulative failure times and predictive intervals

3.2 Failure Times model

This data set were based on the trouble report for one of the larger modules of the Naval Tactical Data System. The first 26 failures were found during the production phase; the remaining 5 errors are detected during the testing phase, Goel and Okumoto(1979) and Mazzuchi and Soyer(1988).

Let $D=\{t_1,\ldots,t_n\}$ be the failure truncated data which is the observed epoch of failure. Assume that $G(\cdot)$ be the uniform distribution on $(0,\lambda)$. And set the number of mixands $L=3$ and $\sigma=1$. The centering function $F_0$ is assumed to be uniform (0,550). Then the likelihood function of this model is

$$L(\theta|D) = [\prod_{i=1}^{n} \lambda B^{*}_{\omega}(t_i)]\exp(-\lambda\ A^{*}_{\omega}(t_n)) \quad (14)$$

where

$$A^{*}_{\omega}(t) = 3\omega_1 F_0(t)(1-F_0(t)) + 3\omega_2 F_0^2(t) \times(1-F_0(t)) + F_0^3(t)$$

and

$$B^{*}_{\omega}(t) = 3\omega_1 f_0(t)(1-2F_0(t) + 3\omega_2 f_0(t)F_0(t) \times(2-3F_0(t)) + 3f_0 F_0^2(t)$$

Note that the mean value function is

$$m_\theta(t) = \lambda A^*_{\omega}(t),$$

and the intensity function is $\mu_\theta(t) = \lambda B^*_{\omega}(t)$ .

We assume the same prior distributions as the error count model.

$$p(\lambda) \sim Gamma(a, b)$$
$$p(\omega) \sim Dirichlet(\alpha_1, \alpha_2, \alpha_3)$$

Then, the full posterior is,

$$\begin{aligned} p(\omega, \lambda | D) &\propto Likelihood \times prior \\ &\propto [\prod_{i=1}^{n} \lambda B^*_{\omega}(t_i)] \exp(-\lambda A^*_{\omega}(t_n)) \\ &\quad \times p(\omega) p(\lambda) \end{aligned} \quad (15)$$

So, the conditional posterior densities are as follows:

$$\begin{aligned} p(\lambda | \omega, D) &\propto \lambda^n \exp(-\lambda A^*_{\omega}(t_n)) p(\lambda) \\ &\propto \lambda^{n+a} \exp(-\lambda (A^*_{\omega}(t_n) + b)) \quad (16) \\ &= Gamma(n+a, A^*_{\omega}(t_n) + b) \end{aligned}$$

$$p(\omega | \lambda, D) \propto [\prod_{i=1}^{n} B^*_{\omega}(t_i)] \exp(-\lambda A^*_{\omega}(t_n)) p(\omega)$$

The Gibbs sampling algorithm is similar to that of the error count model.

Table 2. Posterior summaries of parameters

| | Mean | Median | S.D. |
|---|---|---|---|
| $\lambda$ | 32.22 | 31.90 | 5.702 |
| $\omega_1$ | 0.791 | 0.802 | 0.093 |
| $\omega_2$ | 0.101 | 0.083 | 0.086 |

The results are summarized in Table 2 and Fig. 3. Fig. 3 shows the discrepancy between $N(t)$ and the proposed mean value function. It comes from the fact that this data was obtained at two different phases. Further considerations are needed. We may need to consider larger mixands $L$, in our approach or consider the change point model.

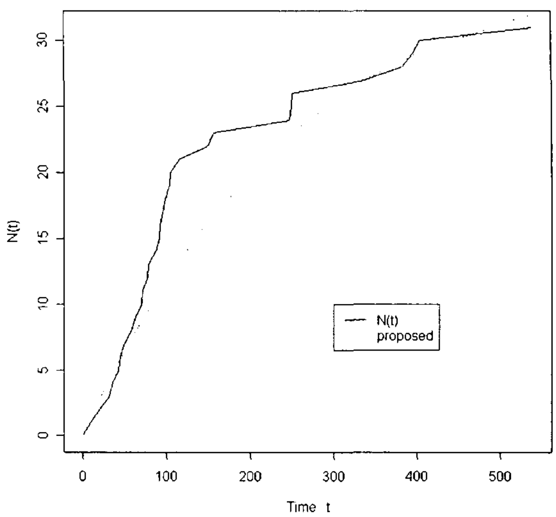


Figure 3. Cumulative number of failures and proposed mean value function

## 4. Conclusion

Software reliability models are used to monitor the faults of software. In this paper, we propose the Bayesian semiparametric approach to software reliability model using the beta-mixture. Advantages of this approach are as follows. Users can select $F_0$ and $G$ flexibly so that the mean value function can have various shapes. In general, it is difficult to infer the density with multiple parameters. To overcome this difficulties, we apply Monte Carlo technique that is Gibbs sampling to estimation. We show that the propose models are applicable to two different real data. We also discuss the predictive approach for checking model adequacy.

## References


[1] Achcar, J. A., Dey, D. K. and Niverthi, M. (1997), A Bayesian Approach using Nonhomoge -neous Poisson Process for Sofrware Reliability Models, *In Frontiers in Reliabilty*, A.P. Basu, S. K. Basu and S. Mukhopadhyay,(Eds.), 1-18,

World Scientific Publishing, Singapore.

[2] Diaconis, P. and Ylvisaker, D. (1985), Quantifying prior opinion, In *Bayesian Statistics* 2, J. M. Bernado, M. H. DeGroot, D. V. Lindley and A. F. M. Smith,(Eds.), 133-156, Amsterdam, North Holland.

[3] Gelfand, A. E. and Mallick, B. K.(1995), Bayesian Analysis of Proportional Hazard Models Built from Monotone Function, *Biometrics*, 51, 843-852.

[4] Gelman,A.E. and Rubin,D. (1992), Inference from iterative simulation using multiple sequences, *Statistical Science*, 7, 457-472

[4] Goel, A. L. and Okumoto, K. (1979), Time dependent error detiction rate model for softeare reliability and othe performance measures, *IEEE Transactions on Reliability*, 28 ,206-211.

[5] Jelinski, Z. and Moranda, P. B. (1972), Sofrware Reliability Research, In *Statistical Computer Performance Evaluation*, W. Freiburg -er,(Eds.), 465-497, New York, Academic Press.

[6] Kim, D. K., Yeo, I. and Park, D. H. (2003), Software Reliability Growth Model Based on Mixtures, Manuscript.

[7] Kuo, L. and Yang, T. Y (1996), Bayesian Computation for Nonhomogeneous Poisson Processes in Software Reliability, *Journal of the American Statistical Association*, 91, 763-772.

[8] Mazzuchi, T. A. and Soyer, R. (1988), A Bayes Empircal Bayes Model for Software Reliability, *IEEE Transactions on Reliability*, 37 , 248-254.

[9] Mallick, B. K. and Gelfand, A. E. (1994), Generalized linear models with unknown link functions, *Biometrika*, 81, 237-245.

[10] Mallick, B. K. and Walker, S. (2003), A Bayesian semiparametric transformation model incorporating frailies, *Journal of Statistical Planning and Inference*, 112, 159-174.

[11] Pham, H. (2003), Software reliability and cost models: Perspectives, comparison, and practice, *European Journal of Operational Research*, 149, 475-489.

[12] Tohma, Y., Yamano, H., Obha, M. and Jacoby,R. (1991), The estimation of parameters of the hypergeometric distribution and its application to the software reliability growth model, *IEEE Transactions on Software Engineering*, 17 (5), 483-489.