

Makeham분포에 의존한 신뢰성모형에 근거한 학습효과 특성에 관한 비교 연구

김희철*, 신현철**

A comparative study on learning effects based on the reliability model depending on Makeham distribution

Hee-Cheul Kim*, Shin-Hyun Cheul**

요약 본 논문에서는 소프트웨어 제품을 개발하여 테스트를 하는 시행에서 소프트웨어 운용자들이 소프트웨어 검사 도구에 적용할 수 있는 학습기법에 근거한 NHPP 소프트웨어 모형에 대하여 비교 연구 하였다. 수명분포는 Makeham 분포를 이용하고 유한고장 NHPP모형을 적용하였다. 소프트웨어 오류 탐색 방법은 미리 인지하지 못하지만 자동적으로 발견되는 에러에 영향을 주는 영향요인과 사전경험에 기초하여 에러를 관찰하기 위하여 테스트 운용자가 미리 설정해놓은 요인인 학습효과의 영향에 대한 문제를 비교 분석하였다. 그 결과 학습요인이 자동 에러 탐색요인보다 큰 경우가 일반적으로 효율적인 모형으로 나타났다. 본 논문의 신뢰특성분석에서는 소프트웨어고장시간을 적용하고 모수추정 방법은 최우추정법을 적용하고 추세분석을 통하여 자료의 신뢰성을 확보한 이후에 평균제곱오차와 R^2 (결정계수)를 적용하여 효율적인 모형을 선택 비교 분석하였다. 이 연구를 통하여 소프트웨어 운영자들은 다양한 학습효과를 고려함으로써 소프트웨어 고장추세에 대한 기본지식을 파악하는데 하나의 지침으로 사용가능함을 보여주고 있다.

Abstract In this study, we investigated the comparative NHPP software model based on learning techniques that operators in the process of software testing and development of software products that can be applied to software test tool. The life distribution was applied Makeham distribution based on finite fault NHPP. Software error detection techniques known in advance, but influencing factors for considering the errors found automatically and learning factors ,by prior experience, to find precisely the error factor setting up the testing manager are presented comparing the problem. As a result, the learning factor is larger than automatic error that is usually well-organized model could be established. This paper, a trust characterization of applying using time among failures and parameter approximation using maximum likelihood estimation, after the effectiveness of the data through trend examination model selection were well-organized using the mean square error and R^2 . From this paper, the software operators must be considered life distribution by the basic knowledge of the software to confirm failure modes which may be helped.

Key Words : Learning Effects, Non-Homogeneous Poisson Process, Makeham Distribution, Mission Time

1. 서론

소프트웨어 결함으로 인한 컴퓨터 시스템의 고장은 컴퓨터 소프트웨어를 주로 사용하는 산업분야에 큰 악영향을 발생 시킬 수 있다. 따라서 소프

트웨어 개발과정에서 소프트웨어 신뢰성에 관한 제반 요인을 사전에 예측하는 작업은 컴퓨터 시스템의 고장을 파악하기 위한 필수적 사항이 된다. 이 요인들은 소프트웨어 관리자의 관리조건과 테

*Corresponding Author : Department of Industrial & Management Engineering, Namseoul University(kim1458@nsu.ac.kr)

**Department of Computer Engineering, BackSeok Culture University, Seoul Korea

Received October 05, 2016

Revised October 27, 2016

Accepted October 28, 2016

스팅 비용의 최소비용을 실행하기 위해서는 반드시 필요하다. 소프트웨어 테스트 혹은 디버깅 측면에서 비용을 최소화하기 이루기 위해서는 소프트웨어의 신뢰요인의 추이와 테스트 비용의 사전 추이를 예측하는 작업은 기본적인 사항이 된다. 결국 신뢰도, 비용 및 방출시간에 대한 추이를 예측하면서 소프트웨어 개발과정이 이루어져야 한다. 따라서 소프트웨어 제품의 결함사항에 대한 모형 개발은 필수적인 작업이 된다. 지금까지 많은 소프트웨어 신뢰성 요인 분석에 대한 연구가 제안되어 왔다. 여러 연구 중에서 비동질적 포아송 과정(Non-Homogeneous Poisson Process ; NHPP)에 의존한 신뢰성 모형[1]은 결함 관찰요인 측면에서는 효율적인 모형으로 알려져 있고 결함이 관찰되면 즉각적으로 디버깅되고 이러한 디버깅 시행에서 새로운 결함이 관찰되지 않는다는 사전 가정에 바탕을 두고 있다.

Gokhale과 Trivedi [1]은 고양된 비동질적인 포아송 과정(Enhanced NHPP) 모형을 제시하였고 Goel과 Okumoto [2]은 지수적 소프트웨어 신뢰성 모형(Exponential software reliability growth model)을 제안 하였다. Pham와 Zhang[3]는 테스트 커버리지(Coverage)에 기초하여 소프트웨어 안정도를 측정 할 수 있는 소프트웨어 안정도 모형을 제시하기도 하였다.

또한, S-형태 모형은 소프트웨어 운영자들이 소프트웨어 고장패턴에 대하여 고장 패턴을 파악 할 수 있는 학습과정을 제시하기도 하였다[4].

따라서 본 연구는 Makeham 분포를 적용한 NHPP 소프트웨어 모형에 대한 자동적으로 발견되는 에러에 영향을 주는 영향요인과 사전경험에 기초하여 에러를 관찰하기 위하여 테스트 운용자가 미리 설정해놓은 요인인 학습효과의 영향에 대한 문제를 비교 분석하였다.

2. 학습효과를 반영한 확률밀도함수와

분포함수

소프트웨어 테스트 시행 동안 학습효과는 테스트

시행자에 의해 유사하거나 조작 가능한 시행이 될 수 있으므로 이러한 효과들을 반영하는 기법은 소프트웨어 신뢰성에 중요한 사항이 될 수 있다.

소프트웨어 에러 추세를 발생하는 특성을 발견하기 위하여 자동 에러 탐색요인(autonomous errors-detected factor) γ 과 학습요인(learning factor) η 으로 구성된 영향요인들(influential factors)이 제안 할 수 있다[4]. 따라서 $f(t)$ 을 t 시점에서 에러를 검출될 확률을 나타내는 확률밀도함수이고 $F(t)$ 을 $(0, t]$ 시점까지의 분포함수라고 가정하면 영향요인에 의존한 관계는 다음과 같다[4].

$$f(t) = (\gamma + \eta F(t)) (1 - F(t)) \quad (1)$$

여기서, $\gamma > 0, \eta > 0$.

(1)식에서 자동 에러 관찰요인 γ 는 미리 인지하지 못하지만 테스트 과정에서 테스트 운용자가 자동적으로 에러를 검출하는 요인을 의미한다. 이에 반해 학습 요인 η 은 사전에 검출된 에러추세를 기초로 하여 에러를 검출하기 위하여 테스트 운용자가 미리 구축해놓은 요인을 의미한다.

한편 (1)식을 위험함수를 이용하면 다음과 같은 형태로 유도된다[4].

$$h(t) = (\gamma + \eta F(t)) \quad (2)$$

여기서, $h(t) = \frac{F'(t)}{1-F(t)} = \frac{f(t)}{1-F(t)}$

(2)식에서 분포함수와 확률밀도함수는 다음과 같이 수정 형태로 변경 가능하다.

$$F(t) = \frac{h(t) - \gamma}{\eta}, f(t) = F'(t) = \frac{h'(t)}{\eta} \quad (3)$$

3. Makeham분포 특성을 반영한 NHPP

유한고장 모형

Makeham 분포는 중요한 수명분포로서 수리적 데이터 모형에 사용되는 분포로서 강도함수는 다음과 같다[5].

$$\lambda(t) = ae^{bt}, a, b > 0 \quad (4)$$

한편, 무한고장모형에서 강도함수는 위험함수와 동일한 형태를 따른다[6, 7]. 즉 다음과 같은 관계를 가진다.

$$h(t) = \frac{F'(t)}{1-F(t)} = ae^{bt} = \lambda(t) \quad (5)$$

따라서 (3)식을 이용하면 다음과 같이 영향요인을 반영한 분포함수와 확률밀도함수는 다음과 같이 적용할 수 있다.

$$F(t) = \frac{(ae^{bt})^{-\gamma}}{\eta}, f(t) = F'(t) = \frac{(a b e^{bt})}{\eta} \quad (6)$$

유한고장 NHPP 모형에서 충분한 테스트 시간이 주어졌을 때 검색되어 질 수 있는 결함의 기대값을 θ 라고 표현하고 $F(t)$ 를 분포함수라고 표현하면 유한 고장 NHPP모형의 평균값 함수는 다음과 같다[6, 7].

$$m(t) = \theta F(t) \quad (7)$$

즉, (6)식과 (7)식을 이용하여 학습효과를 반영한 Makeham 분포 모형의 평균값함수 $m(t)$ 는 다음과 같이 적용할 수 있다[8].

$$m(t) = \theta F(t) = \theta \left[\frac{(ae^{bt})^{-\gamma}}{\eta} \right] \quad (8)$$

(7)식으로 부터 강도함수 $\lambda(t)$ 는 다음과 같이 유도된다[8, 9].

$$\lambda(t) = \theta F'(t) \quad (9)$$

즉, (6)식과 (9)식을 이용하여 학습효과를 반영한 Makeham 분포 모형의 강도함수 $\lambda(t)$ 는 다음과 같이 적용된다[9].

$$\lambda(t) = \theta f(t) = \theta F'(t) = \theta \frac{(a b e^{bt})}{\eta} \quad (10)$$

한편, 시간 $(0, t]$ 까지 관찰하기 위한 시간 멈춤(time truncated)모형은 n 번째 까지 관찰된 고장시점 자료를

$$x_k = \sum_{i=1}^k t_k \quad (k=1, 2, \dots, n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad (11)$$

이라고 하면 데이터 집합 D_n 는 $\{n, x_1, x_2, \dots, x_n; t\}$ 와 같이 구성된다. 이 시간 절단 모형에서의 θ 를 모수공간이라고 표시하면 우도함수는 다음과 같이 유도된다고 알려져 있다[9, 10].

$$L_{NHPP}(\theta | D_{x_n}) = \left(\prod_{i=1}^n \lambda(x_i) \right) \exp(-m(x_n)) \quad (12)$$

즉, (12)식을 이용하여 학습효과를 반영한 Makeham 분포 모형의 우도함수는 다음과 같이 유도된다.

$$L_{NHPP}(\theta | D_{x_n}) = \quad (13)$$

$$\left[\prod_{i=1}^n \theta \left(\frac{a b e^{bx_i}}{\eta} \right) \right] \exp \left[-\theta \left(\frac{(ae^{bx_n})^{-\gamma}}{\eta} \right) \right]$$

모수 추정방법은 최우추정법(MLE)을 적용하였고 최우추정법을 적용하기 위한 로그 우도 함수는 (13)식을 이용하면 다음과 같이 유도된다.

$$\ln L_{NHPP}(\theta | D_{x_n}) = \quad (14)$$

$$n \ln \theta + n \ln a + n \ln b + b \sum_{i=1}^n x_i - n \ln \eta - \theta \left(\frac{(ae^{bx_n})^{-\gamma}}{\eta} \right)$$

(14)식에서 분포의 속성을 잃지 않고 근 수렴을 용이하게 하기 위하여 척도모수 a 을 0.4로 고정시켜 수식을 전개하고자 한다. 즉 모수 θ 와 b 에 대하여 편미분 하여 다음과 같은 식을 만족하는

$\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 을 수치 해석적 방법으로 추정할 수 있다[8, 9].

$$\frac{\partial \ln L_{NHPP}(\theta | D_{x_n})}{\partial \theta} = \frac{n}{\theta} - \frac{(ae^{bx_n}) - \gamma}{\eta} = 0 \quad (15)$$

$$\text{즉, } \hat{\theta} = \frac{\eta n}{(ae^{bx_n}) - \gamma} \quad (16)$$

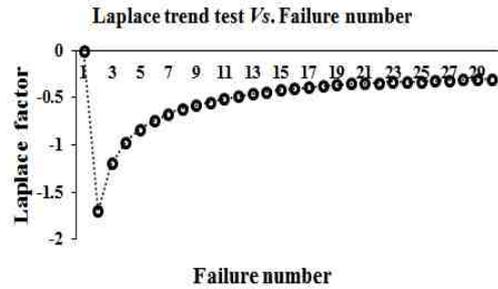
$$\frac{\partial \ln L_{NHPP}(\theta | D_{x_n})}{\partial b} = \frac{n}{b} + \sum_{i=1}^n x_i - \frac{\theta a}{\eta} x_n e^{bx_n} = 0 \quad (17)$$

4. 소프트웨어 고장자료 신뢰특성 분석

이 장에서 소프트웨어 고장시간 자료[10]를 적용하여 제시하는 신뢰모형에 대한 신뢰특성을 분석하고자 한다. 이 자료의 고장 시간은 [표 1]에 나열 되어 있다.

[표 1] 소프트웨어의 고장시간자료
[Table 1] software failure time data

Failure number	Failure time (hours)	Failure numbe	Failure time(hours)
1	30.02	16	151.78
2	31.46	17	177.5
3	53.93	18	180.29
4	55.29	19	182.21
5	58.72	20	186.34
6	71.92	21	256.81
7	77.07	22	273.88
8	80.9	23	277.87
9	101.9	24	453.93
10	114.87	25	535
11	115.34	26	537.27
12	121.57	27	552.9
13	124.97	28	673.68
14	134.07	29	704.49
15	136.25	30	738.68



[그림 1] 라플라스 추세검정
[Fig. 1] Laplace trend test

또 한 제시하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다 [11, 12].

추세 분석에는 라플라스 추세 검정(Laplace trend test)을 적용하였다. 이 검정을 실시한 결과 [그림 1]에 요약되었다. 이 그림에서 라플라스 요인(Factor)의 추정값이 -2와 2사이에 추정값을 나타냄으로서 극단값이 존재하지 않으므로 신뢰성장(reliability growth) 속성을 따른다고 할 수 있다. 따라서 이 자료를 적용하여 신뢰성 모형을 비교 분석하는 시행이 효율적임을 보여주고 있다[12].

모수추정은 최우추정법을 이용하고 모수 추정을 용이하게 하기 위하여 원래의 고장시간 데이터를 변수변환(Failure time $\times 10^{-1}$)하여 이용하였다. 비선형 방정식의 계산방법은 이분법(Bisection method)을 사용하였다. 이러한 모수추정은 초기 값을 10^{-3} 과 5.00을, 허용한계(Tolerance for width of interval)는 10^{-5} 을 주고 수렴성을 확인 하면서 충분한 반복횟수인 50번을 C-언어를 이용하여 모수 추정을 수행하였다.

[표 2]에서 학습요인 η 과 자동 에러 탐색요인 γ 이 같을 경우($\eta = \gamma = 0.5$)나 자동 에러 탐색요인이 학습요인보다 큰 경우는($\gamma > \eta$) 근이 수렴되지 않아 본 연구에서 η 은 0.9에서 0.6, 자동 에러 탐색요인 γ 은 각각 0.1부터 0.4까지 고려하였다. 추가적으로 모형비교를 위하여 MSE 와 R^2 의 추정값도 요약되었다.

[표 2] η 과 γ 을 고려한 각 모형의 모수 최우추정값 (MLE) 및 MSE 와 R^2

[Table 2] Parameter estimation of the each model and MSE , R^2 considering η and γ

Model		MLE		Model Comparison	
η	γ	\hat{b}_{MLE}	$\hat{\theta}_{MLE}$	MSE	R^2
0.9	0.1	0.0182	18.893 9	37.4092	0.6957
0.8	0.2	0.0157	22.254 0	38.8720	0.7104
0.7	0.3	0.0121	30.944 7	40.9105	0.7317
0.6	0.4	0.0111	24.481 1	127.818	0.7376

Note. MLE : Maximum likelihood estimation;

MSE : Mean square error;

R^2 : Coefficient of determination;

η : the learning factor;

γ : autonomous errors-detected factor

[표 2]에서 평균제곱오차(MSE)는 실제 관찰 값 과 예측 값에 대한 차이를 추정하는 도구로서 다음과 같다[9, 12].

$$MSE = \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{n - k} \quad (18)$$

여기서, $m(x_i)$ 은 시간(0, x_i]까지 나타난 고장 들의 누적수를 의미하고 $\hat{m}(x_i)$ 는 x_i 시점까지 평균값 함수로부터 추정된 고장의 누적수를 나타낸다. 그리고 n 은 관찰 값의 수를 의미하고 k 는 모 수의 수를 나타낸다. 즉 비교결과에 있어서 평균제 곱오차 값이 작으면 비교적으로 효율적인 모형이 된다.

R^2 (결정계수)는 관찰 값의 차이에 대한 설명력 을 나타내는 기준으로 다음과 같다[9, 12].

$$R^2 = 1 - \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{\sum_{i=1}^n \left(m(x_i) - \frac{\sum_{j=1}^n m(x_j)}{n} \right)^2} \quad (19)$$

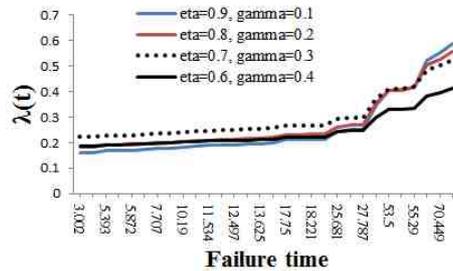
즉 비교에 있어서 결정계수 값이 크면 상대적으

로 효율적인 모형이 된다.

5. 결과분석 및 검토항목

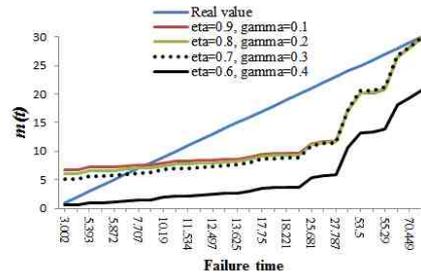
결과적으로 [표 2]에서 학습요인 η 이 증가 할수 록 효율적인 모형으로 나타나고 있다. 그리고 설 명력 측면에서는 결정 계수값이 0.6보다 크게 나타 나 실제 값과 추정된 값에 대한 차이의 대한 설명 력이 높다. 그러나 학습요인 η 이 증가 할수록 관 찰 값의 차이에 대한 설명력이 낮게 나타나고 있 다.

Intensity function Vs. Failure time



[그림 2] 강도함수의 형태
[Fig. 2] Pattern of intensity function

Mean value function Vs. Failure time

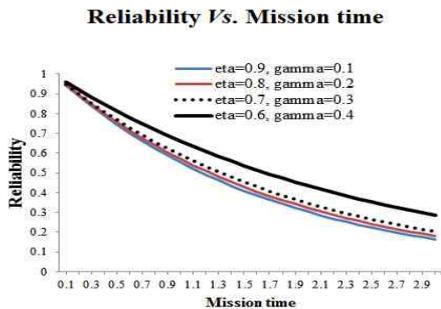


[그림 3] 평균값 함수의 형태
[Fig. 3] Pattern of mean value function

[그림 2]의 강도함수의 추이는 고장시간 초기에 는 학습요인 η 이 작을수록 높게 나타나지만 고장 시간이 경과함에 따라 높게 나타나는 비감소함수 형태를 나타내고 있다. [그림 3]은 평균값 함수에 대한 패턴이 요약 되었다.이 그림에서 모든 모형은

참값과의 차이에서 과소평가 추이가 이루어졌고 학습요인 η 이 작을수록 참값과의 차이의 폭이 작은 추세를 보이고 있다.

[그림 4]의 신뢰도 함수의 형태에서 임무시간이 경과함에 따라 완만히 비증가 패턴으로 나타나고 학습요인 η 의 값이 작을수록 높은 신뢰도를 나타내고 있다



[그림 4] 신뢰도 함수의 형태
[Fig. 4] Pattern of reliability

6. 결론

빅 데이터 처리를 위하여 소프트웨어의 성능을 비교 분석하는 과정에서 결함의 발생을 거의 피할 수 없는 상황이 현실이다. 따라서 소프트웨어 운용자들이 소프트웨어 및 검사 도구에 적용 할 수 있는 학습과정에 관련된 소프트웨어 모형에 대하여 연구 하였다. 미리 사전에 인지하지는 못하지만 저절로 발생하는 에러를 관여하는 영향요인과 사전정보에 의해 경험적으로 에러를 발견하기 위하여 테스트 운용자가 설정해놓은 요인인 학습효과의 속성에 대한 문제를 비교 분석 하였다.

따라서 본 논문에서 적합 시킨 소프트웨어 고장자료에 의한 모수 추정 값이기 때문에 다른 자료를 적합 시키면 모수 추정 결과는 달라질 수 있지만 학습요인이 자동 에러 탐색요인보다 큰 경우가 일반적으로 효율적인 모형으로 간주 될 수 있다. 따라서 학습요인을 반영한 이 분야에서 효율적 모형으로 선택 될 수 있음을 확인 할 수 있었다.

이 연구를 통하여 소프트웨어 운용자들은 수명분포에서 파생된 학습요인을 인지하는데 도움을 줄 수 있다.

REFERENCES

- [1] Gokhale, S. S. and Trivedi, K. S. A, "time/structure based software reliability model", *Annals of Software Engineering*, 8, pp. 85-121, 1999.
- [2] Goel A L, Okumoto K, "Time-dependent fault detection rate model for software and other performance measures", *IEEE Trans. Reliab*, 28, pp.206-11, 1978.
- [3] Pham H, Zhang X., "NHPP software reliability and cost models with testing coverage", *Eur. J. Oper. Res*, 145, pp.445-454, 2003.
- [4] Kuei-Chen, C., Yeu-Shiang, H., and Tzai-Zang, L., "A study of software reliability growth from the perspective of learning effects", *Reliability Engineering and System Safety* 93, pp. 1410 - 1421, 2008.
- [5] K. G. Manton, E. stallard and J. W. Vaupel, "Alternative Models for the Heterogeneity of Mortality Risks Among the Aged", *Journal of the American Statistical Association*, Vol. 81, No. 395, pp.635-6 44, 1986.
- [6] L. Kuo and T. Y. Yang, "Bayesian Computation of Software Reliability", *Journal of the American Statistic al Association*, Vol.91, pp. 763-773, 1996.
- [7]Hee-Cheul Kim, "The Property of Learning effect based on Delayed Software S-Shaped Reliability Model using Finite NHPP Software Cost Model", *Indian Journal of Science and Technology*, Vol. 8, No. 34, pp. 1-7, 2015.
- [8] Kim H-C, "A Performance Analysis of Software Reliability Model using Lomax and Gompertz Distribution Property", *Indian*

Journal of Science and Technology, Vol. 9, No. 20, pp. 1-6, 2016.

[9] Tae-Hyun Yoo, "The Infinite NHPP Software Reliability Model based on Monotonic Intensity Function", *Indian Journal of Science and Technology*, Vol. 8, No. 14, pp. 1-7, 2015.

[10] Satya Prasad R, Rao KRH, Kantha RRL. "Software reliability measuring using modified maximum likelihood estimation and SP",. *International Journal of Computer Applications*, Vol. 21, No. 7, pp.1-5,,2011.

[11] K. Kanoun and J. C. Laprie, "Handbook of Software Reliability Engineering", *M.R.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY*, pp. 401-437, 1996.

[12] Kim H-C, Kim K-S, "Software Development Cost Model based on NHPP Gompertz Distribution", *Indian Journal of Science and Technology*, Vol. 9, No. 20, pp. 1-6, 2016.

저자 약력

김 희 철 (Hee-Cheul Kim) [정회원]



- 1992년 2월 : 동국대학교 통계학과 (이학석사)
- 1998년 8월 : 동국대학교 통계학과 (이학박사)
- 2005년 3월 ~ 현재 : 남서울대학교 산업경영공학과 교수

<관심분야>

소프트웨어 신뢰성 공학, 전산 통계, 웹프로그래밍

신 현 철 (Shin-Hyun Cheul) [정회원]



- 2002년 2월 : 원광대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 : 한국정보처리학회 이사
- 1994년 3월 ~ 현재 : 백석문화대학교 컴퓨터공학부 교수

<관심분야>

통신공학, 컴퓨터과학, 소프트웨어공학
소프트웨어공학