韓國컴퓨터情報學會 論文誌
第14卷 第7號, 2009. 7.

2009-14-7-1-1

포아송 실행시간 모형에 의존한 소프트웨어 최적방출시기에 대한 베이지안 접근 방법에 대한 연구

김 희 철*, 신 현 철**

The Bayesian Approach of Software Optimal Release Time Based on Log Poisson Execution Time Model

Kim Hee Cheul *, Hyun-Cheul Shin **

요 약

본 연구에서는 소프트웨어 제품을 개발하여 테스팅을 거친 후 사용자에게 인도하는 시기를 결정하는 방출문제에 대하여 연구하였다. 따라서 최적 소프트웨어 방출 정책은 소프트웨어 요구 신뢰도를 만족시키고 소프트웨어 개발 및 유지 총비용을 최소화 시키는 정책을 수용해야 한다. 본 논문에서는 로그포아송 실행시간모형에 대하여 베이지안 모수 추정법(마코브체인 몬테칼로(MCMC) 기법 중에 하나인 깁스샘플링과 메트로폴리스 알고리즘을 이용한 근사기법)이 사용되었다. 본 논문의 수치적인 예에서는 Musa의 T1 자료를 적용하여 최우추정법과 베이지안 모수 추정과의 관계를 비교하고 또한 최적 방출시기를 추정하였다.

Abstract

In this paper, make a study decision problem called an optimal release policies after testing a software system in development phase and transfer it to the user. The optimal software release policies which minimize a total average software cost of development and maintenance under the constraint of satisfying a software reliability requirement is generally accepted. The Bayesian parametric inference of model using log Poisson execution time employ tool of Markov chain(Gibbs sampling and Metropolis algorithm). In a numerical example by T1 data was illustrated, make out estimating software optimal release time from the maximum likelihood estimation and Bayesian parametric estimation.

▸ Keyword : 로그 포아송 실행 시간모형(Log Poisson Execution Time Model), 베이지안 모수추정(Bayesian Parametric Estimation), 최적소프트웨어 방출정책(Optimal Software Release Policies)

• 제1저자 : 김희철
• 투고일 : 2009. 06. 18, 심사일 : 2009. 07. 14, 게재확정일 : 2009. 07. 18.
* 남서울대학교 산업경영공학과 ** 백석문화대학 컴퓨터정보학부

Ⅰ. 서 론

소프트웨어 신뢰성은 컴퓨터 시스템에 대한 적용과 이에 대한 연구 분야에서 중요한 역할을 담당해 오고 있다. 소프트웨어 고장으로 인한 컴퓨터 시스템의 고장은 우리 사회에 엄청난 손실을 유발 할 수 도 있다. 따라서 소프트웨어 신뢰도는 현대의 소프트웨어 생산품 개발에서 중요한 분야 가운데 하나이다.

소프트웨어 신뢰성 엔지니어링에서의 연구 활동은 지난 30년 전부터 행해져 오고 있고 많은 신뢰도 성장 모형들이 소프트웨어에 남아 있는 고장들의 수와 소프트웨어 신뢰도의 추정에 관한 문제들을 제안해 왔다. 일반적으로 소프트웨어 개발과정은 설계단계, 디자인, 코딩 그리고 테스팅 단계를 거친다. 이러한 과정을 거친 후 소프트웨어 제품을 방출하게 되는데 방출이후에 발견되지 않은 고장들이 나타난다면 이것들에 대한 보전 비용(Maintenance cost)은 크게 증가 할 것이다. 결국, 소프트웨어 시스템 시험을 끝내고 그것을 사용자에게 넘기는 시기 결정은 매우 중요한 사항이 된다.

이러한 소프트웨어 방출시간에 대한 연구는 대부분 NHPP(Non-Homogeneous Poisson Process)모형을 사용하였다[1, 2, 3]. 이 분야에서는 Musa-Okumoto의 로그 포아송 실행시간 모형[4, 5, 6] 과 로그-파우어 모형[7]을 이용한 방출 문제에 대한 문제 들이 이미 연구되었고 최근에도 이와 관련된 문제에 대한 연구는 Yang 과 Xie(2000) 와 Huang(2005)에 의해 연구되고 있다[8, 9]. 그러나 본 연구에서는 대수 포아송 실행시간 모형에 대하여 베이지안적 모수 추정을 적용한 최적 방출시기에 관한 문제를 다루었다. 본 논문의 2절에서는 관련 연구로서 NHPP와 베이지안 추정법, 로그 포아송 실행시간모형에서의 최우추정법 및 베이지안 모수추정에 대하여 약술하였고 3장에서는 요구 신뢰도와 비용 최소화를 고려한 방출시간에 대하여 서술하고 4장과 5장에서는 각각 수치적인 예와 그 결론을 나열 하였다.

Ⅱ. 관련 연구

1. NHPP

NHPP(Non-Homogeneous Poisson Process)모형에서 평균값 함수 $m(t)$ (Mean value function)와 강도 함수(Intensity function) $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다[2, 11].

$$m(t)=\int_0^t \lambda(s)ds \ , \frac{dm(t)}{dt}=\lambda(t) \quad \cdots\cdots (1)$$

따라서 $N(t)$ 는 모수 $m(t)$ 을 가진 포아송 확률 밀도 함수(Probability density function; Pdf)로 알려져 있다. 즉,

$$P\{N(t)=n\}=\frac{[m(t)]^n \cdot e^{-m(t)}}{n!},\ n=0,1,2,\ldots\infty \quad \cdots\cdots (2)$$

이처럼 시간 관련 모형(Time domain models) 들은 NHPP에 의해서 확률 고장 과정으로 설명이 가능하다. 이러한 모형들은 고장 강도 함수 $\lambda(t)$ 가 다르게 표현됨으로서 평균값 함수 $m(t)$ 도 역시 다르게 나타나고 $\lambda(t)$가 상수($m(t)$이면 선형(Linear) 추세)이면 동질적 포아송 과정(HPP)이고, t 에 대한 함수형태이면 NHPP가 된다. 이러한 NHPP 모형들은 유한 고장 모형과 무한 고장 범주로 분류한다[11].

이러한 모형에 대한 우도함수는 다음과 같이 알려져 있다. 즉, 관측시간 $(0,t)$사이에서 n 번째까지 고장시점이 관찰된 고장 절단 모형일 경우($x_n=t$)를 x_i $(i=1,2,\cdots,n)$로 나타내면 자료 집합 D_t 은 $\{x_1,x_2\ \cdots\ x_n;t\}$ 으로 구성되고 x_1에서부터 x_n까지의 자료가 주어졌을 때 NHPP의 우도함수는 다음과 같이 됨이 알려져 있다[1, 2, 12].

$$L_{NHPP}=[\prod_{i=1}^{n}\lambda(x_i)]\exp[-m(x_n)] \quad \cdots\cdots (3)$$

단, $t=x_n$(최종 고장 시점)이면 시간절단 모형(Time truncated model)이 된다. 본 논문에서는 이러한 시간절단 모형을 이용하고자 한다.

2. 베이지안 추정법

신뢰도의 정량적인 값을 얻는데 있어 일반적인 방법은 최우추정법(MLE)을 많이 사용하였다. 그러나 새로운 자료가 얻어지면 그 자료를 이제까지 얻었던 자료와 결합시켜 새로운 결론에 도달하려는 이론이 베이즈 추정법이다. 즉, 베이즈 이론은 알려져 있는 사실에 대한 주관적 의견을 경험이나 지식을 바탕으로 하여 사전정보를 만든 다음 실험을 통하여 얻어

진 자료와 결합시켜 사후정보를 추출하는 과정이다. 그러나 베이즈 추정법에서 사전확률 분포인 수명분포가 복잡하면 적분이 불가능해 지기 때문에 사후정보의 추출이 불가능해진다. 따라서 본 연구에서는 최우추정법과 적분이 난해한 경우에 깁스 샘플링(Gibbs sampling)을 이용하여 근사적 깁스 추정량을 유도하여 비교하고 그 특징이 분석되었다[12]. 본 논문에서는 대수형 포아송 모형(Logarithmic Poisson)을 적용하여 모수를 추정하고 베이즈 추정치들을 계산하기 위한 사후분포의 형태를 구하는데 메트로폴리스(Metropolis) 알고리즘을 사용하였다. 사후 분포가 임의의 분포를 따르지 않을 경우는 메트로폴리스 알고리즘을 사용할 수 있는데 이 기법은 깁스 알고리즘의 대안으로 이용된다[12, 13, 14].

깁스 알고리즘은 MCMC(Markov Chain Monte Carlo) 알고리즘이다. 이는 바람직한 사후 분포로서 정상성 분포를 가지는 마코프 체인에 따라 표본들이 변화하는 알고리즘이다[12, 13]

이 마코프 체인의 추이 측정은 주로 보통 조건부 밀도함수의 곱이다. 즉, 한 변수는 소프트웨어에 남아있는 수많은 오류들이며, 다른 변수는 포아송 확률들 중에서 척도모수(Scale parameter)의 변화에 따라서 어떤 요소가 도움을 주는지를 나타내는 것이다.

3. 로그 포아송 실행시간모형

3.1 최우추정법

로그 포아송 실행시간(Log Poisson execution time ; PET)모형[4.5,6]은 1984년에 Musa 와 Okumoto에 의해서 소개된 무한 고장 소프트웨어 모형으로 평균값함수와 강도함수는 다음과 같이 알려져 있다.

$$m(t) = \frac{1}{\theta} ln(\lambda_0 \theta t + 1) \quad \cdots\cdots (4)$$

$$\lambda(t) = \frac{\lambda_0}{\lambda_0 \theta t + 1} \quad \cdots\cdots (5)$$

한편, (4)식과 (5)식을 (3)식에 대입하면 우도함수는 다음과 같이 유도 할 수 있다.

$$L(\theta, \lambda_0 \mid \underline{x}) = \left(\prod_{i=1}^{n} \frac{\lambda_0}{\lambda_0 \theta x_i + 1} \right) \cdot e^{-\frac{1}{\theta} ln(\lambda_0 \theta x_n + 1)} \quad \cdots\cdots (6)$$

단, $\underline{x} = (x_1, x_2, x_3, \cdots x_n]$.

최우추정법(MLE)을 이용하기 위한 로그 우도 함수는 다음과 같이 유도된다.

$$\ln L(\theta, \lambda_0 \mid \underline{x} = n \ln \lambda_0 - \ln \sum_{i=1}^{n} (\lambda_0 \theta x_i + 1) - \frac{1}{\theta} ln(\lambda_0 \theta x_n + 1) \quad \cdots\cdots (7)$$

(7)식을 이용하여 최우추정치 $\hat{\theta}_{MLE}$ 와 $\widehat{\lambda_0}_{MLE}$ 은 다음과 같이 구할 수 있다고 하였다[5, 6].

$$\hat{\theta}_{MLE} = \frac{1}{n} ln(\hat{\phi} \ x_n + 1) \quad \cdots\cdots (8)$$

$$\widehat{\lambda_0}_{MLE} = \hat{\phi} / \widehat{\theta_{MLE}} \quad \cdots\cdots (9)$$

단, $\phi \left(= \widehat{\lambda_{0_{MLE}}} \cdot \widehat{\theta_{MLE}} \right)$는 (10) 식의 근이 된다.

즉, 이 ϕ 근을 구하기 위해서는 수치 해석적 방법으로 다음과 같은 식을 이용하여 계산할 수 있다[5].

$$\frac{\partial \ln L(\phi \mid \underline{x})}{\partial \phi} = \frac{n}{\phi} - \sum_{i=1}^{n} \frac{x_i}{\phi x_i + 1} - \frac{n \ x_n}{(\phi \ x_n + 1) \ln(\phi x_n + 1)} = 0 \quad \cdots\cdots (10)$$

3.2 베이지안 추정법

대수형 포아송 실행 모형에 대한 평균 값 함수와 강도함수를 이용하면 다음과 같은 우도함수를 나타낼 수 있다. 변수변환을 위하여 (6)식에서 $\beta_0 = 1/\theta$와 $\beta_1 = \lambda_0 \theta$으로 변수변환하면 다음과 같이 유도 된다[12].

$$L_{NHPP}(\beta_0, \beta_1 \mid D_t) = \prod_{i=1}^{n} \left(\frac{\beta_0 \beta_1}{1 + \beta_1 x_i} \right) \cdot \exp[-\beta_0 \ln(1 + \beta_1 x_n)] \quad \cdots\cdots (11)$$

이 모형에 대한 사후밀도 함수를 관찰하기 위해서는 베이즈 정리를 이용하면 다음과 같이 나타낼 수 있다[11].

$$f_{NHPP}(\beta_0,\beta_1 \mid D_t) \propto \prod_{i=1}^{n}\left(\frac{\beta_0\beta_1}{1+\beta_1 x_i}\right) \cdot \exp[-\beta_0 \ln(1+\beta_1 x_n)] \cdot \pi_0(\beta_0) \cdot \pi_0(\beta_1) \quad \cdots (12)$$

단, $\pi_0(\beta_0)$는 평균이 $\frac{a_1}{b_1}$ $(a_1 > 0, b_1 > 0)$인 감마분포 $\Gamma(a_1, b_1)$을 따르는 β_0의 사전밀도(분포)를 나타내고 $\pi_0(\beta_1)$는 $\frac{1}{\beta_1}$ $(0 < \beta_1 < 1)$을 따르는 β_1의 비정보사전밀도(noninformative prior density)를 의미한다.

식 (12)을 이용하여 모수 추정을 하기 위하여 깁스 추출법을 사용하고자 하는 조건부 사후 분포는 다음과 같이 유도 된다[12].

$$\beta_0 \mid \beta_1, D_t \sim \Gamma(n+a_1,\, b_1+\ln(1+\beta_1 x_n)) \quad \cdots (13)$$

$$\beta_1 \mid \beta_0, D_t \propto \beta_1^{n-1} \frac{1}{\prod_{i=1}^{n}(1+\beta_1 x_i)} \cdot \exp[-\beta_0 \ln(1+\beta_1 x_n)] \cdot \pi_0(\beta_1) \quad \cdots (14)$$

• 깁스 알고리즘 시행 단계[11, 12, 13]

위 식 (13)과 식 (14)을 이용하여 다음과 같은 단계를 이용한다. 본 논문에서는 β_0에 대한 사전분포는 비교적 넓은 범위에서 표본 발생이 이루어지도록 분산이 큰 감마분포 $\Gamma(1, 0.001)$를 주고 β_1에 대한 사전분포는 임의의 상수 예를 들면 $0 < \beta_1 < 1$를 만족하는 임의의 상수를 초기치로 준다.

① 1-1 단계

$a_1 = 1,\quad b_1 = 0.001,\quad \beta_1 = \frac{1}{0.001}$ 자료를 식(13)에 대입하여 랜덤 표본 $\beta_0^{(1)}$을 얻는다.

$$\beta_0^{(1)} \sim \Gamma(n+a_1,\ b_1+\ln(1+\beta_1 x_n))$$

② 1-2 단계 : 메트로폴리스 알고리즘

설명을 간결하게 하기 위해서 식 (14)의 오른쪽 식, 즉 목적 분포를 $f(\beta_1)$로 표시하자.

이 식에서 β_0는 (1-1) 단계에서 얻은 값을 대입하고 β_1는 임의의 상수 $0 < \beta_1 < 1$을 만족하는 값을 대입하고 추이 커널(Transitional kernel)은 거의 대칭을 이루는 감마분포 $\Gamma(1, 0.001)$에서 β_1'을 랜덤 표본발생하고 균등분포(0, 1]에서 임의 확률변량을 w라 하면 $\log w \le \log f(\beta_1') - \log(\beta_1)$을 만족하면 β_1'을 $\beta_1^{(1)}$으로 간주되고 만족하지 않으면 $\beta_1^{(1)}$은 다시 β_1'으로 대치되면서 계속 반복된다.

③ 2 단계

1-1 단계와 1-2 단계를 시행하여 최근에 생성된 랜덤 표본의 값들로 대체하면서 충분히 큰 수 t만큼 반복한다. 이렇게 하여 얻은 표본을 $(\beta_0^{(t)}, \beta_1^{(t)})$하자.

④ 3 단계

1-1 단계와 1-2 단계를 다시 (m-1)번 충분히 반복적용하면 m개의 랜덤 표본을 다음과 같이 얻어진다.

$(\beta_{0(1)}^{(t)}, \beta_{1(1)}^{(t)}),\ (\beta_{0(2)}^{(t)}, \beta_{1(2)}^{(t)}),\ \cdots,\ (\beta_{0(m)}^{(t)}, \beta_{1(m)}^{(t)})$

⑤ 4 단계

최종적인 결과에 의해 β_0와 β_1의 추정은 다음과 같다.

$$\hat{\beta}_{0\,Gibbs} = \frac{1}{m}\sum_{l=1}^{m}\beta_{0(l)}^{(t)},\ \hat{\beta}_{1\,Gibbs} = \frac{1}{m}\sum_{l=1}^{m}\beta_{1(l)}^{(t)}$$

III. 요구 신뢰도 및 비용최소화를 고려한 방출시간

NHPP 모형에서 테스트 시점 x_n(마지막 고장시점)에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간 (x_n, x_n+x) (단, x는 임무시간(Mission time))동안 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도 $\hat{R}(x \mid x_n)$는 다음과 같이 됨이 알려져 있다[1, 10].

$$\hat{R}(x \mid x_n) = \exp\left(-\int_{x_n}^{x_n+x}\lambda(\tau)d\tau\right) = \exp[-\{m(x+x_n)-m(x_n)\}] \quad \cdots (15)$$

따라서 로그 포아송 실행시간모형에 대한 신뢰도는 평균값함수 (15)식과 $t = x_n$을 이용하면 다음과 같이 표현된다.

$$R(x \mid T_R) = \exp\left(-\frac{1}{\theta}\left[\ln(\lambda_0\theta(x+t)+1) - \ln(\lambda_0\theta t+1)\right]\right) \quad \cdots (16)$$

따라서 소프트웨어 방출시간 T_R이 신뢰도 $R_0 = \hat{R}(x \mid T_R)$

을 확보해야 한다면 다음 방정식을 만족해야 한다.

$$\ln R_0 = -\frac{1}{\theta}\left[\ln(\lambda_0\theta(x+T_R)+1) - \ln(\lambda_0\theta T_R+1)\right] \quad \cdots (17)$$

단, 신뢰도 $R_0 = \hat{R}(x \mid T_R)$.

비용 최소화와 관련된 최적 방출시간은 신뢰도와 함께 비용모형에 의해서 결정된다. 소프트웨어 방출시간을 T로 표현하고 $m(T)$와 $m(\infty)$을 각 각 $(0, T]$와 $(0, \infty)$의 기간에 발견된 기대 고장수라고 표현하고 $C(T)$을 소프트웨어 라이프사이클(life cycle) 동안에 기대되는 소프트웨어 비용이라고 하면 $C(T)$는 다음과 같이 표현된다[3,5].

$$C(T) = c_1 m(T) + c_2[m(\infty) - m(T)] + c_3 T \quad \cdots (18)$$

위 식에서 c_1는 테스팅 동안에 하나의 고장을 수리하는 비용이고 c_2 가동 중에 하나의 고장을 수리하는 비용($c_2 > c_1$), 그리고 c_3는 단위 시간당 테스팅 비용을 나타낸다. 이와 관련하여 총비용의 최소화는 무한고장 평균값 함수를 가진 NHPP 모형에 대하여 발생 할 수 있다. 무한 수명에 대한 비용함수 $C(T)$식인 (18)식에서 $m(\infty)$은 직접 추정 할 수 없기 때문에 이 식을 사용하기 위해서는 소프트웨어 수명시간인 T_{LC}을 지정하여 분석한다[6]. 이러한 T_{LC}는 소프트웨어마다 서로 다른 임의의 값이기 때문에 유한 고장 NHPP 모형이라고 할 수 는 없다.

따라서 비용함수를 고려하여 소프트웨어의 모든 수명에서 총비용을 최소화함으로서 최적 테스팅 시간을 결정 할 수 있고 다음과 같은 식을 만족하면 비용함수 $C(T)$는 유일한 최소값을 가진다[3,5,6].

$$\frac{dC(T)}{dT} = 0, \quad \frac{d^2 C(T)}{d^2 T} > 0 \quad \cdots (19)$$

결국 소프트웨어 지정된 수명T_{LC}을 이용한 로그 포아송 실행시간모형 비용함수 $C(T)$는 다음과 같이 유도된다.

$$\begin{aligned} C(T) &= c_1 m(T) + c_2[m(\infty) - m(T)] + c_3 T \quad \cdots (20) \\ &= (c_1 - c_2)\frac{1}{\theta} ln(\lambda_0\theta T + 1) \\ &\quad + \frac{c_2}{\theta} ln(\lambda_0\theta T_{LC} + 1) + c_3 T \end{aligned}$$

T에 관해서 비용함수 $C(T)$을 미분하면 다음과 같은 방정식을 만족하는 최적방출시간 T_C를 계산 할 수 있다[5, 6].

$$\frac{(c_1 - c_2)\lambda_0}{\lambda_0\theta T_C + 1} + c_3 = 0 \quad \cdots (21)$$

위 식에서 최적방출시간은 소프트웨어 지정 수명시간 인 T_{LC} 와 의존하지 않는다는 것을 알 수 있다. 이러한 사실은 무한 고장 평균값 함수를 가진 NHPP모형들이 새로운 결점들이 발생함으로서 몇 개의 고장이 야기 될 수 있는 점을 고려한 모형으로 적합 시킬 수 있다[5]. 실제로 만족할 만한 신뢰도가 부여되고 동시에 시스템 고장과 연계된 기대 총비용을 최소화시키기 위하여 필요하다면 충분한 테스팅을 계속해야 한다. 따라서 신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적 방출 시간이다. 따라서 로그 포아송 실행시간모형을 사용한 최적 방출시간 T_{OP}는 T_R과 T_C에 대하여 다음을 만족한다[3].

$$T_{OP} = Max(T_C, T_R) \quad \cdots (22)$$

(22) 식에서 T_R과 T_C는 다음 두 방정식에 의해서 계산될 수 있다.

$$\ln R_0 = -\frac{1}{\theta}\left[\ln(\lambda_0\theta(x+T_R)+1) - \ln(\lambda_0\theta T_R+1)\right] \quad \cdots (23)$$

$$\frac{(c_1 - c_2)\lambda_0}{\lambda_0\Theta T_C + 1} + c_3 = 0 \quad \cdots (24)$$

Ⅳ. 수치적인 예

이 장에서 Musa의 T1 자료[10]가 인용한 고장 간격 시간 자료(Failure interval time data)를 가지고 최적 방출시기를 분석하고자 한다. 이 자료는 최종 고장시간 $x_{15} = 296$

$(x_0 = 0)$이고 각 고장 간격시간에 대한 자료는 〈표 1〉 에 요약하였고 로그 포아송 실행시간모형 대한 모수 추정은 최우추정법을 이용하였고 비선형 방정식의 계산방법은 수치해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기값을 10^{-2} 와 10 을, 허용 한계(Tolerance for width of interval)는 10^{-10} 을 주고 수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였고 .베이지안 추정법을 이용하기 위해서 사전분포는 즉, β_1, β_2는 감마분포 $\Gamma(1, 0.001)$을 선택 이용하여 2절에 제시한 깁스 알고리즘을 적용하였다. IMSL〔15〕 소프트웨어를 사용하여 각 깁스 열의 전반부 $t/2$번 반복을 제외하고 후반부 $t/2$번 반복만을 고려하는 기법 즉, 분산 분석표를 이용하는 Gelman & Rubin〔14〕이 제시한 MC(Markov Chain)방법을 이용하여 각 모형에 대한 사후(Posterior) 밀도의 결과를 〈표 2〉 에 나타내었다. 이 표에서 수렴성을 확인하기 위해서 500, 2000(m)번 적용에 50, 70(t)번의 결과를 나타내었다〔12〕. 그리고 각 모수에 대한 사후 평균 $\hat{E}(\theta)$(단, θ는 모수)과 신용구간(C.I), 표준편차(S.D)를 나타내었다. 따라서 이 표에서 보여 주듯이 거의 유사한 값에 수렴함을 볼 수 있기 때문에 모형 선택이나 신뢰도에 있어서 2000번 적용에 70번 반복한 사후 평균(추정값)을 이용하였다. 이러한 결과는 〈표 2〉와 〈표 3〉에 요약되었다.

〈표 4〉에서는 $c_1 = 5(\$)$, $c_2 = 20(\$)$ 그리고 $c_3 = 0.5(\$)$ 라고 가정하고 시스템 수명시간은 2000시간이고 임무시간 x을 1.4이고 R_0을 0.95(95%)를 투입하여 각 모형에 대한 추정시간의 결과와 최적방출시간은 〈표 4〉에 요약되었다. 이 표에서 일반적인 최우추정법을 통한 모수 추정의 결과에 따른 방출시간은 베이지안 추정을 통한 결과를 비교해 보면 신뢰도 측면이든 비용측면이든 베이지안 추정을 반영한 최적방출시기가 길지 않음을 알 수 있었다. 물론 임무시간과 자료가 변경되면 그 결과는 달라질 수 있지만 베이지안 추정 접근은 이 분야에서 효율적임을 시사하고 있다. 즉, 알려져 있는 사실에 대한 주관적 의견을 경험이나 지식을 바탕으로 하여 사전정보를 만든 다음 실험을 통하여 얻어진 자료와 결합시켜 사후정보를 추출하는 과정으로 설명되는 베이즈 이론은 이 분야에서도 적용가능하다는 결론을 도출 할 수 있다.

표 1. 고장 간격자료
Table 1. Failure Interval data

고장번호 (i)	고장간격시간 $x_i - x_{i-1}$	누적고장시간 (x_i)
1	10	10
2	9	19
3	13	32
4	11	43
5	15	58
6	12	70
7	18	88
8	15	103
9	22	125
10	25	150
11	19	169
12	30	199
13	32	231
14	25	256
15	40	296

표 2. 대수형 포아송 모형에 대한 사후 밀도
Table 2. Poster densities of log Poisson execution time model

모 수	적용수 (m)	반복수 (t)	$\hat{E}(\beta_0)$	S.D	95% CI
β 0	500	50	5.3241	3.8521	(2.0832, 8.8754)
		70	5.3245	3.8424	(2.4215, 8.8436)
	2000	50	5.4321	3.8322	(2.3435, 8.5852)
		70	5.4342	3.8320	(2.8581, 8.4422)
모 수	적용수 (m)	반복수 (t)	$\hat{E}(\beta_1)$	S.D	95% CI
β 1	500	50	0.0516	0.0293	(0.0094, 0.0753)
		70	0.0527	0.0205	(0.0084, 0.0623)
	2000	50	0.0526	0.0188	(0.0071, 0.0695)
		70	0.0528	0.0147	(0.0023, 0.0393)

표 3. 각 모형의 모수 추정값
Table 3. Estimation of each model

Model	Log Poisson execution time	
MLE	$\hat{\theta}=0.0667$	$\hat{\lambda}_0=0.0871$
베이지안 추정	$\hat{\beta}_0=5.4332$ $(\hat{\theta}=0.1841)$	$\hat{\beta}_1=0.0528$ $(\hat{\lambda}_0=0.2869)$

표 4. 최적 방출시간 계산($R_0=0.95$)
Table 4. Calculation for optimal release time ($R_0=0.95$)

Log Poisson execution time	추정시간	최적방출 시간(T_{OP})
MLE	$\hat{T}_R=236.3763$ $\hat{T}_C=277.6453$	277.6453
베이지안 추정	$\hat{T}_R=128.6247$ $\hat{T}_C=144.0221$	128.6247

V. 결 론

본 연구는 베이지안 모수 추정을 통한 로그 포아송 실행시간 모형을 이용하여 최적 방출시기에 관한 문제를 알아보았다. 즉, 대용량 소프트웨어가 수정과 변경하는 과정에서 결점의 발생을 거의 피할 수 없는 상황이 현실이다. 실제로 만족할 만한 신뢰도가 부여되고 동시에 시스템 고장과 연계된 기대 총비용을 최소화시키기 위하여 필요하다면 충분한 테스팅을 계속해야 한다. 따라서 신뢰성 요구를 만족하고 총 비용을 최소화하는 상황이 최적 방출 시간이다. 본 연구에서는 베이지안 모수 추정이 이 분야에서도 적용 가능한 알고리즘이 될 수 있음을 확인하였다. 경우에 따라서는 왜도와 첨도 측면에서 효율적인 카파분포, 지수화지수분포 등 업데이트된 분포에 대한 방출 시기 문제를 비교 분석하는 연구도 가치 있는 일이라 판단되고 이 연구를 통하여 소프트웨어 개발자들은 방출최적시기를 파악 하는데 사전 정보를 포함한 베이지안 추정을 도구로 삼는다면 어느 정도 도움을 줄 수 있으리라 사료 된다.

참고문헌

〔1〕 김희철, "일반화 감마분포를 이용한 NHPP 소프트웨어 신뢰도 모형에 관한. 연구", 한국 컴퓨터정보학회 논문지, 제10권, 제6호, 27-35쪽, 2005년 12월.

〔2〕 S. S. Gokhale and K. S. Trivedi. "A time/structure based software reliability model", Annals of Software Engineering. 8, pp. 85-121, 1999.

〔3〕 Xie, M. and Homg, G. Y. " Software release time determination based on unbound NHPP model", Proceeding of the 24th International Conference on Computers and Industrial Engineering, pp. 165-168, 1999.

〔4〕 Musa, J. D and Okumoto, K. "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement", Proceeding the 7th International Conference on Software Engineering, pp. 230-238, 1984.

〔5〕 김대경, " Musa-Okumoto의 대수 포아송 실행시 간 모형에 근거한 비용-신뢰성 최적 정책", 품질 경영학회지, 제26권, 제3호, 141-149쪽, 1998년 9월.

〔6〕 김희철, "Log-Logistic 분포모형에 근거한 소 프트웨어 최적방출시기에 관한 비교연구", 한국컴퓨터정보학회 논문지, 제13권, 제7호, 1-9쪽, 2008년 12월.

〔7〕 Almering, V. and Genuchten, M, V and Cloudt, G. and Sonnemans, P. J. M. "Using Software Reliability Growth Models in Practice", IEEE SOFTWARE, pp. 82-88, 2007.

〔8〕 Yang, B. and Xie. M. "A study of operational and testing reliability in software reliability analysis", Reliability Engineering & System Safety, Vol, 70, pp,323-329, 2000.

〔9〕 Huang, C. Y. "Cost-Reliability-optimal release policy for software reliability models incorporating improvements in testing efficiency", The journal of Systems and software. Vol, 77, pp, 139-155, 2005.

〔10〕 Musa, J. D, Iannino, A. and Okumoto, K. "Software Reliability: Measurement, Prediction, Application", McGraw Hill, New York, 1987.

〔11〕 Kuo, L. and Yang, T. Y. "Bayesian Computation of Software Reliability", Journal of the American Statistical Association,Vol.91, pp.763-773, 1996.

〔12〕 이상식, 김희철, 송영재 "비동질적인 포아송과정을 사용한 소프트웨어 신뢰성장 모형에 대한 베이지안 신뢰성 분석에 관한 연구", 정보처리 학회논문지 D, 제10-D권, 제4호 .805-812쪽, 2003년 8월.

〔13〕 Casella, G. and George, E. I., "Explaining the Gibbs Sampler", The American Statistician, 46, pp.167-174, 1992.

〔14〕 Gelman, A. E. and Rubin D., "Inference from Iterative

Simulation Using Multiple Sequences", Statistical Science, 7, pp.457-472, 1992.
〔15〕 "User's Manual Stat/Library Fortran Subroutines for statistical analysis", IMSL, Vol.3, pp.1050-1054, 1987.

저 자 소 개

김 희 철
1992년 동국대학교 통계학과 석사
1998년 동국대학교 통계학과 박사
2000년 3월 ~ 2004년 2월
송호대학 정보산업계열 조교수
2005년 3월~현재 남서울대학교
산업경영공학과 전임강사
관심분야 : 소프트웨어 신뢰성공학, 웹 프로그래밍, 전산통계, 해외투자 및 쇼핑몰

신 현 철
2002년 원광대학교 컴퓨터공학과 (공학박사)
1994년 ~ 현재 백석문화대학 컴퓨터정보학부 교수
2005 ~ 현재 한국정보처리학회 이사
2005 ~ 현재 한국사이버테러정보전학회 부회장
관심분야 : 통신공학, 컴퓨터과학, 소프트웨어공학