음이항분포 정보를 가진 베이지안 소프트웨어 신뢰도 성장모형에 관한 연구

김 희 철† · 박 종 구†† · 이 병 수†††

요 약

소프트웨어 테스팅 단계에서 소프트웨어 오류 수와 고장간격시간에 의해 소프트웨어 고장현상을 수리적으로 모형화를 하기 위하여 소프트웨어 신뢰성장모형이 사용되었다. 본 논문은 사전에 소프트웨어 오류의 총갯수가 음이항 분포를 따르고 오류 발견률이 감마분포를 따른다는 사전정보를 이용한 베이지안 방법으로 Jelinski-Moranda 모형과 Goel-Okumoto 모형 그리고 Schick-Wolverton 모형에 대한 모수추정을 하고 모형선택을 위하여 상대오차의 합과 Braun 통계량과 중위수 변량의 합을 이용하였다 모수추정 과정에서 난해한 적분을 피하기 위해서 사용된 깁스 샘플링을 이용하였고 이 기법은 마코브 체인 몬테 칼로(MCMC)기법의 일종이고 이 마코브 체인의 정상성분포는 우리가 구하고자 하는 사후분포(posterior distribution)가 된다. 모의실험 자료를 이용하여 모수추정과 모형선택의 결과를 나열하였다

Bayesian Analysis of Software Reliability Growth Model with Negative Binomial Information

Hee-Cheul Kim† · Jong-Goo Park†† · Byoung-soo Lee†††

ABSTRACT

Software reliability growth models are used in testing stages of software development to model the error content and time intervals between software failures In this paper, using priors for the number of fault with the negative binomial distribution and the error rate with gamma distribution, Bayesian inference and model selection method for Jelinski-Moranda and Goel-Okumoto and Schick-Wolverton models in software reliability. For model selection, we explored the sum of the relative error, Braun statistic and median variation. In Bayesian computation process, we could avoid the multiple integration by the use of Gibbs sampling, which is a kind of Markov Chain Monte Carlo method to compute the posterior distribution Using simulated data, Bayesian inference and model selection is studied.

1. 서 론

소프트웨어 고장들로 인한 컴퓨터 시스템의 고장은 우리 사회에 엄청난 손실을 초래할 수도 있다. 따라서 소프트웨어 신뢰도 엔지니어링에서의 연구 활동은 지난 20년 넘게 행해졌고 많은 신뢰도 성장 모델들이 소프트웨어에 남아 있는 고장들의 수와 소프트웨어 신뢰도의 추정을 위해서 제안되었다 제품이 고장이 나면 수리를 해서 사용할 수 있는 수리계 시스템(repairable system)에서의 신뢰성에 대한 연구 역시 중요한 문제이다

시스템이 고장이 나면 고장이 난 원인을 찾아 필요

※ 본 연구는 한국과학재단 지정 인천대학교 멀티미디어 연구센터의 지원에 의한 것임
† 정 회 원 : 송호대학 정보산업계열 교수
†† 종신회원 : 원광대학교 컴퓨터공학과 교수
††† 종신회원 : 시립인천대학교 컴퓨터정보통신학부 교수
논문접수 1999년 3월 22일, 심사완료 : 2000년 2월 9일

할 경우 새로운 디자인을 개발하거나 새로운 기술을 도입하게 된다 따라서, 시간이 지남에 따라 신뢰도의 증가가 기대되어 진다

이런 모형을 신뢰도 성장모형(reliability growth model)이라고 한다. 소프트웨어 신뢰성의 예측 문제에 대한 많은 경험기법들은 그들의 정도에 있어서 많은 차이를 보이고 있다. 확실한 것은 모든 환경 하에서 신뢰성 있는 결과를 만족시켜 주는 기법은 아주 어렵다. 소프트웨어 테스팅 단계에서 소프트웨어 오류 수와 고장간격시간에 의해 소프트웨어 고장 현상을 수리적으로 모형화를 하면 소프트웨어에 대한 평가를 쉽게 할 수 있으며, 신뢰도 성장모형에 의해 소프트웨어 오류 수, 소프트웨어 고장발생간격시간, 소프트웨어 신뢰도 및 고장률등의 신뢰성 평가 측도들이 추정되어 예측할 수 있다 본 논문에서는 기본적인 소프트웨어 신뢰모형인 Jelinsk-Moranda 모형[13]과 Goel-Okumoto 모형[11] 그리고 Schick-Wolverton 모형[24]에 대해 비교 연구하고자 한다. Jelinski-Moranda 모형은 소프트웨어 신뢰모형에서 가장 기본적인 모형이다. 그러나 이 모형은 모든 오류들이 같은 크기를 가진다는 가정하에서 순수하게 결정적이고 중요한 비율의 열만 취급한다는 비판이 있었다. 이러한 비판을 개선하기 위해 Goel-Okumoto가 제시한 개선된 Jelinski-Moranda모형과 Schick-Wolverton이 제시한 모형에 대해 깁스추출법을 이용하여 위험함수(hazard function)의 형태와 초기오류수의 형태의 사전분포가 음이항분포인 경우에 베이지안 추론과 모형선택에 관해 논의 하고자 한다.

깁스 샘플링은 마코브 체인 몬테 카를로(MCMC, Markov Chain Monte Carlo)기법의 일종이고 이 마코브 체인의 정상성 분포는 우리가 구하고자 하는 사후분포(posterior distribution)가 되며 즉 독립적인 출발점(independent starting points)을 가진 마코브 체인을 발전시켜서 사후분포로부터 다중 샘플들을 얻을 수 있다[9]. 본 논문은 사전분포가 음이항분포[15]를 선택하여 Jelinski-Moranda모형과 Goel-Okumoto모형 그리고 Schick-Wolverton모형에 적용을 하고, 상대오차의 합을 이용하여 모형선택을 시도 하였다. 2절에서는 본 논문에서 사용되는 신뢰도 성장모형과 베이즈 정리 그리고 깁스샘플링을 요약하였고 3절에서는 사전분포가 음이항분포를 가진 여러 모형에 대한 깁스 샘플링 알고리즘을 설명하였고 4절에서는 모형선택을 5절에서는 수치적인 예를 제안하였고 5절에서는 결론 및 향후 연구과제를 제시하였다.

2. 관련연구

2.1. 신뢰도 성장 모형

2 1 1 Jelinski-Moranda모형

신뢰성 모형 중에서 대표적인 모형은 Jelinski-Moranda (JM)모형이다. 이 모형은 초기 고장 수에 의존하는 모형이며 소프트웨어 공학분야에서 일반화된 소프트웨어 신뢰도 성장 모형으로 알려져 왔고. 소프트웨어 고장 데이터를 기술하는 가장 간단한 모형이다 N을 테스팅 스테이지 초기에의 소프트웨어 오류의 총개수라고 하고, ϕ는 한 오류마다의 오류 발견률을 나타내는 상수라 하면 JM모형의 가정은 다음과 같다.

i) N은 소프트웨어에 존재하는 초기 오류의 총개수이고 미지의 고정된 상수이다.

ii) 고장의 원인은 한 개의 오류에 의해 발생하고, 추가적인 오류는 발생하지 않으며 디 버깅 절차는 완벽(perfect)하다.

iii) 고장이 일생되면 즉시 수정되고 고장발생 간격시간 $t_i = x_i - x_{i-1}$는 각각 독립적으로 평균이 $1/\phi(N-i+1)$인 지수분포를 따른다 즉, T_i에 대한 위험함수는 모든 t_i에 대하여 $h_{T_i}(t_i) = \phi(N-i+1)$이 된다. 따라서 t_i에 대한 확률밀도함수는 다음과 같다.

$$f(t_i) = (N-i+1)\phi e^{-\phi(N-i+1)t}$$

2 1.2 Goel-Okumoto모형

Goel-Okumoto가 제안한 개선된 JM모형(GO모형)이 있다 이 모형은 JM모형과 비슷하지만 하나의 버그가 발견되었을 때 그것을 고치는 데에 확률 $w\,(0 \le w \le 1)$가 있는 불완전한 디버깅 모형(imperfect debugging model)을 제안했다 Goel-Okumoto모형에서의 가정은 JM 모형에서의 가정 i)과 ii)는 같고 i^{th}의 고장 간격 시간에 대한 위험 함수는 모든 t에 대하여 $h_{T_i}(t) = \phi(N-w(i-1))$이 된다. 즉 t_i에 대한 확률밀도함수는 다음과 같다.

$$f(t_i) = (N-w(i-1))\phi e^{-\phi(N-w(i-1))t_i}$$

2.1.3 Schick-Wolverton모형

Schock-Wolverton은 JM모형을 수정한 톱니 연결 위험률 함수(sawtooth concatenated hazard function)를 제안하였다. 즉, 그들은 고장간격시간 $t_1, \cdots, t_n$을 N과 ϕ가 주어졌을 때 t_i가 위험함수 $h_T(t) = \phi(N-i+1)t_i$을 가지는 레일리(Rayleigh)분포를 따르고 조건적으로 독립이라고 가정한다. 즉 t_i에 대한 확률밀도 함수는 다음과 같다.

$$f(t_i) = (N-i+1)\phi t_i e^{-\frac{\phi}{2}(N-i+1)t_i^2}$$

2.1.4 음의 이항분포
(negative binomial distribution)

연속적인 성공확률 p를 가진 베르누이(Bernoulli)시행에서 a번째 까지 관찰될 때까지 필요한 고장의 수를 X라고 하면 다음과 같이 평균이 $a(1-p)/p$인 음의 이항분포를 따른다고 하고 $NB(a, p)$이라고 표시한다.

$$P(X=x) = \binom{x+a-1}{x} p^a(1-p)^x, \; x=0,1,2,\cdots$$

위 식에서 a가 1인 경우는 기하분포(geometric distribution)가 되고 p를 작게하고 충분한 시행을 하다보면 포아송(Poisson)분포를 따른다고 알려져 있다.

본 논문에서는 소프트웨어에 존재하는 초기 오류의 총갯수인 상수 N에 대한 사전분포를 음의 이항분포를 가정하였다. 따라서 다음과 같은 사전분포가 된다[15].

$$P(N) = \binom{N+a-1}{N} p^a(1-p)^N$$

2.2. 모형에 대한 깁스 샘플링

이 장에서는, 깁스 샘플링 알고리즘에서 쓰여질 조건부 분포들을 연구할 것이다 깁스 샘플링은 마코브 체인 몬테 카를로(MCMC, Markov Chain Monte Carlo)기법이다. 이 마코브 체인의 정상성 분포는 우리가 구하고자 하는 사후 분포(posterior distribution)이다 또한 이 독립적인 출발점(independent starting points)을 가진 마코브 체인을 발전시켜서 사후 분포로부터 다중 샘플들을 얻을 수 있다[10].

2.2.1 베이즈 정리

깁스 샘플링 알고리즘에서 쓰여질 조건부 분포들을 구하는 방법은 베이지안 방법을 많이 이용하게 된다. 즉, 소프트웨어 신뢰도 성장 모형 중에서 베이지안 모형들이 미지의 모수에 대하여 소위 사전정보를 결합하여 소프트웨어 고장 데이터를 해석하는 방법으로 제안되어져 왔다. 그러므로 소프트웨어 신뢰도 모형에 대한 베이지안 추정의 근본적인 도구는 베이즈 정리가 이용되며 다음과 같은 관계식으로 표현될 수 있다.

사후분포 = 사전분포 × (우도함수 / 주변분포)
∝ 사전분포 × 우도함수

여기에서 우도 함수(Likelihood Function)는 표본 소프트웨어 테스팅 데이타를 이용한 결합확률 밀도 함수이며, 사전분포(Prior Distribution)는 소프트웨어 신뢰도 측도들에 관한 알려진 또는 가정된 모든 정보를 표현하며, 사후분포는 관찰된 표본 소프트웨어 테스팅 자료를 기반으로 하여 사전분포에 의해서 표현된 사전정보의 변경되고 갱신된 정보를 나타내는 분포 함수이다

2.2.2 깁스 샘플링

깁스 샘플링은 화상공정(image-processing)모형을 연구하는 Geman과 German(1984)의 논문을 통하여 비로소 연구가 시작되었다. 이 방법의 근원은 Metropolis, Rosenbiuth. Rosenbuith, Teller(1953)에 의해 처음으로 연구되다가, 후에 Hastings(1970)에 의해 발전되었다.

깁스 샘플링은 다차원 이상의 적분을 계산하기 위한 것으로서 신경망(Neural network), 전문가 시스템(Expert system)과 같은 척도가 큰(large scale) 복잡한 모형에 적용되어 사용되어 왔다. 본 절에서는 Geman과 German (1984)에서 소개된 깁스 샘플링을 이용하고 최근에는 Gelfand와 Smith(1990)에의해 더욱 구체화되었다. 그리고 Gelfand와 Rubin(1992)은 다중열(multiple sequence)을 이용한 반복적 시뮬레이션 기법을 제시하였다.

깁스 샘플링의 가정은 추출된 표본은 서로 독립이고 충분한 수의 반복이 이루어지면 깁스 알고리즘으로부터 추출된 표본은 안정된 특정분포로 수렴한다.

본 절에서는 다음과 같은 Gelman과 Rubin의 기법을 요약하고자 한다 우선, 과산포분포(over-dispersed distribution)에서 유도된 초기점을 가지는 각각 길이 $2n$인 독립열 $m \geq 2$인 열을 시뮬레이트하고, 초기분포의 의존성을 작게하기 위해 각 열에 대해 전반부 n번 반복을 제외하고 후반부 n번 반복만 고려하게 된다.

확률변수 $U_1, U_2, \cdots, U_p$의 결합분포가 주어졌을 때 각각의 주변확률밀도가 이용가능하다고 가정하고 초기값을 $(u_1^{(0)}, u_2^{(0)}, \cdots, u_p^{(0)})$라고 정의하면 다음과 같은 조건밀도로부터

변량이 추출된다

$$u_1^{(1)} \sim f(U_1 \mid U_2^{(0)}, U_3^{(0)}, \cdots, U_p^{(0)}, D_n),$$
$$u_2^{(1)} \sim f(U_2 \mid U_1^{(1)}, U_3^{(0)}, \cdots, U_p^{(0)}, D_n),$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$u_p^{(1)} \sim f(U_p \mid U_1^{(1)}, U_2^{(1)}, \cdots, U_{p-1}^{(1)}, D_n)$$

각각의 변수들은 최신값들로 갱신되면서 p개의 변량을 발생시킬 수 있고 $2n$번의 반복이후에 다음과 같은 변량을 얻을 수 있다.

$$(U_1^{(2n)}, U_2^{(2n)}, \cdots, U_p^{(2n)})$$

따라서, Geman과 Geman(1984)은 적당한 조건하에서 n이 충분히 크면($n \to \infty$) 다음과 같이 수렴함을 제시하였다.

$$(U_1^{(n)}, \cdots, U_p^{(n)}) \xrightarrow{d} (U_1, U_2, \cdots, U_p)$$

그러므로 후반부 n번 반복(iteration)에 m번 적용(replication)하는 깁스 샘플링은 다음과 같이 mn개를 발생시킨다.

$$(U_1^{(l)}, \cdots, U_p^{(l)}) \; (j = 1, 2, \cdots, m, \; l = n+1, \cdots, 2n)$$

위 식에서 $U_1, \cdots, U_p$는 경우에 따라서 벡터가 될 수 있다.

사후밀도의 값을 계산하기 위해서 본 논문은 다음과 같이 Rao-Blackwell개념을 사용하였다[9].

$$f(\widehat{U_s}) \simeq (mn)^{-1} \sum_{j=1}^{m} \sum_{l=n+1}^{2n} f(U_s \mid U_r^l, \; r \neq s)$$

3. 사전분포가 음이항분포를 가지는 신뢰모형에 대한 깁스샘플링

3.1 Jelinski-Moranda모형

사전분포를 정의함에 있어 $NB(a, p)$는 평균이 $a(1-p)/p$인 음이항분포 그리고 $\Gamma(\alpha, \beta)$는 평균이 α/β인 감마분포를 나타내고 $N \perp \phi$는 N와 ϕ가 독립이라는 것을 의미하고 JM모형은 n^{th}번 고장날 때까지 관찰된 우도함수는 다음과 같다.

$$L(N, \phi; D_{t_n}) = \prod_{i=1}^{n} (N-i+1)\phi e^{-(N-i+1)\phi t_i}$$
$$= \{\prod_{i=1}^{n} (N-i+1)\} \phi^n e^{-\phi T}$$

● 깁스 추출 알고리즘

(i) 사전분포 : $N \sim NB(a, p), \; \phi \sim \Gamma(\alpha, \beta);$
$N \perp \phi$ (3.1)

(ii) N과 ϕ의 결합밀도함수

$$P(N, \phi \mid D_{t_n}) \propto L(N, \phi \mid D_{t_n}) \cdot P(N, \phi)$$
$$\propto L(N, \phi \mid D_{t_n}) \cdot P(N) \cdot P(\phi)$$
$$\propto \{\prod_{i=1}^{n} (N-i+1)\} \phi^n e^{-\phi T} \cdot \binom{N+a-1}{N} p^a (1-p)^N \cdot \frac{\beta^\alpha \phi^{\alpha-1} e^{-\beta\phi}}{\Gamma(\alpha)}$$

(iii) $N' = N - n$에 대한 사후분포

$$P(N' \mid \phi, D_{t_n}) \propto L(N, \phi \mid D_{t_n}) \cdot P(N)$$
$$\propto \binom{N'+n+a+1}{N'} \cdot \left[(1-p)e^{[-\phi \sum_{i=1}^{n} t_i]}\right]^{N'} \cdot \left[1-(1-p)e^{[-\phi \sum_{i=1}^{n} t_i]}\right]^{a+n}$$

그러므로 $N' = N - n$에 대한 사후분포는 다음과 같은 모수를 가지는 음이항분포가 된다

$$N' \mid \phi, D_{t_n} \sim NB\left(a+n, \; 1-(1-p)e^{[-\phi \sum_{i=1}^{n} t_i]}\right) \quad (3.2)$$

(iv) ϕ에 대한 사후분포

ϕ에 대한 사후분포는 베이즈 정리와 장애모수(nuisance parameter)개념을 이용하여 ϕ에 대한 사후분포는 다음과 같이 계산할 수 있다.

$$P(\phi \mid N, D_{t_n}) \propto \{\prod_{i=1}^{n} (N-i+1)\} \phi^n e^{[-\phi \sum_{i=1}^{n} (N-i+1)t_i]} \frac{\beta^\alpha \phi^{\alpha-1} e^{-\beta\phi}}{\Gamma(\alpha)}$$

$$\phi \mid N, D_{t_n} \sim \Gamma(\alpha+n,\ \beta+N'x_n+\sum_{i=1}^{n} x_i) \qquad (3.3)$$

깁스 알고리즘은 (3.1)의 ϕ의 사전분포에서 임의난수를 초기치로 하여 $P(N' \mid \phi, D_{t_n})$에 대입하여 $N'(=N-n)$의 랜덤표본을 얻고 이 값을 $P(\phi \mid N, D_{t_n})$에 대입하여 ϕ의 랜덤표본을 얻는다. 여기서 얻은 ϕ, N'을 새로운 초기치로 하여 다시 반복한다(n번). 이러한 과정을 m번 적용한다. 예를들어 사후평균은 발생된 표본 전체의 개수를 평균하면 되고 보다 빠른 수렴을 위해서는 Rao-Blackwell 개념을 사용하게 된다. 이 개념은 매 반복횟수를 2등분하여 전반부는 버리고 후반부만 사용하여 사후평균을 구하게 된다[9].

3.2 Goel-Okumoto모형

GO모형인 경우도 유사한 방법으로 $T=\sum_{i=1}^{n}(N-w(i-1))t_i$ 이므로, n^{th}번 고장날 때까지 관찰된 우도함수는 다음과 같다.

$$L(N,\phi;D_{t_n}) = \prod_{i=1}^{n}(N-w(i-1))\phi e^{-(N-w(i-1))\phi t_i}$$

$$= \{\prod_{i=1}^{n}(N-w(i-1))\}\phi^n e^{-\sum(N-w(i-1))\phi t_i}$$

- 깁스 추출 알고리즘

(i) 사전분포 $N \sim NB(a,p)$; $\phi \sim \Gamma(\alpha,\beta)$; $N \perp \phi$

(ii) N과 ϕ의 결합밀도함수

$$P(N,\phi \mid D_{t_n}) \propto L(N,\phi \mid D_{t_n}) \cdot P(N,\phi)$$

$$\propto L(N,\phi \mid D_{t_n}) \cdot P(N) \cdot P(\phi)$$

$$\propto \{\prod_{i=1}^{n}(N-w(i-1)\}\phi^n e^{-\phi T} \cdot \binom{N+a-1}{N} p^a(1-p)^N \cdot \frac{\beta^\alpha \phi^{\alpha-1} e^{-\beta\phi}}{\Gamma(\alpha)}$$

(iii) $N''=N-mw$에 대한 사후분포

$$P(N'' \mid \phi, D_{t_n}) \propto L(N,\phi \mid D_{t_n}) \cdot P(N)$$

$$\propto \{\prod_{i=1}^{n}(N-w(i-1)\}\phi^n e^{-\phi T} \cdot \binom{N+a-1}{N} p^a(1-p)^N$$

$$\propto \binom{N''+nw+a-1}{N''} \cdot \left[(1-p)e^{[-\phi\sum_{i=1}^{n} t_i]}\right]^{N'} \cdot \left[1-(1-p)e^{[-\phi\sum_{i=1}^{n} t_i]}\right]^{nw+a-1}$$

그러므로 $N''=N-mw$에 대한 사후분포는 다음과 같은 모수를 가지는 음이항분포가 된다.

$$N'' \mid \phi, D_{t_n} \sim NB\left(nw+a+n,\ 1-(1-p)e^{[-\phi\sum_{i=1}^{n} t_i]}\right)$$

(iv) ϕ에 대한 사후분포

ϕ에 대한 사후분포는 베이즈 정리와 장애모수(nuisance parameter)개념을 이용하여 ϕ에 대한 사후분포는 다음과 같이 계산할 수 있다.

$$P(\phi \mid N, D_{t_n}) \propto \{\prod_{i=1}^{n}(N-w(i-1))\}\phi^n e^{[-\phi\sum_{i=1}^{n}(N-w(i-1))t_i]} \cdot \frac{\beta^\alpha \phi^{\alpha-1} e^{-\beta\phi}}{\Gamma(\alpha)}$$

$$\phi \mid N, D_{t_n} \sim \Gamma(\alpha+n,\ \sum_{i=1}^{n}(N-w(i-1))t_i+\beta)$$

3.3 Schick-Wolverton모형

사전분포를 정의함에 있어 $NB(a,p)$는 평균이 $a(1-p)/p$인 음이항분포 그리고 $\Gamma(\alpha,\beta)$는 평균이 α/β인 감마분포를 나타낸다

SW모형은 앞의 사전분포와 유사하게 n번 실패할 때까지 관찰된 우도함수는 다음과 같다.

$$L(N,\phi,D_{t_n}) = \prod_{i=1}^{n}[(N-i+1)\phi t_i \exp\{-\frac{\phi}{2}(N-i+1)\ t_i^2\}]$$

$$= \{\prod_{i=1}^{n}(N-i+1)\}\phi^n\{\prod_{i=1}^{n} t_i\} \exp\{-\frac{\phi}{2}\sum_{i=1}^{n}(N-i+1)\ t_i^2\}$$

- 깁스 추출 알고리즘

(i) 사전분포 $N \sim NB(a,p)$; $\phi \sim \Gamma(\alpha,\beta)$; $N \perp \phi$

(ii) N과 ϕ의 결합밀도함수

$$P(N,\phi \mid D_{t_n}) \propto L(N,\phi \mid D_{t_n}) \cdot P(N) \cdot P(\phi)$$

$$\propto \{\prod_{i=1}^{n}(N-i+1)\}\phi^n\{\prod_{i=1}^{n} t_i\}$$

$$\exp\{-\frac{\phi}{2}\sum_{i=1}^{n}(N-i+1)\ t_i^2\}\cdot \binom{N+a-1}{N}p^a(1-p)^N\cdot \frac{\beta^a\phi^{a-1}e^{-\beta\phi}}{\Gamma(a)}$$

(iii) $N'=N-n$에 대한 사후분포

$$P(N'\mid\phi,D_{t_n})\propto L(N,\phi\mid D_{t_n})\cdot P(N)$$
$$\propto\{\prod_{i=1}^{n}(N-i+1)\}\phi^n\{\prod_{i=1}^{n}t_i\}\exp\{-\frac{\phi}{2}\sum_{i=1}^{n}(N-i+1)\ t_i^2\}\cdot\binom{N+a-1}{N}p^a(1-p)^N$$

따라서 $N'=N-n$에 대한 사후분포는 다음과 같은 모수를 가지는 음이항분포가 된다 즉,

$$N'\mid\phi,D_{t_n}\sim NB\left(a+n,\ 1-(1-p)e^{\left[-\frac{\phi}{2}\sum t_i^2\right]}\right)$$

(iv) ϕ에 대한 사후분포

ϕ에 대한 사후분포는 베이즈 정리와 장애모수(nuisance parameter)개념을 이용하여 ϕ에 대한 사후분포는 다음과 같이 계산할 수 있다.

$$P(\phi\mid N,D_{t_n})\propto L(N,\phi\mid D_{t_n})\cdot P(N)$$
$$\propto\frac{N!}{(N-n)!}\exp[-\frac{\phi}{2}(\sum_{i=1}^{n}(N-i+1)t_i+2\beta)]\cdot\phi^{n+a-1}\cdot\frac{\beta^a}{\Gamma(a)}$$

그러므로 ϕ에 대한 사후분포는 다음과 같은 모수를 가지는 감마분포가 된다.

$$\phi\mid N,D_{t_n}\sim\Gamma(a+n,\ \frac{1}{2}\{(N-n)\sum_{i=1}^{n}t_i^2+n\sum_{i=1}^{n}t_i^2-\sum_{i=1}^{n}(i-1)\ t_i^2+2\beta\})$$

4. 모형선택

본 논문에서는 모형선택에 있어서 Braun statistic과 중위수 변량을 이용하고자 한다[1].

Braun은 다음과 같은 직관적 비교를 위한 통계량을 제시하였다.

$$B(l)=\frac{\sum_i\{t_i-\widehat{E}(T_i)\}^2}{\sum_i\{t_i-\bar{t}\}^2}\cdot\frac{n-1}{n-2}$$

단, $\widehat{E}(T_i)$는 T_i의 추정된 평균을 의미하고 $B(l)$의 값이 작은 모형일수록 효율적인 모형이 된다 그리고 중위수 변량은 다음과 같이 정의하고 있다.

$$M(v)=\sum_{i=1}^{n}\left|\frac{m_i-m_{i-1}}{m_{i-1}}\right|$$

단, m_i는 T_i의 시점에서 추정된 중위수를 의미하고 $M(v)$의 값이 작은 모형일수록 효율적인 모형이 된다.

참값을 알고있는 특수한 경우에는 상대오차의 합(the sum of relative errors)으로 모형 비교를 할 수 있다. 상대오차의 합은 다음과 같이 정의된다

$$RE(l)=\sum_{i=1}^{n}\left|\frac{\text{참값모형}-\text{추정값모형}}{\text{참값모형}}\right|$$

단, 참값모형은 각각의 모형에 관찰된 값 t_i와 N, ϕ값을 대입했을 때의 값이고 추정값 모형은 각각의 모형에 관찰된 값 t_i와 깁스 알고리즘에 의해 구해진 $\hat{N}$과 $\hat{\phi}$를 대입했을 때의 값이고, l은 인덱스(index)된 모형을 나타내고 $RE(l)$의 값이 작으면 보다 좋은 모형이라고 한다[15].

5. 수치적인 예

t_i에 대한 자료는 SAS 소프트웨어를 이용하여 총 오류갯수인 $N=35$, 오류 발견율 $\phi=0.00045$로 주고 와이블분포의 특수한 형태인 레일리분포에서 랜덤추출하였다

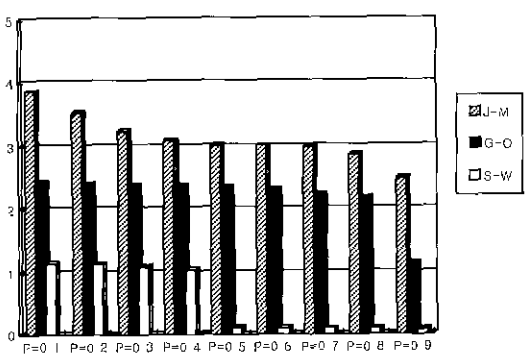
깁스 샘플링을 하는데 있어서의 N에 대한 사전분포를 $N\sim NB(30,p)$로 초기치를 주었으며, ϕ에 대한 사전분포는 디퓨즈(diffuse)한 사전분포 상태를 만들기 위해 분산이 비교적 큰 $\Gamma(1,0.0001)$을 택하였다 깁스 샘플링에서 SAS 소프트웨어를 이용하여 반복은 500번, 반복에 대한 적용은 각각 1000, 2000, 3000번씩 하였다 <표 1>은 레일리 분포에서 추출한 30개의 t_i의 자료값이다. <표 2>는 사전분포가 음이항분포일때 J-M 모형에서 추정된 사후평균의 값이다 이 표에서 음이항분포의 p값은 0.1에서 0.9까지 주고 구하였다. 즉, p값이 커짐에 따라 N추정값은 작아지고 ϕ값은 조금씩 커지고 있음을 알 수 있다. <표 3>은 사전분포가 음이항분포일때 G-O모형에서 $w=0.5$일 때 추정된 값이

다. G-O모형에서도 p값이 커짐에 따라 N 추정값은 작아지고 ϕ 값은 조금씩 커지고 있음을 알 수 있다 <표 4>는 사전분포가 음이항분포일때 S-W모형에서 추정된 값이다. S-W모형 역시 p값이 커짐에 따라 N 추정값은 작아지고 ϕ값은 조금씩 커지고 있음을 알 수 있다 <표 5>는 앞에서 구한 각 모형에 대한 3000번의 적용으로서 N과 ϕ의 추정값으로 부터 구한 상대오차의 합을 구한 것이고 (그림 1)과 (그림 2)는 Braun statistic과 중위수 변량의 합을 이용한 모형선택의 결과에 대한 그림이다

모의실험 결과를 보면 사전분포가 음이항분포일 때 상대오차의 합이 J-M모형 보다는 G-O모형이 상대오차의 합이 작게 나왔고, G-O모형보다는 S-W모형이 상대오차의 합이 작게 나왔고 (그림 1)과 (그림 2)에 그려진 Braun statistic과 중위수 변량의 합을 이용한 모형선택의 결과도 같다. 음이항분포에서 J-M모형과 G-O모형에서는 p가 커질수록 상대오차의 합이 점점 작아지고 있음을 알 수 있었고 S-W모형에서는 p가

〈표 1〉 고장간격시간(t_i)에 대한 모의자료

고장번호 i	t_i	고장번호 i	t_i	고장번호 i	t_i
1	14	11	11	21	11
2	17	22	10	22	38
3	20	13	5	23	14
1	4	14	16	24	2
5	7	15	8	25	6
6	8	16	18	26	20
7	14	17	25	27	18
8	5	18	1	28	14
9	13	19	13	29	35
10	11	20	24	30	17



(그림 1) Braun statistic의 합의 비교

〈표 2〉 p에 따른 J-M모형의 사후평균 추정값

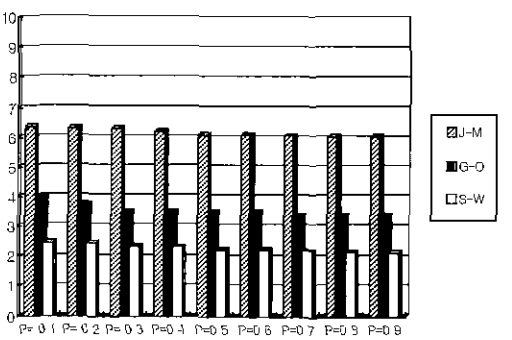
p	적 용	반복	N 추정값	ϕ 추정값
p=0.1	1000	500	263.31297601	0.0003067834
	2000	500	261.59212588	0.0003061762
	3000	500	261.20517697	0.0003081069
p=0.2	1000	500	116.10616298	0.0007609889
	2000	500	115.20964389	0.0007691966
	3000	500	114.83654979	0.0007673552
p=0.3	1000	500	67.882820521	0.0014579936
	2000	500	68.225854528	0.0014592963
	3000	500	67.798739935	0.0014810973
p=0.4	1000	500	48.362685953	0.0023604762
	2000	500	47.884462671	0.0024056643
	3000	500	47.807354535	0.0024012637
p=0.5	1000	500	39.706937585	0.0031842159
	2000	500	39.569539953	0.0032126848
	3000	500	39.630036389	0.0032075738
p=0.6	1000	500	35.570400818	0.0038861427
	2000	500	35.641937155	0.0038193514
	3000	500	35.553415112	0.0038573429
p=0.7	1000	500	33.328499299	0.0043369059
	2000	500	33.263476065	0.0043272423
	3000	500	33.287665944	0.0043384104
p=0.8	1000	500	31.813136192	0.0047456359
	2000	500	31.835193434	0.0047222608
	3000	500	31.820248642	0.0047264133
p=0.9	1000	500	30.780521243	0.0050319813
	2000	500	30.781718651	0.0050370761
	3000	500	30.793881312	0.0050428152

〈표 3〉 p에 따른 G-O모형 (w=0.5)의 모수 추정값(사후평균)

p	적 용	반복	N 추정값	ϕ 추정값
p=0.1	1000	500	389.29427 063	0.000187794 2
	2000	500	386.96209867	0.0001895586
	3000	500	387.52226858	0.0001896855
p=0.2	1000	500	167.82589334	0.0004260022
	2000	500	167.83371192	0.0004270822
	3000	500	167.70998961	0.0004279451
p=0.3	1000	500	96.3560843 49	0.0007247244
	2000	500	96.080822205	0.0007255182
	3000	500	96.14261788	0.0007345912
p=0.4	1000	500	62.060760197	0.0010994277
	2000	500	61.754627979	0.0010917608
	3000	500	61.691898438	0.0010934637
p=0.5	1000	500	42.742307514	0.001501 4836
	2000	500	42.489275454	0.001517731
	3000	500	42.663020286	0.0015112972
p=0.6	1000	500	31.726056233	0.0019415845
	2000	500	31.578698677	0.0019350302
	3000	500	31.626210178	0.0019181157
p=0.7	1000	500	24.82870192	0.0023349951
	2000	500	24.760834563	0.0023487684
	3000	500	24.78222836	0.0023426148
p=0.8	1000	500	20.276371424	0.0027365627
	2000	500	20.168357831	0.0027457337
	3000	500	20.302125462	0.0027137186
p=0.9	1000	500	17.207929952	0.0030508907
	2000	500	17.217400268	0.0030607973
	3000	500	17.189582977	0.0030642072

〈표 4〉 p에 따른 S-W모형의 모수 추정값(사후평균)

p	적 용	반 복	N 추정값	ϕ 추정값
p=0.1	1000	500	261 47468899	0.0000324143
	2000	500	260 26878961	0.0000324075
	3000	500	259 72143986	0 0000326037
p=0.2	1000	500	111.04519916	0.0000855745
	2000	500	111 04570292	0.0000853196
	3000	500	110.92546016	0 0000863 14
p=0.3	1000	3500	61.513645759	0.0001882643
	2000	500	61 722046888	0.000186689
	3000	500	61 905413146	0.0001857711
p=0 4	1000	500	42.786810022	0.0003311261
	2000	500	42.785455743	0.0003259567
	3000	500	42.723380952	0 0003279534
p=0.5	1000	500	36.455667266	0.0004309503
	2000	500	36.418766676	0 0004298692
	3000	500	36 410538468	0.0004301192
p=0 6	1000	500	33 826840922	0 0004978141
	2000	500	33.702545176	0 0005026752
	3000	500	33 747075441	0 0005021933
p=0 7	1000	500	32.245079113	0 0005461543
	2000	500	32 278220197	0 0005498607
	3000	500	32 242531818	0 0005526431
p=0.8	1000	500	31 268694299	0 0005941504
	2000	500	31.264647859	0 0005885977
	3000	500	31.255710374	0.0005869103
p=0 9	1000	500	30.549564746	0.0006257209
	2000	500	30 550527959	0.0006255494
	3000	500	30.543895795	0.0006211245



(그림 2) 중위수 변량의 합의 비교

0.5와 0.6일 때 상대오차의 합이 가장 작음을 알 수 있었다 따라서 모형선택에 있어서는 J-M모형보다는 G-O 모형이, G-O모형보다는 S-W모형이 소프트웨어 신뢰 모형에서 더 적합하다고 결론 지을 수 있다.

5. 결론 및 향후 연구과제

소프트웨어의 신뢰성은 개발의 최종단계에 있는 테스트 공정이나 실제 사용 단계에 있어서 소프트웨어 내에 존재하는 에러 수나 소프트웨어의 고장발생시간에 의해 효과적 평가할 수 있는 것으로 그 평가 기술이 중요하게 된다 소프트웨어 개발의 테스트 공정이나 실제 사용단계에 있어서 에러 발생상황이나 소프트웨어 고장 발생현상을 수리적 모델화가 가능하다면 평가를 할 수 있다. 테스트의 개발 상황의 파악, 테스트에 의해 미발생 되었던 에러에 대한 보수 코스트의 예측 등 구체적인 소프트웨어 개발의 보수관리문제에도 적용 가능하다 따라서 테스트 시간 혹은 실행시간과 발생된 에러 수나 소프트웨어 고장의 발생시간과의 관계를 소프트웨어 신뢰도 성장과정이라고 볼 수 있다.

테스트 공정 혹은 실행 단계에 있어서 소프트웨어가 컴퓨터 상에서 실행될 때 어떤 입력 데이터에 대해서는 잠재하는 에러에 의해 소프트웨어 고장을 일으킨다. 이와 같은 소프트웨어 고장의 발생현상 혹은 에러의 발생사상은 불확정 사상이며 이 중에서 소프트웨어의 신뢰성 현상을 발견하여 수량화하기 위해서는 수리적으로 취급하여야 한다. 그 이유는 소프트웨어의 고장의 발생시간을 추정하기 때문이다 일반적으로 된 에러 수나 소프트웨어 고장 시간에 대한 데이터를 사용해서 소프트웨어의 신뢰성을 평가하는 수리모델은 해석적 모델(analytical model)로 분류된다. 특히 소프트웨어 개발의 테스트 공정에서는 공정 수 등의 대량의 테스트 자원을 투입하여 에러의 발생과 수정이 이루어지는 것으로 소프트웨어 내에 잠재하는 에러 수는 테스트 시간의 경과와 함께 감소한다. 여기에서 발생

〈표 5〉 상대오차의 합

모형	n	p=0.1	p=0 2	p=0.3	p=0.4	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9
		합	합	합	합	합	합	합	합	합
IM	30	0.53564	0.54021	0.54571	0.5432	0 53498	0 52439	0 51299	0 50214	0 4918
G-O	30	0.42328	0.40940	0 39089	0.3672	0 33363	0 28868	0 24195	0 19337	0 1481
S-W	30	0 14169	0 10741	0 05674	0 0141	0 00129	0 00202	0 00850	0 01952	0 0979

된 에러는 모두 수정 혹은 제거되고 수정 시에 새로운 에러는 투입 않는 것으로 가정하고 있다. 실제로 소프트웨어의 신뢰성을 모델 화하는데 있어서는 이 가정은 완화될 수 있다. 따라서 테스트 시간의 경과와 함께 소프트웨어 고장이 발생할 확률은 감소하고 소프트웨어의 신뢰도나 소프트웨어 고장의 발생 시간 간격이 증가하는 추세를 보인다. 현실적 환경을 반영한 소프트웨어의 신뢰성 평가를 실시하기 위해서는 비현실적인 모델적용의 가정은 가능한 한 완화하게 할 필요가 있다. 하드웨어 제품의 신뢰성 특성과 같은 방법으로 소프트웨어에 대해서도 발생된 에러 수나 소프트웨어 고장의 발생시간을 수량화하여 확률 변수로 생각할 수 있다

일반적인 추정 문제에서 적분 계산이 복잡하여 많은 시간과 노력을 필요로 하는 경우를 직면하게 된다 이러한 점 때문에 복잡한 계산을 보다 쉽게 해결하려는 연구가 다양하게 이루어져 왔다. 이러한 문제는 Gelman과 Rubin[10]에 의해 구체화된 깁스 알고리즘이 소개되었고 이 기법은 적분대신 적절한 조건부분포로부터 반복표본을 이용한 몬데칼로 적분으로 쉽게 추정할 수 있게 되었다. 본 논문에서는 소프트웨어 고장현상을 수리적으로 모형화를 하기 위한 소프트웨어 신뢰성장 모형에 이 기법을 적용하여 모형선택에 모수추정이 사용되었다

수치적인 예에서는 레일리분포에서 랜덤 추출한 모의자료를 Braun Statistic 과 중위수 편랑의 합, 상대오차의 합을 이용하여 모형 선택한 결과 J-M모형보다는 G-O모형이, G-O모형보다는 S-W모형이 소프트웨어 신뢰모형에서 더 적합하다고 결론 지을 수 있다

본 논문은 수명분포가 감마족인 경우에 국한되었다. 앞으로 비감마족인 (예를 들어 정규분포, 로그로말, 파레토 등) 경우에 대한 연구가 시행되고 사전분포가 (음이항분포가 아닌) 비공액(nonconjugate)분포인 경우에 대한 연구가 필요하고 신뢰도 성장모형에 대한 분포 이론 및 응용에 관한 연구가 기대된다

참 고 문 헌

[1] Abdel-Ghaly, A., A and Chan, P, Y and Littlewood, B., (1986), "Evaluation of Competing Software Reliability Predictions," *IEEE Transactions on Software Engineering*, 9, pp.950-967.

[2] Berger, J. O. and Sun, D., "Bayesian Analysis For The Poly-Weibull Distribution," *Journal of the American Statistical Association*, 88, pp.1412-1418, 1993.

[3] Box, G., "Sampling and Bayes' Inference in Scientific Modeling and Robustness (with discussion)," *Journal of the Royal Statistical Society, Ser. A*, 143, pp.382-430, 1980

[4] Casella, G. and George, E. I., "Explaining the Gibbs Sampler," *The American Statistician*, 46, pp.167-174, 1992.

[5] Cinlar, E, '*Introduction To Stochastic Process*,' New Jersey: Prentice-Hall, 1975.

[6] Cox, D R and Lewis, P A., 'Statistical Analysis of Series of Events,' Londen: Methuen, 1966.

[7] Dawid, A P., "Statistical Theory: The Prequential Approach." *Journal of the Royal Statistical Society, Ser* A, 147, pp.278-292, 1984.

[8] Geisser, S, and Eddy, W., "A Predictive Approach to Model Selection," *Journal of the American Statistical Association*, 74, pp.153-160, 1979.

[9] Gelfand, A E. and Smith, A. F. M., "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, 85, pp.398-409, 1990

[10] Gelman, A. E., and Rubin D., "Inference from Iterative Simulation Using Multiple Sequences," *Statistical Science*, 7, pp.457-472. 1992/

[11] Goel, A L and Okumoto, K., "An analysis of recurrent software failures on a real-time control systems," Proceedings of the ACM Annual Technical Conference, ACM' Washing D. C. pp.496-500, 1978

[12] Greenberg, E. and Chib, S, "Understanding the Metropolis-Hastings Agorithm," *The American Statistician*,, 49, pp 327-335, 1995

[13] Jelinski, Z, and Moranda, P B., "*Software Reliability Research*, in Statistical Computer Performance Evaluation," ed W. Freiberger, New York: Academic Press, pp.465-497, 1972.

[14] Joe, H, "Statistical Inference for General Order Statistics and Nonhogeneous Poisson process Software Reliability Models," IEEE Transactions

on Software Engineering, 15. pp.1485-1490,1989

[15] Kuo, L, and Yang, T Y., "Bayesian Computation of Software Reliability," *Journal of Computational and Graphical Statistics*, pp 65-82, 1995.

[16] Kuo, L., and Yang, T Y., "Bayesian Computation for Nonhomogeneous Poisson process in Software Reliability," *Journal of the American Statistical Association*, 91, pp.763-773, 1996.

[17] Langberg, N., and Singpurwalla, N. D., A Unification of Some Software Reliability Models, *SIAM Journal on Scientific and Staistical Computing*, 6, pp.781-790, 1985.

[18] Lawless, J. F., "*Statistical Models and Methods for lifetime Data*," New York: John Wiley & Sons, 1982.

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

[20] Musa, J D, and Okumoto. K., A, "Logarithmic Poisson Execution Time Model for Software Reliability Measurement." *in Proceedings Seventh International Conference on Software Engineering Orlando*, pp.230-238, 1984.

[21] Parzen, E., '*Stochastic Process*,' San Francisco: Holden-Day, 1962.

[22] Raftery, A E., "Inference and Prediction for a General Order Statistic Model with Unknown Population Size," *Journal of the American Statistical Association*, 92, pp 1195-1212, 1997.

[23] Pesnick, S. I., 'Extreme Values. Regular Variation, and Point *Process*,' *Berlin*: Springer-Verlag, 1987.

[24] Schick, G. J and Wolverton, R. W., "An Analysis of Competing Software Reliability Models," IEEE Transactions on Software Engineering, SE-4, 2, pp.104-120, 1978.

[25] Shiha, D. and Day, D. K., "Semiparametric Bayesian Analysis of Survival Data," *Journal of the American Statistical Association*, 81, pp.82-86, 1987.

[26] Tanner, M. and Wong. W, "The Calculation of Posterior Distributions by Data Augmentation (with discussion)," *Journal of the American Statistical Association*, 81, pp.82-86, 1987.

[27] "*USER'S MANUAL STAT/LIBRARY FORTRAN Subroutines for statistical analysis*," IMSL. Vol 3, pp.1050-1054, 1987

김 희 철

e-mail : khc@songho.ac.kr

1997년 동국대학교 대학원 통계학과 졸업(이학박사)

2000년~현재 송호대학 정보산업계열 전임강사

관심분야 : 소프트웨어신뢰성공학, 컴퓨터정보처리, 전산통계

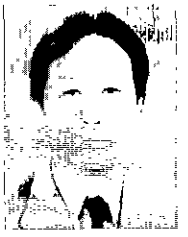
박 종 구

e-mail : parkjg@wonbon.wongkang.ac.kr

1999년 동국대학교 대학원 통계학과 졸업(이학박사)

1981년~현재 원광대학교 컴퓨터공학과 교수

관심분야 소프트웨어신뢰성공학, 전문가시스템, 시스템프로그래밍



이 병 수

e-mail : bsl@hon.inchon.ac.kr

1999년 경기대학교 대학원 전자계산학과 졸업(이학박사)

1981년~현재 시립인천대학교 컴퓨터정보통신학부 교수

관심분야 : 소프트웨어공학, 의사결정지원시스템, IT