

## 미니맥스 수명분포의 형상모수를 따르는 소프트웨어 신뢰모형에 관한 신뢰속성에 관한 연구

\*

### A Study on the Reliability Attributes of the Software Reliability Model Following the Shape Parameter of Minimax Life Distribution

Hee-Cheul Kim\*

**요약** 본 논문은 베타분포의 특수형태인 미니맥스(Minimax distribution)를 수명분포를 따르고 유한고장수를 가진 비동질적인 포아송 과정에 기초한 소프트웨어의 신뢰성 모형에 대한 형상모수의 특성과 유용성에 대하여 비교 논의 되었다. 그 결과 평균제곱오차는 미니맥스 수명분포의 형상모수가 1인 경우가 1보다 작은 경우와 큰 경우에 비해 상대적으로 가장 작고 또한 형상모수가 1인 경우가 결정계수도 높은 성향으로 나타나기 때문에 형상모수가 1인 경우가 효율적 모형으로 판단된다. 본 연구에서 평균제곱오차는 미니맥스 수명분포의 형상모수를 적용할 경우에 소프트웨어 신뢰성에 대한 고장유형을 인지하기 위하여 평균제곱오차와 결정계수, 신뢰구간을 사용하면 소프트웨어 고장 특성을 파악하는데 기본지침으로 사용 할 수 있으리라 사료된다. 제안된 모형의 결정계수 값이 95%이상 추정되어 비교적 소프트웨어 신뢰성 분야에서 유용성이 있는 모형임을 확인 할 수 있었다. 이 연구를 통하여 소프트웨어 설계 및 사용자들은 수명분포의 특징을 반영한 형상모수를 적용할 경우에 소프트웨어 신뢰성에 대한 고장유형을 인지하기 위하여 평균제곱오차와 결정계수, 신뢰구간을 사용하면 소프트웨어 고장특성을 파악하는데 기본지침으로 사용 할 수 있으리라 사료된다.

**Abstract** This paper, following the shape parameters of the minimax distribution, describes the special form of the beta distribution, the Minimax distribution, as a function of the shape parameters for the software reliability model based on the non-homogeneous Poisson process. Characteristics and usefulness were discussed. As a result, the case of the shape parameter 1 of Minimax distribution than less than and greater in mean squared error is the smallest, in determination coefficient, appears to be high, the shape parameter 1 of Minimax distribution regard as an efficient model. The estimated determination coefficient of the proposed model is estimated to be more than 95%, which is a useful model in the field of software reliability. Through this study, software design and users can identify the software failure characteristics using mean square error, decision coefficient, and confidence interval can be used as a basic guideline..

**Key Words** : Confidence interval, Determination coefficient, Mean squared error, Minimax distribution, NHPP,

#### 1.

소프트웨어 시스템은 우리 삶의 필수요소가 되었다.

이러한 시스템은 유용성과 안정성을 확보 할 수 있다면 고객에게 고품질의 서비스를 제공 할 수 있기 때문에 매우 중요한 문제 중에 하나이다. 그러나 소프트웨어 개발

Funding for this paper was provided by Namseoul University year 2018

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

Received June 31, 2018

Revised June 06, 2018

Accepted July 11, 2018

은 어렵고 복잡한 과정으로 판단된다. 따라서 소프트웨어 개발자들의 주된 관심사는 소프트웨어 시스템의 안정성을 향상시키는 데 있다. 이로 인해 소프트웨어 안정성 엔지니어링 연구가 시작되었으며 지난 수십 년 동안 소프트웨어 신뢰 성장모형(Software reliability growth model)이 제안되었다. 즉, 잔류 고장의 수, 고장률 등의 신뢰성 특성을 추정하기 위해 제어된 시험환경 내에서 결합 강도 함수 및 평균값 함수를 이용하여 비동질 포아송 과정(Non-homogeneous Poisson process ; NHPP)에 기초한 소프트웨어 신뢰도 모형이 개발되었다[1]. 소프트웨어 신뢰성 모형은 소프트웨어의 신뢰성, 남은 고장 수, 고장강도, 전체 소프트웨어 개발 비용 등을 추정하고 예측하는데 사용된다. 현재까지 다양한 소프트웨어 안정성 모형연구가 개발되었다. Yamada와 Osaki [2]는 최우추정법을 이용하여 평균값 함수의 결과를 추정 할 수 있다고 하였고 Huang [3]은 평균값 함수의 신뢰 구간을 나타내는 그래프를 제시하여 설명하였다. Chatterjee와 Singh [4]는 불완전한 디버깅을 통한 물류 지수 테스트 커버리지 함수를 통합한 NHPP에 기반한 소프트웨어 신뢰성 모형을 제안했다.

Teng 과 Pham [5]은 환경의 불확실성과 소프트웨어 고장률에 미치는 영향을 포착하는 일반화 된 모형에 대해 논의했다.

본 연구에서는 수명분포가 베타분포의 특수형태인 미니맥스(Minimax distribution)를 적용하고 유한고장수를 가진 비동질적인 포아송 과정에 기초한 소프트웨어의 신뢰성 모형에 대한 형상모수의 특성과 유용성에 대하여 비교 논의 되었다.

## 2. NHPP

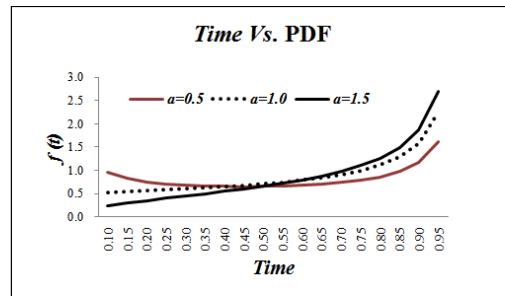
### 2.1

미니맥스 분포(Minimax distribution)는 베타분포(Beta distribution)에 비해 사용하기가 비교적 쉽다는 이유로 베타분포의 대안으로 사용되어 지고 미니맥스 분포의 확률밀도 함수(PDF)와 누적분포함수(CDF)는 각각 다음과 같다[6].

$$f(t) = ab t^{a-1} (1-t^a)^{b-1}, \quad t \in (0, 1] \quad (1)$$

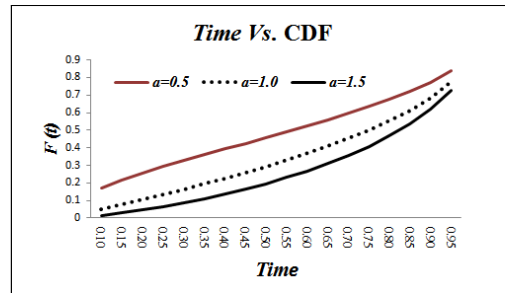
$$F(t) = 1 - (1-t^a)^b, \quad (2)$$

(1)식과 (2)식에서  $a$ 와  $b$ 는 형상모수를 의미한다. 이러한 미니맥스 분포에서 Oguntunde와 Adejumo[6]는  $a = 1$ 인 경우를 분석하였다. 따라서 본 연구는 형상모수  $a$ 가 0.5와 1 그리고 1.5인 경우를 선택하여 소프트웨어 신뢰성 모형을 분석하고자 한다.  $b=0.5$ 일 때 형상모수( $a$ )에 따른 확률밀도함수와 누적분포함수는 [그림 1]과 [그림 2]에 나타내었다.



1.  $b=0.5$  (a)

Fig. 1. Probability density distribution functions according to shape parameters ( $a$ ) and  $b=0.5$



2.  $b=0.5$  (a)

Fig. 2. Cumulative distribution function according to shape parameters ( $a$ ) and  $b=0.5$

### 2.2

유한고장 NHPP 모형에서 테스트링 시간  $(0, t]$  동안 발견되는 결함의 기대값을  $\theta$ 라고 하면 유한 고장 NHPP모형의 평균값 함수와 강도함수는 다음과 같다[1,2].

$$m(t) = \theta F(t), \quad \lambda(t) = \theta F'(t) \quad (3)$$

따라서 시간  $(0, t]$ 까지 관찰하기 위한 시간 절단(Time

truncated)모형은  $n$  번째까지 고장시점 자료는 다음과 같은 패턴으로 표현 할 수 있다.

$$x_k = \sum_{i=1}^k t_i \quad (k=1,2,\dots,n; 0 \leq x_1 \leq x_2 \leq \dots \leq x_n) \quad \dots (4)$$

즉,  $\theta$  을 모수공간이라고 표시하면 유한고장 우도함수는 다음과 같다[7].

$$L_{NHPP}(\theta | \underline{x}) = \left( \prod_{i=1}^n \lambda(x_i) \right) \exp(-m(x_n)) \quad \dots \dots \dots (5)$$

단,  $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$

(3)식과 (5)식을 이용하여 미니맥스 수명분포를 적용하면 다음과 같은 형태로 유도 할 수 있다.

$$L_{NHPP}(\theta | \underline{x}) = \left( \prod_{i=1}^n ab x_i^{a-1} (1-x_i^a)^{b-1} \right) \dots \dots \dots (6) \\ \cdot \exp[-(1-(1-x_n^a)^b)]$$

최우추정법(MLE)을 위하여 (6)식을 이용한 로그우도함수는 다음과 같다.

$$\ln L(\theta | \underline{x}) \quad (7) \\ = n \ln \theta + n \ln a + n \ln b + (a-1) \sum_{i=1}^n \ln x_i \\ - (b-1) \sum_{i=1}^n \ln(1-x_i^a) - \theta [1-(1-x_n^a)^b]$$

(7)식을 이용하고 형상모수  $a$ 는 사전에 지정하고 다른 모수에 대한 최우추정량  $\hat{\theta}_{MLE}$  와  $\hat{b}_{MLE}$  은 다음 조건식을 만족해야 한다.

$$\frac{\partial \ln L(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - [1-(1-x_n^a)^b] = 0 \quad (8)$$

$$\frac{\partial \ln L(\theta | \underline{x})}{\partial b} = \frac{n}{b} + \sum_{i=1}^n \ln(1-x_i^a) + \theta(1-x_n^a)^b \ln(1-x_n^a) = 0 \quad (9)$$

### 2.3

#### 2.3.1

평균제곱오차(Mean square error;  $MSE$ )는 참값(실제 관찰 값)과 추정 값(예측 값)에 대한 차이를 비교하는 기준으로서 다음과 같다[9]

$$MSE = \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{n-k} \quad (10)$$

단,  $m(x_i)$ 은 관찰시점(0,  $x_i$ ]까지 고장들의 누적개수를 나타내고  $\hat{m}(x_i)$ 는  $x_i$  시점까지 평균값 함수로부터 추정된 고장의 누적개수를 나타내고  $n$  은 관찰된 고장의 수이고  $k$ 는 추정된 모수의 수를 나타낸다. 따라서 평균제곱오차 값이 작으면 상대적으로 효율적인 모형으로 판단된다. 그리고 관찰 값의 차이에 대한 설명력을 의미하는 기준인 결정계수(Determination coefficient;  $R^2$ )는 다음과 같다 [8].

$$R^2 = 1 - \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{\sum_{i=1}^n \left( m(x_i) - \sum_{j=1}^n m(x_j)/n \right)^2} \quad (11)$$

즉 비교에 있어서 결정계수 값이 크면 오차가 작아 상대적으로 유용성 있는 모형으로 간주된다.

#### 2.3.2

소프트웨어의 신뢰모형에 대한 신뢰 구간은 신뢰성 특징을 이용하여 추정 될 수 있다. 특히 평균값 함수에 대한 신뢰구간은 다음과 같다[1]

$$\hat{m}(t) \pm Z_{\alpha/2} \sqrt{\hat{m}(t)} \quad (12)$$

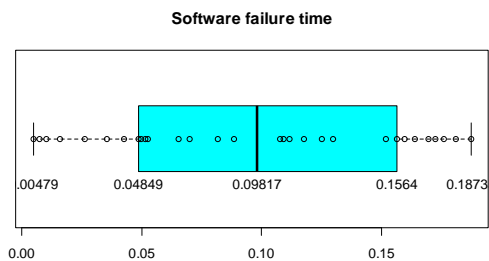
단  $Z_{\alpha/2}$  는 표준화 정규분포의  $100(1-\alpha)$  백분위수를 의미한다.

3.

1.

Table 1. Failure time data

Failure number	Failure time (hours)	Failure time $\times 10^{-2}$ (hours)
1	0.479	0.00479
2	0.745	0.00745
3	1.022	0.01022
4	1.576	0.01576
5	2.610	0.02610
6	3.559	0.03559
7	4.252	0.04252
8	4.849	0.04849
9	4.966	0.04966
10	5.136	0.05136
11	5.253	0.05253
12	6.527	0.06527
13	6.996	0.06996
14	8.170	0.08170
15	8.863	0.08863
16	10.771	0.10771
17	10.906	0.10906
18	11.183	0.11183
19	11.779	0.11779
20	12.536	0.12536
21	12.973	0.12973
22	15.203	0.15203
23	15.640	0.15640
24	15.980	0.15980
25	16.385	0.16385
26	16.960	0.16960
27	17.237	0.17237
28	17.600	0.17600
29	18.122	0.18122
30	18.735	0.18735



3.

Fig. 3. Result of Box-plot

이 장에서 소프트웨어 고장시간 정보자료[9] (Failure time information data)를 적용하여 신뢰모형들의 특징을 비교 분석하고자 한다. 이 자료는 [표 1]에 나열 되었고 극단값이 발생하는지를 알아보기 위하여 자료에 대한 추세 검정이 선행 되어야 하는데 본 연구에서는 추세검정을 박스-플롯(Box-plot)을 이용하였다[10].

[그림 3]의 결과에서 보여 주듯이

상한(=  $0.1564 + 1.5 * (0.1564 - 0.04849) = 0.318265$ )과 하한(=  $0.04849 - 1.5 * (0.1564 - 0.04849) = -0.11338$ )사이를 벗어난 자료가 없기 때문에 이상값 혹은 극단값이 존재하지 않음을 보여주고 있다. 따라서 이 자료를 사용하여 신뢰성 모형의 특징을 추정하는 작업이 효율적이다[8, 10, 11]

제시하는 미니맥스 수명분포는  $t \in (0, 1]$ 의 값에서 정의되기 때문에 원래의 데이터를 수치변환(Failure time  $\times 10^{-2}$ ) 하여 모수추정을 최우추정법을 이용하고 비선형 방정식의 계산방법은 수치 해석적 방법인 이분법(Bisection method)을 활용하였다. 이러한 계산은 초기 값을  $10^{-5}$ 과 1.0을, 허용 한계(Tolerance for width of interval)는  $10^{-5}$ 을 선택하고 수렴성을 확인되는 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였다. 그 결과는 [표 2]에 나타내었다.

2.  $MSE, R^2$

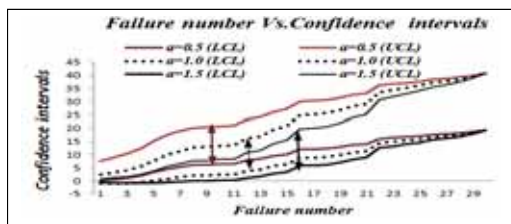
Table 2. Parameter estimation and  $MSE, R^2$

Model	MLE	Model Comparison	
		$MSE$	$R^2$
$a = 0.5$	$\hat{\theta} = 928049.3$ $\hat{b} = 3.23 \times 10^{-5}$	12.716	0.9824
$a = 1.0$	$\hat{\theta} = 477.188$ $\hat{b} = 6.29 \times 10^{-2}$	1.782	0.9897
$a = 1.5$	$\hat{\theta} = 303469.4$ $\hat{b} = 9.89 \times 10^{-5}$	15.088	0.9652

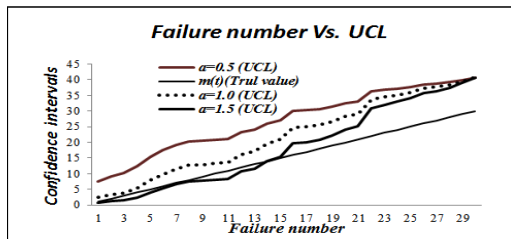
[표 2]에서 실제 값과 예측 값에 대한 차이에 대한 척도를 의미하는 평균제곱오차( $MSE$ )는 미니맥스 수명분포의 형상모수  $a = 1.0$ 인 경우가  $a = 0.5$ 와  $a = 1.5$  상대적으로 가장 작고 예측 값의 차이에 대한 설명력에 대한 척도를 나타내는 결정계수도  $a = 1.0$ 인 경우가 다른 형상모수를 적용 했을 때 보다 높은 성향으로 나타나기 때문

에 형상모수  $a = 1.0$ 인 경우가  $a = 0.5$ 와  $a = 1.5$ 보다 효율적이라 판단되며 제안된 모형의 결정계수 값이 95%이상 추정되어 비교적 소프트웨어 신뢰성 분야에서 유용성이 있는 모형임을 확인 할 수 있다[8].

[그림 4]에서는 각 고장 수에 대한 신뢰구간(유의수준 ( $\alpha$ )이 0.05)이 요약되었다. 이 그림에서 수명분포의 형상모수  $a = 1.0$ 인 경우의 신뢰구간을 기준으로  $a = 0.5$  경우는 높은 쪽에  $a = 1.5$ 은 낮은 쪽에 신뢰구간이 형성되었다.



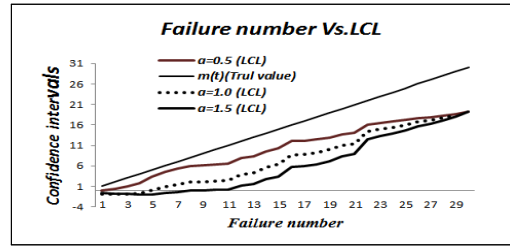
4. Fig. 4. Confident interval of mean value function



5. Fig. 5 Upper confident interval of mean value function

[그림 5]에서는 각 고장수에 대한 상한 신뢰구간이 요약되었다. 이 그림에서 참값과 비교하여 상한은 수명분포의 형상모수  $a = 1.5$ 가  $a = 0.5$ 경우와  $a = 1.0$ 인 경우가 가깝게 추정되었다.

또한 [그림 6]에서는 각 고장수에 대한 하한 신뢰구간이 요약되었다. 이 그림에서 참값과 비교하여 하한은 수명분포의 형상모수  $a = 0.5$ 경우가  $a = 1.0$ 와  $a = 1.5$ 인 경우보다 가깝게 추정되었다.



6. Fig. 6. Lower confident interval of mean value function

4.

소프트웨어 개발과정에서 테스트 작업이나 실제소프트웨어 사용과정에서 고장발생 속성 혹은 고장 발생현상을 정량적으로 모형화하면 소프트웨어 유용성을 비교 분석하여 상대적으로 효율성 평가를 할 수 있다. 본 논문에서는 운영 환경의 불확실성에 따라 미니맥스 수명분포의 형상모수를 따를 때 오류 탐지율 기능을 기반으로 한 새로운 소프트웨어 신뢰성 모형에 대해 비교 논의했다.

그 결과 본 연구에서 평균제곱오차는 미니맥스 수명분포의 형상모수  $a = 1.0$ 인 경우가  $a = 0.5$ 와  $a = 1.5$  상대적으로 가장 작고 결정계수도  $a = 1.0$ 인 경우가 다른 형상모수를 적용 했을 때 보다 높은 성향으로 나타나기 때문에 형상모수  $a = 1.0$ 인 경우가  $a = 0.5$ 와  $a = 1.5$  가 효율적이라 판단되며 제안된 모형의 결정계수 값이 95%이상 추정되어 비교적 소프트웨어 신뢰성 분야에서 유용성이 있는 모형임을 확인 할 수 있다.

각 고장수에 대한 상한 신뢰구간은 수명분포의 형상모수  $a = 1.0$ 인 경우의 신뢰구간을 기준으로  $a = 0.5$  경우는 높은 쪽에  $a = 1.5$ 은 낮은 쪽에 신뢰구간이 형성되었고 상한은 참값과 비교하여 수명분포의 형상모수  $a = 1.5$ 가  $a = 0.5$ 경우와  $a = 1.0$ 인 경우보다 가깝게 추정되었지만 하한 신뢰구간은 하한은 수명분포의 형상모수  $a = 0.5$ 경우가  $a = 1.0$ 와  $a = 1.5$ 인 경우보다 가깝게 추정되었다.

이 연구를 통하여 소프트웨어 설계 및 사용자들은 수명분포의 특징을 반영한 형상모수를 적용할 경우에 소프트웨어 신뢰성에 대한 고장유형을 인지하기 위하여 평균제곱오차와 결정계수, 신뢰구간을 사용하면 소프트웨어 고장 특성을 파악하는데 기본지침으로 사용 할 수 있으리라 사료된다

## REFERENCES

- [1] K. Y. Song, I. H. Chang, H. Pham, "A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments", *Applied Science*, Vol.7, No.983, pp. 1-16, 2017.
- [2] S. Yamada, S. Osaki, "Software reliability growth modeling: models and applications", *IEEE Transactions on Software Engineering*, Vol. 11, No. 12, pp.1431-1437, 1985.
- [3] C.Y. Huang, "Performance analysis of software reliability growth models with testing-effort and change-point", *Journal of Systems and Software*, Vol. 76, No. 2, pp. 181-194, 2005.
- [4] S. Chatterjee, J. B. Singh, "A NHPP based software reliability model and optimal release policy with logistic-exponential test coverage under imperfect debugging", *International Journal of System Assurance Engineering and Management*, Vol. 5, Issue 3, pp.399-406, 2014.
- [5] X. Teng, H. Pham, "A new methodology for predicting software reliability in the random field environments", *IEEE Transactions on Reliability*, Vol.55, No.3, pp.458-468, 2006.
- [6] P. E. Oguntunde, A. O. Adejumo, "A Note on the Minimax Distribution", *Covenant Journal of Physical and Life Sciences (CJPL)*, Vol. 3 No. 1. pp. 1-8, 2015.
- [7] S. S. Gokhale, K. S. Trivedi, "A time/structure based software reliability model", *Annals of Software Engineering*, Vol. 8, pp. 85-121, 1999.
- [8] C. Kuei-Chen, H. B. Yeu-Shiang, L. Tzai-Zhang, "A study of software reliability growth from the perspective of learning effects", *Reliability Engineering and System Safety*, Vol. 93, Issue 10, pp. 1410-1421, 2008.
- [9] Y. Hayakawa, G. Telfar "Mixed poisson-type processes with application in software reliability", *Mathematical and Computer Modelling*, Vol. 31, pp.151-156, 2000.
- [10] Hee-Cheul Kim, "A Comparative Study on Software Reliability Models with Shape Parameter of Type-2 Gumble Life Distribution", *International Journal of Soft Computing*, Vol. 12, No. 5-6, pp. 351-354, 2017.
- [11] Kim K-S, Kim H-C, "The Performance Analysis of the Software Reliability NHPP Log-linear Model Depend on Viewpoint of the Learning Effects", *Indian Journal of Science and Technology*, Vol. 9, No. 37, pp. 1-5, 2016.

(Hee-Cheul Kim)

[ ]



- 1992 2 : ( )
- 1998 8 : ( )
- 2005 3 :

&lt; &gt;

,