소프트웨어 신뢰도의 평가와 예측을 위한 베이지안 알고리즘

박 만 곤† Raymond H. Dean††

요 약

본 논문은 스미스의 베이지안 소프트웨어 신뢰도 성장모형을 기반으로 테스팅 단계에서의 소프트웨어 신뢰도에 대한 두가지 베이즈 추정량과 그에 대한 평가 알고리즘을 제안하는데 목적이 있다. 그 방법으로 사전정보 클래스로서 일양사전분포보다 더 일반적인 베타사전분포 BE(a,b)를 사용하였다. 그 연구 과정으로 베이지안 추정절차에 있어서 제곱오차결손함수와 해리스결손함수를 고려하고, 컴퓨터 시뮬레이션을 통해서 소프트웨어 신뢰도에 대한 베이즈추정량들과 그에 따른 알고리즘을 이용하여 평균자승오차 성능을 비교한다. 연구 결과로써 a가 크면 클수록 그리고 b가 적으면 적을수록 해리스결손함수하의 소프트웨어 신뢰도의 베이즈추정량이 평균자승오차 성능의 관점에서는 더욱 유효하고, a가 b보다 더 클때 공액사전분포인 베타사전분포상의 소프트웨어 신뢰도의 베이즈추정량이 비정보사전분포인 일양사전분포상에서 소프트웨어 신뢰도의 베이즈추정량보다는 성능이 더 좋다는 결론을 얻는다.

Bayesian Algorithms for Evaluation and Prediction of Software Reliability

Man-Gon PARK† and Raymond H. DEAN††

ABSTRACT

This paper proposes two Bayes estimators and their evaluation algorithms of the software reliability at the end testing stage in the Smith's Bayesian software reliability growth model under the beta prior distribution BE(a, b), which is more general than uniform distribution, as a class of prior information. We consider both a squared- error loss function and the Harris loss function in the Bayesian estimation procedures. We also compare the MSE performances of the Bayes estimators and their algorithms of software reliability using computer simulations. And we conclude that the Bayes estimator of software reliability under the Harris loss function is more efficient than other estimators in terms of the MSE performances as a is larger and b is smaller, and that the Bayes estimators using the beta prior distribution as a conjugate prior is better than the Bayes estimators under the uniform prior distribution as a noninformative prior when a>b.

1. Introduction

One of the most important issues in the field of software engineering is to produce highly reliable computer software. The software development procedure consists of analysis, design, coding, testing and maintenance phases sequentially. The ability to produce highly reliable computer software products is related directly to the estimation and prediction of the software reliability as a part of the performance evaluation in the testing phase. The testing phase has three successive steps: unit test, integration test, and validation test. Unit testing attempts to validate the functional performance of an individual modular component of a software system, integration testing provides a means for the construction of the software architecture while at the same time testing its function and interfaces, and validation testing verifies that all requirements have been satisfied. In these testing

· 이 논문은 92년도 교육부 대학교수 국비해외파견 연구지원에 의해 이루어짐.
† 정 회 원 : 부산수산대학교 전자계산학과 교수
†† 비 회 원 : Professor, Dept. of Electrical and Computer Engineer Univ. of Kansas
논문접수 : 1993년 12월 23일, 심사완료 : 1994년 3월 20일

steps, a developed software system is tested repeatedly to detect and correct latent software errors.

Throughout the testing phase, improvements of the reliability of the software system are made by decreasing the number of software errors and by lengthening the software inter-failure times through error corrections. This phenomenon is known as software reliability growth.

In recent years, many software reliability growth models (SRGM's) have been proposed for software reliability analysis and predictions based on both classical and Bayesian estimation methods [Jelinski and Moranda(1972), Shooman(1972), Littlewood and Verrall(1973,1974), Jewell(1985), Miller (1986), Musa, Ianninoand Okumoto1987), Bittanti(1988), Shooman(1991), and Xie(1991).

Also the problem of estimating the software reliability at the end stage under the assumption of binomial trials in the testing phase has been studied by Barlow and Scheuer(1966), Read(1971), Smith(1977), and Fard and Dietrich(1983,1987). Smith(1977) developed a Bayesian algorithm to estimate the software reliability of a development testing program using a the squared-error loss function and a uniform prior distribution at all stages. He considered an m-stage development process with improvement between each stage by ordered reliabilities and obtained the test data by assuming binomial trials at all stages. Fard and Dietrich(1987) showed that an algebraic error exists in the Bayesian algorithm of Smith(1977), and they compared the performances of the Smith's corrected estimators with other estimators of software reliability in the SRGM using computer simulations.

In this paper, we first discuss the Smith's Bayesian SRGM. In Section 3 we propose two Bayesian estimators and their evaluation algorithms of the software reliability at the end testing stage under the beta prior distribution, which is more general than uniform distribution as a class of prior information, in the Smith's Bayesian SRGM. We consider both a squared-error loss function and the Harris loss function in the Bayesian estimation procedures. In Section 4 we also compare the MSE performances of the Bayesian estimators and their algorithms of software reliability carrying out Monte Carlo simulations.

2. Smith's Software Reliability Growth Model

One of the difficulties in using stochastic models such as Markov and nonhomogeneous poisson process SRGM's during the testing phase of a developed software system is that the parameter estimation is very hard task. There are some classical methods for the parameter estimaton such as maximum likelihood, least squares, uniformly minimum variance unbiased, and others. Although they can directly be applied and their asymptotic behavior can easily be determined, they sometimes do not give adequate results [Joe and Reid(1985), and Littlewood (1981)].

To make more accurate estimation we should utilize the previous experiences which was gained from similar software and prior information about the developed software system. However for this reason many Bayesian SRGM's have been proposed for the estimation of software reliability by previous knowledge. Smith's model is one of these Bayesian SRGM's. This model has to be revaluated by the development of new software testing techniques such as mutation testing systems. Mutation analysis and mutation testing system were proposed by several authors in order to test the functional correctness of computer programs [Budd(1981) and Demillo(1980)]. The results of software testing experiments in mutation analysis can be well applied to Simith's SRGM.

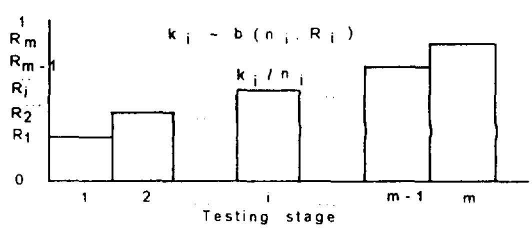
Following Smith(1977), we make the assumptions of binomial test trials in the testing phase:

<A1> Binomial test trials are assumed to be statistically independent. During the i-th stage, the probability of a successful trial is the i-th ordered reliability R_i, and there are n_i trials resulting in k_i successes ($0 \le k_i \le n_i$), $i = 1,2,\ldots,m$.

<A2> It is assumed that improvements to the software system can be made by error corrections between each stage, which implies that

$$R_1 \le R_2 \le \ldots \le R_m .$$

These assumptions are depicted in Fig.1.



(Fig. 1) Smith's software reliability growth model

3. Bayesian approaches on Smith's model

The fundamental tool used to approach at Bayesian software reliability analysis is Bayes' theorem. We may write Bayes' theorem in words as

Posterior Distribution = Prior Distribution

x Likelihood / Marginal Distribution

$\propto$ Prior Distribution x Likelihood. ....(1)

The likelihood function is the function through which the sample software test data modify prior information of the software reliability. The prior distribution rspresents all information that is known or assumed about the software reliability. The posterior distribution is a modified and updated version of the previous information expressed by the prior distribution on the basis of the observed sample software test data.

According to assumptions (A1)-(A2), the likelihood function of $R_1, R_2, \ldots, R_m$ can be written as

$$L(R_1, R_2, \ldots R_m | (n_1, k_1), (n_2, k_2), \ldots (n_m, k_m))$$

$$= m! \prod_{i=1}^{m} \binom{n_i}{k_i} R_i^{k_i} (1 - R_i)^{n_i - k_i}$$

$$\propto \prod_{i=1}^{m} R_i^{k_i} (1 - R_i)^{n_i - k_i}, \qquad \text{..........(2)}$$

where $0 \le R_1 \le R_2 \le \ldots \le R_m \le 1$, $0 \le k_i \le n_i$ for i=1,2,...,m.

3.1. Prior and posterior distributions

If we assume that the i-th ordered software reliability R_i has a beta prior distribution, the prior distribution of R_i can be written as

$$g(R_i) = \frac{1}{B(a,b)} R_i^{a-1} (1 - R_i)^{b-1},$$

$$0 \le R_i \le 1, \; a, b > 0, \qquad \text{..........(3)}$$

where B(a,b) = Γ(a)Γ(b)/Γ(a+b) is the beta function. Expanding (3) into all testing stages, we obtain the joint prior distribution of $R_1, R_2, \ldots, R_m$ as

$$g_0(R_1, R_2, \ldots, R_m)$$

$$\propto \left[\prod_{i=1}^{m} R_i \right]^{a-1} \left[\prod_{i=1}^{m} (1 - R_i) \right]^{b-1} \qquad \text{..........(4)}$$

$0 \le R_1 \le R_2 \le \ldots \le R_m \le 1$.

Using Bayes' theorem in (1), we obtain the following lemma.

Lemma 1. The joint posterior distribution of $R_1, R_2, \ldots, R_m$ given (n_1, k_1), (n_2, k_2), ..., (n_m, k_m), as the sample software test data gained by the binomial test trials, becomes

$$g_1(R_1, R_2, \ldots, R_m | (n_1, k_1), (n_2, k_2), \ldots, (n_m, k_m))$$

$$= I_m^{-1} \prod_{i=1}^{m} R_i^{k_i + a - 1} (1 - R_i)^{n_i - k_i + b - 1}, \qquad \text{..........(5)}$$

where

$$I_m = \int_0^1 \int_0^{R_m} \cdots \int_0^{R_2} \prod_{i=1}^{m} R_i^{k_i + a - 1} \bullet (1 - R_i)^{n_i - k_i + b - 1} dR_1 dR_2 \ldots dR_m, \qquad \text{..........(6)}$$

and a,b > 0, $0 \le R_1 \le R_2 \le \ldots \le R_m \le 1$, $0 \le k_i \le n_i$ for i = 1,2,...,m.

Proof. From (2) and (4), the joint posterior distribution of $R_1, R_2, \ldots, R_m$ is given by

$$g_1(R_1, R_2, \ldots, R_m | (n_1, k_1), (n_2, k_2), \ldots, (n_m, k_m))$$

$$= \frac{L(R_1, R_2, \ldots R_m | (n_1, k_1), \ldots, (n_m, k_m)) g_0(R_1, \ldots, R_m)}{\int_0^1 \int_0^{R_m} \cdots \int_0^{R_2} L(R_1, R_2, \ldots R_m | (n_1, k_1), \ldots, (n_m, k_m)) g_0(R_1, \ldots, R_m) dR_1 dR_2 \ldots dR_m}$$

$$= \frac{\prod_{i=1}^{m} R_i^{k_i + a - 1} (1 - R_i)^{n_i - k_i + b - 1}}{\int_0^1 \int_0^{R_m} \cdots \int_0^{R_2} \prod_{i=1}^{m} R_i^{k_i + a - 1} (1 - R_i)^{n_i - k_i + b - 1} dR_1 dR_2 \ldots dR_m}.$$

The joint prior distribution in (4) and the joint posterior distribution in (5) are from the same beta family. Hence, the beta prior distribution has a conjugate prior information. Also the case a=b=1 of beta prior distribution in (3) is a uniform prior distribution, as noninformative prior distribution which is desirable when we have no previous information.

The integration I_m can be calculated, although quite cumbersome, by making use of the formula of the incomplete beta integral given by Chao(1982). If u and v are positive and $0 \le R \le 1$, then we can obtain

$$\int_0^R x^{u-1}(1-x)^{v-1}dx$$

$$= B(u,v)\sum_{i=0}^{v-1}\binom{u+v-1}{u+i}R^{u+i}(1-R)^{v-1-i}$$

or

$$B(u,v)\sum_{j=u}^{u+v-1}\binom{u+v-1}{j}R^{j}(1-R)^{(u+v-1)-j} \qquad \text{.......(7)}$$

Accordingly, the integration I_m can be expressed as

$$I_m = B(k_1+a, n_1-k_1+b)\sum_{j_1=0}^{n_1-k_1+b-1}\binom{n_1+a+b-1}{k_1+a+j_1}$$

$$\bullet B(k_1+k_2+2a+j_1, n_1+n_2-k_1-k_2+2b-1-j_1)$$

$$\sum_{j_2=0}^{n_1+n_2-k_1-k_2+2(b-1)-j_1}\binom{n_1+n_2+2(a+b-1)}{k_1+k_2+2a+j_1+j_2}\bullet\bullet\bullet$$

$$\bullet B\Big(\sum_{i=1}^{m-1}k_i+(m-1)a+\sum_{i=1}^{m-2}j_i,$$

$$\sum_{i=1}^{m-1}(n_i-k_i)+(m-1)(b-1)+1-\sum_{i=1}^{m-2}j_i\Big)$$

$$\sum_{j_{m-1}=0}^{\sum_{i=1}^{m-1}(n_i-k_i)+(m-1)(b-1)-\sum_{i=1}^{m-2}j_i}\begin{bmatrix}\sum_{i=1}^{m-1}n_i+(m-1)(a+b-1)\\ \sum_{i=1}^{m-1}(k_i+j_i)+(m-1)a\end{bmatrix}$$

$$\bullet B\Big(\sum_{i=1}^{m}k_i+\sum_{i=1}^{m-1}j_i+ma, \sum_{i=1}^{m}(n_i-k_i)+mb-m+1-\sum_{i=1}^{m-1}j_i\Big). \qquad \text{.......(8)}$$

Letting

$$T_i = \sum_{j=1}^{i}(k_j+a),\ S_i = \sum_{j=1}^{i}(n_j-k_j+b-1),$$

$$J_i = \sum_{k=1}^{i}j_k, \text{ and } Q_i = T_i+S_i, \qquad \text{.......(9)}$$

the integration I_m in (8) becomes

$$I_m = \sum_{j_1=0}^{S_1}\sum_{j_2=0}^{S_2-J_1}\cdots\sum_{j_{m-1}=0}^{S_{m-1}-J_{m-2}}\prod_{\iota=1}^{m-1}C_\iota$$

$$\bullet B(T_m+J_{m-1}, S_m-J_{m-1}+1),$$

where

$$C_\iota = B(T_\iota+J_{\iota-1}, S_\iota-J_{\iota-1}+1)\binom{Q_\iota}{T_\iota+J_\iota}. \qquad \text{.......(10)}$$

In order to get the Bayes estimators of software reliability at the m-th end stage, represented by R_m, we need the marginal posterior distribution of R_m. Hence, integrating with respect to $R_1, R_2, \ldots, R_{m-1}$ from the joint posterior distribution (5), we obtain the following lemma.

<u>**Lemma 2.**</u> **The marginal posterior distribution of R_m given $(n_1,k_1), (n_2,k_2), \ldots, (n_m,k_m)$ can be written as**

$$g_2(R_m \mid (n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m))$$

$$= I_m^{-1}\sum_{j_1=0}^{S_1}\sum_{j_2=0}^{S_2-J_1}\cdots\sum_{j_{m-1}=0}^{S_{m-1}-J_{m-2}}\prod_{\iota=1}^{m-1}C_\iota$$

$$\bullet R_m^{T_m+J_{m-1}-1}(1-R_m)^{S_m-J_{m-1}}, \qquad \text{.......(11)}$$

where $0 \le R_m \le 1$, $0 \le k_i \le n_i$, and T_i, S_i, J_i and C_ι are given in (9) and (10), for $i = 1,2,\ldots,m$.

3.2. Bayes estimators of R_m using a squared-error loss function

The loss occured in estimating the software reliablity R_m by an estimator R^*_m should reflect the difference between R_m and R^*_m. The squared-error loss function which can be differentiated twice,

$$L_1(R_m, R^*_m) = (R^*_m - R_m)^2, \qquad \text{.........(12)}$$

is widely used in Bayesian estimation. For the squared-error loss function (12), the Bayes estimator of R_m can be found by minimizing the posterior expected loss function $E[L_1(R_m|(n_1,k_1),(n_2,{}_2),\ldots,(n_m,k_m)]$, and is simply the mean of the posterior distribution of R_m. Therefore, we obtain the following theorem:

Theorem 3. Under the squared-error loss function and the beta prior distribution of R_i in (3), the Bayes estimator of software reliability at the m-th end stage is given by

$$R_m^{BES} = \frac{\sum_{J_1=0}^{S_1} \sum_{J_2=0}^{S_2-J_1} \cdots \sum_{J_{m-1}=0}^{S_{m-1}-J_{m-2}} \prod_{i=1}^{m-1} C_i \bullet B(T_m+J_{m-1}+1, S_m-J_{m-1}+1)}{\sum_{J_1=0}^{S_1} \sum_{J_2=0}^{S_2-J_1} \cdots \sum_{J_{m-1}=0}^{S_{m-1}-J_{m-2}} \prod_{i=1}^{m-1} C_i \bullet B(T_m+J_{m-1}, S_m-J_{m-1}+1)}, \qquad \text{.........(13)}$$

where T_i, S_i, J_i and C_i are given in (9) and (10).

Proof. Since the Bayes estimator of R_m under the squared-error loss is the mean of the marginal posterior distribution of R_m in (11),

$$R_m^{BES} = E[R_m|(n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m)]$$

$$= \int_0^1 R_m \bullet g_2(R_m|(n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m))dR_m$$

$$= I_m^{-1} \sum_{j_1=0}^{S_1} \sum_{j_2=0}^{S_2-J_1} \cdots \sum_{j_{m-1}=0}^{S_{m-1}-J_{m-2}} \prod_{i=1}^{m-1} C_i \bullet \int_0^1 R_m^{T_m+Jm-1}(1-R_m)^{S_m-J_{m-1}} dR_m$$

Integrating of the last term,

$$\int_0^1 R_m^{T_m+Jm-1}(1-R_m)^{S_m-J_{m-1}} dR_m$$

$$= B(T_m + J_{m-1} + 1, S_m - J_{m-1} + 1),$$

which proves the theorem.

Corollary 4. The Bayes estimator of software reliability at the m-th end stage of the testing phase under the squared-error loss function and beta prior distribution, which is contained as algebraic error by Smith(1977), is converted into

$$R_m^{BESM} = \frac{\sum_{J_1=0}^{S_1} \sum_{J_2=0}^{S_2-J_1} \cdots \sum_{J_{m-1}=0}^{S_{m-1}-J_{m-2}} \prod_{i=1}^{m-1} D_i \bullet B(T_m+J_{m-1}+1, S_m-J_{m-1}+1)}{\sum_{J_1=0}^{S_1} \sum_{J_2=0}^{S_2-J_1} \cdots \sum_{J_{m-1}=0}^{S_{m-1}-J_{m-2}} \prod_{i=1}^{m-1} D_i \bullet B(T_m+J_{m-1}, S_m-J_{m-1}+1)}, \qquad \text{.........(14)}$$

where T_i, S_i and J_i are given in (9), and

$$D_i = B(T_i + J_{i-1}, S_i - J_{i-1} + 1)\binom{Q_{i+1}}{T_i + J_i} \qquad \text{.........(15)}$$

3.3. Bayes estimators of R_m using the Harris loss function

Harris(1977) suggested another loss function that depends on $1/(1-R_m)$, which is given by

$$L_2(R_m, R^*_m) = \left| \frac{1}{1-R^*_m} - \frac{1}{1-R_m} \right|^2, \qquad \text{.........(16)}$$

where R^*_m is an estimator of R_m. Higgins and Tsokos(1978) compared five different classes of loss functions, including the usual squared-error loss, squared relative error loss, linear weighted loss, Harris's loss, and exponential loss. They concluded that the squared-error loss function is not robust with respect to the other loss functions, and that Bayes estimators are very sensitive to changes of the loss function.

Under the Harris loss function in (16), the Bayes estimator of R_m can be obtained as follows;

$$R^{BEH} = 1 - \left[1 / E[\tfrac{1}{1-R_m}|(n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m)] \right], \qquad \text{.........(17)}$$

where $0 \le k_i \le n_i$ for $i = 1,2,\ldots,m$.

Then one can obtain the following theorem.

Theorem 5. Under the Harris loss function and the beta prior distribution of R_i in (3), the Bayes estimator of the software reliability at the m-th end stage is given by

$$R_m^{BEH} = \frac{\sum_{j_1=0}^{s_1} \sum_{j_2=0}^{s_2-j_1} \cdots \sum_{j_{m-1}=0}^{s_{m-1}-j_{m-2}} \prod_{\iota=1}^{m-1} C_\iota \bullet B(T_m+J_{m-1}, S_m-J_{m-1}+1)}{\sum_{j_1=0}^{s_1} \sum_{j_2=0}^{s_2-j_1} \cdots \sum_{j_{m-1}=0}^{s_{m-1}-j_{m-2}} \prod_{\iota=1}^{m-1} C_\iota \bullet B(T_m+J_{m-1}, S_m-J_{m-1})}, \qquad \cdots\cdots(18)$$

where T_i, S_i, J_i and C_ι are given in (9) and (10).

Proof. By means of (11) in Lemma 2, and (17), the expected value of $1/(1-R_m)$ given $(n_1,k_1), (n_2,k_2),\ldots,(n_m,k_m)$ is

$$E[\tfrac{1}{1-R_m}|(n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m)]$$

$$= \int_0^1 (1-R_m)^{-1} g_2(R_m|(n_1,k_1),(n_2,k_2),\ldots,(n_m,k_m))dR_m$$

$$= I_m^{-1} \sum_{j_1=0}^{S_1} \sum_{j_2=0}^{S_2-J_1} \cdots \sum_{j_{m-1}=0}^{S_{m-1}-J_{m-2}} \prod_{\iota=1}^{m-1} C_\iota \bullet \int_0^1 R_m^{T_m+J_{m-1}-1}(1-R_m)^{S_m-J_{m-1}-1} dR_m.$$

$$= I_m^{-1} \sum_{j_1=0}^{S_1} \sum_{j_2=0}^{S_2-J_1} \cdots \sum_{j_{m-1}=0}^{S_{m-1}-J_{m-2}} \prod_{\iota=1}^{m-1} C_\iota \bullet B(T_m+J_{m-1}, S_m-J_{m-1}).$$

If we replace denominator of the last term in (17) by this expected value, we can obtain the result of the theorem.

4. Monte Carlo simulations

Since the Bayes risks with respect to estimators of software reliability under squared-error loss and the Harris loss functions can not be expressed in closed mathematical form, we can not compare the Bayes risk as a measure of stabilities of the Bayes estimators by the mathematical expressions. Thus we represented as the algorithm SSRGM in Appendix to evaluate the Bayes estimates, and carried out the Monte Carlo simulations using this algorithm and IMSL software packages on VAX-9000 system of the University of Kansas computer networks(KUHUB). Monte Carlo simulation procedure is carried out as follows.

<S1> We generated 500 sets of binomial random variates according to the times of binomial testing trials and true reliability in each stage, and a=1(2)5 and b=1(2) which is the parameters of beta prior, respectively,

<S2> and evaluated the estimates of the Bayesian software reliabilities at the 9-th end stage by using Algorithm SSRGM and IMSL subroutines according to the random variates generated in S1.

<S3> We also calculated the variances, biases and mean-squared errors(MSE) of the Bayesian software reliabilities evaluated in S2 with respect to the true software reliabilities,

<S4> and calculated the relative efficiencies(REFF) of the Bayes estimates with respect to R_m^{BES} (a=1,b=1). The formula of REFF is given by

$$REFF(\hat{R}_m^1, \hat{R}_m^2) = MSE(\hat{R}_m^1)/MSE(\hat{R}_m^2),$$

where $MSE(\hat{R}_m) = E[(\hat{R}_m - R_m)^2 = Var(\hat{R}_m) + bias^2(R_m, \hat{R}_m)$, $\hat{R}_m$ is the Bayesian estimator of software reliability and R_m is the true software reliability.

Table 1 shows the performances of the proposed Bayesian estimators and their algorithms in terms of MSE and bias according to the values of a=1(2)5 and b=1(2)5. In Figures 2 through 4, we plot the graphical behaviors for REFF's of the proposed Bayesian software reliability estimators with respect to R_m^{BES}(a=1,b=1), which is the Bayesian estimator of software reliability under the uniform prior and squared-error loss proposed by Fard and Dietrich(1987).

If we define "X is(are) more efficient than Y" as X >> Y, we can obtain the following results from Tables 1 through 2 and Figures 2 through 4.

<R1> $R_m^{BEH} >> R_m^{BES} >> R_m^{BESM}$ in general.

<R2> a is larger, $R_m^{BEH} >> R_m^{BES}$.

<R3> The larger a is and the smaller b is, the more efficient R_m^{BES} and R_m^{BEH} are, but the smaller a is and the larger b is, the more efficient R_m^{BESM} is.

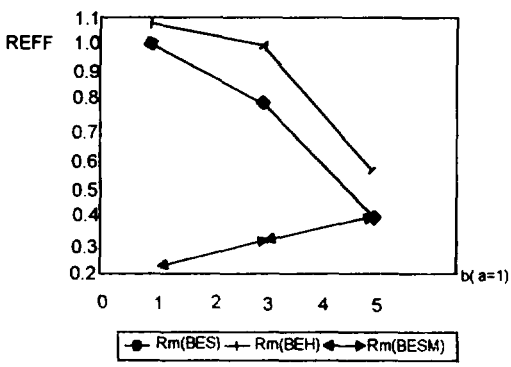
<Table 1> Comparison of MSE performances

(a, b)	MSE (R_m^{BES})	MSE (R_m^{BEH})	MSE (R_m^{BESM})
(1, 1)	0.00389	0.00369	0.01619
(1, 3)	0.00491	0.00407	0.01238
(1, 5)	0.00944	0.00730	0.00943
(3, 1)	0.00347	0.00285	0.01730
(3, 3)	0.00358	0.00324	0.01350
(3, 5)	0.00687	0.00525	0.01051
(5, 1)	0.00254	0.00251	0.01831
(5, 3)	0.00274	0.00260	0.01455
(5, 5)	0.00637	0.00331	0.01154

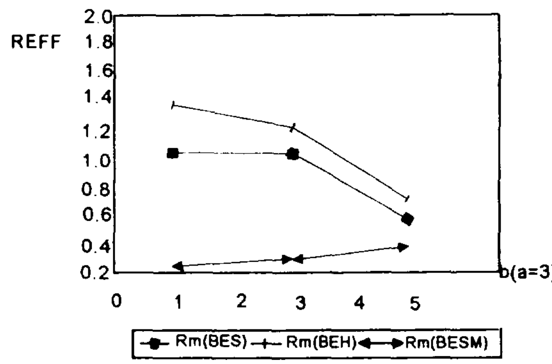
<Table 2> Comparison of RMSE performances

(a, b)	RMSE (R_m^{BES})	RMSE (R_m^{BEH})	RMSE (R_m^{BESM})
(1, 1)	0.06236	0.06075	0.12723
(1, 3)	0.07003	0.06381	0.11124
(1, 5)	0.09715	0.08545	0.09711
(3, 1)	0.05893	0.05338	0.13152
(3, 3)	0.05981	0.05692	0.11619
(3, 5)	0.08288	0.07243	0.10253
(5, 1)	0.05039	0.05010	0.13533
(5, 3)	0.05237	0.05099	0.12063
(5, 5)	0.07984	0.05751	0.10744

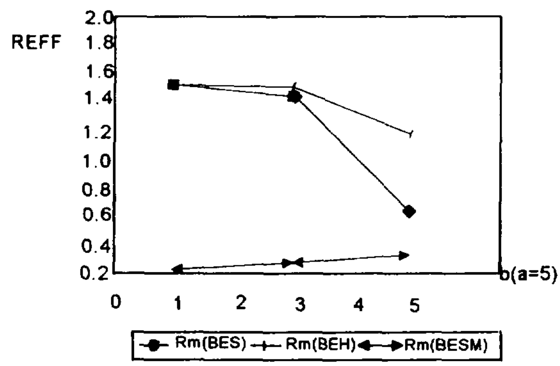
* m=9, n=30, R_m=0.85



(Fig. 2) Comparison of the relative efficiencies according to b (a=1)



(Fig. 3) Comparison of the relative efficiencies according to b (a=3)



(Fig. 4) Comparison of the relative efficiencies according to b (a=5)

5. Conclusions

It is apparent from these results that (1) the beta conjugate prior distribution BE(a,b) is more manifold than uniform prior distribution in terms of the prior selection, and the more efficient priors according to the values of two parameters a and b can be selected, (2) the Bayes estimator of software reliability under the Harris loss function is more efficient than other usual Bayes estimators in terms of the MSE performances as a is larger and b is smaller, and (3) the binomial sampling is available for attribute software test data on the Smith's Bayesian software reliability growth model.

Acknowledgment

We are grateful to Dr. James R. Rowland who is with the Department of Electrical and Computer Engineering, University of Kansas, for his helpful suggestions and comments.

Appendix

Algorithm SSRM. To evaluate the Bayesian software reliability estimators at m-th end stage in Smith's SRGM

INPUT M = number of stages in the testing phase
A and B = 1st and 2nd parameters of beta prior
N(i) and K(i) = times of testing trials and successes in the i-th stage, for i=1,2,...,M.
OUTPUT RMBES,RMBEH,and RMBESM = Bayesian software reliability estimates(1: R_m^{BES}, 2: R_m^{BEH}, 3: R_m^{BESM})

```
Step 1 : For i=1,...,M do
    for j=1,...,i do
      T(i)=T(i)+K(j)+A
      S(i)=S(i)+N(j)-K(j)+(B-1)
    end j
    Q(i)=T(i)+S(i)
  end i
Step 2 : TOTUS=0.0 : TOTLS=0.0
    TOTUH=0.0 : TOTLH=0.0
    TOTUM=0.0 : TOTLM=0.0
```

```
for j_1=0,...,S(1) do
  JT(1)=j_1
  for j_2=0,...,S(2)-JT(1) do
    JT(2)=j_1+j_2
    for j_3=0,...,S(3)-JT(2) do
      ......
      for j_{M-1}=0,...,S(M-1)-JT(M-2) do
        VAL1=1.0
        for L=1,...,M-1 do
          XA=T(L)+JT(L-1)
          XB=S(L)-JT(L-1)+1
          BT(L)=BETA(XA,XB)
          KC=Q(L) : KD=T(L)+JT(L)
          KM=Q(L+1)
          CM(L)=BINOM(KC,KD)
          DM(L)=BINOM(KM,KD)
          VAL1=VAL1*BT(L)*CM(L)
          VAL2=VAL2*BT(L)*DM(L)
        end L
        YAU=T(M)+JT(M-1)+1
        YAL=T(M)+JT(M-1)
        YBU=S(M)-JT(M-1)+1
        YBL=S(M)-JT(M-1)
        BTUS=BETA(YAU,YBU)
        BTLS =BETA(YAL,YBU)
        BTUH=BETA(YAL,YBU)
        BTLH=BETA(YAL,YBL)
      BTUM=BETA(YAU,YBU)
      BTLM=BETA(YAL,YBU)
      TOTUS=TOTUS+VAL1*BTUS
      TOTLS=TOTLS+VAL1*BTLS
      TOTUH=TOTUH+VAL1*BTUH
      TOTLH=TOTLH+VAL1*BTLH
      TOTUM=TOTUM+VAL2*BTUM
      TOTLM=TOTLM+VAL2*BTLM
      end j_{M-1}
      ......
    end j_3
  end j_2
end j_1
Step 3 : RMBES=TOTUS/TOTLS
      RMBEH=TOTUH/TOTLH
      RMBESM=TOTUM/TOTLM
      print RMBES,RMBEH,RMBESM
      end ALGORITHM SSRGM
```

References

[1] Barlow,R.E. and E.M.Scheuer, "Reliability growth during a development testing program", Technometrics, 8, pp.53-60, 1966.

[2] Bittanti,S., "Software Reliability Modelling and Identification (Lecture Notes in Computer Science)", Spring-Verlag, Berlin Heidelberg, 1988.

[3] Budd,T.A., "Mutation analysis : ideas, examples, problems and prospects", North- Holland Publishing Company, New York, pp. 129-148, 1981.

[4] Chao,A., "Another approach to incomplete integrals", The American Statistician, Vol.36(1), p. 482, 1982.

[5] Demillo,R.A., "Mutation analysis as a tool for software quality assurance", IEEE COMSPAC, pp.390-393, 1980.

[6] Fard,N.S. and D.L.Dietrich, "A Bayes reliability growth model for a development testing program", IEEE Trans. on Reliability, R-36(5), pp.568-572, 1987.

[7] Harris,B., "A survey of statistical methods in systems reliability using Bernoulli sampling of components. The Theory and Applications of Reliability with Emphasis on Bayesian and Nonparametric Methods", Vol. II, Academic, New York, pp.275-279, 1977.

[8] Higgins,J.J. and C.P.Tsokos, "A study of the effect of the loss function on Bayes estimates of failure intensity, MTBF, and Reliability", Unpublished Manuscript, University of South Florida, FL, 1978.

[9] Jelinski,Z. and P.B.Moarnda, "Software reliability research. Statistical Computer Performance Evaluation", W.Freiberger Ed., Academic Press, pp.465-484, 1972.

[10] Jewell,W.S., "Bayesian extensions to a basic model of software reliability", IEEE Trans. on Software Engineering, SE-11, pp.1465-1471, 1985.

[11] Joe,H. and N. Reid, "Estimating the number of faults in a system", J. American Stat. Assoc., 80(389), pp.222-226, 1985.

[12] Littlewood,B., "Stochastic reliability growth : A model for fault-removal in computer- programs and hardware-design", IEEE Trans. on Reliability, R-30(4), pp.313-320, 1981.

[13] Littlewood,B. and J.L.Verrall, "A Bayesian reliability growth model for computer software", J.R. Statist. Soc., Series C, 22(3), pp.332-346, 1973.

[14] Littlewood,B. and J.L.Verrall, "A Bayesian

reliability model with a stochastically monotone failure rate", IEEE Trans. on Reliability, R-23(2), pp.108-114, 1974.

[15] Miller,D.R., "Exponential order statistic model of software reliability growth", IEEE Trans. on Software Engineering, SE-12, pp.12-24, 1986.

[16] Musa,J.D., "A theory of software reliability and its application", IEEE Trans. on Software Engineering, SE-1, pp312-327, 1975.

[17] Musa,J.D., A.Iannino and K.Okumoto, "Software Reliability : Measurement, Prediction, Application", McGraw-Hill Book Company, N.Y., 1987.

[18] Ramamoorthy,C.V. and F.B.Bastani, "Soft- ware reliability status and perspectives", IEEE Trans. on Software Engineering, SE-8, pp.354 - 371, 1982.

[19] Read,R.R., "A Remark on the Barlow- Scheuer reliability growth estimation scheme", Technometrics, 13, pp.199-200, 1971.

[20] Shooman,M.L., "Probabilistic models for software reliability prediction. Statistical Computer Performance Evaluation", W.Freiberger Ed., Academic Press, New York, 1972.

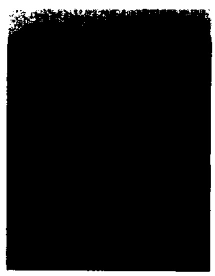
[21] Shooman,M.L., "A micro software reliability model for prediction and test apportionment", Proceedings of 1991 International Symposium on Software Reliability Engineering, Texas Austin, pp.52-59, 1991.

[22] Smith,A.F.M., "A Bayesian note on reliability growth during a development testing program", IEEE Trans. on Reliability, R-26(5), pp.346-347, 1971.

[23] Xie,M. "Software Reliability Modelling", World Scientific Publishing Co., Singapore, 1991.

박 만 곤

1976년 경북대학교 수학교육과 졸업
1987년 경북대학교 대학원 전산통계학(이학박사)
1980년~1981년 경남공전 전산학과 교수
1990년~1991년 영국 리버풀대학 전자계산학과 객원교수
1992년~1993년 미국 캔사스대학교 컴퓨터공학과 교환교수
1981년~현재 부산수산대학교 전자계산학과 교수
관심분야 : 소프트웨어 공학(신뢰성 평가모형, 결함허용 소프트웨어 신뢰도추정, 소프트웨어 품질관리 등)



Raymond H. DEAN

1958 BS, Univ. of Kansas, Electrical Engineering
1961 MS, MIT, Electrical Engineering
1968 Ph. D & MA, Princeton Univ.
1961~1985 TEC, Midwest Research Institute, RCA, CEO, TII : Member of Technical Staff, Chief Engineer
1985-Present Professor, Electrical and Computer Engineering, Univ. of Kansas
He has 29 Published Technical Papers and 21 Issued U.S. Patents in the fields of Electrical and Computer Engineering.