

Gompertz 성장곡선 기반 소프트웨어 신뢰성 성장 모델

박 석 규* · 이 상 운**

요 약

Gompertz 성장곡선에 기반한 기존의 소프트웨어 신뢰성 성장모델들은 모두 대수형이다. 대수형 Gompertz 성장곡선에 기반한 소프트웨어 신뢰성 성장 모델들은 모수 추정에 어려움을 갖고 있다. 그러므로 본 논문은 로지스틱형 Gompertz 성장곡선에 기반한 신뢰성 성장 모델을 제안한다. 13개의 다른 소프트웨어 프로젝트로부터 얻은 고장 데이터를 분석하여 그 유용성을 검토하였다. 모델의 모수들은 변수변환을 통한 선형 회귀분석과 Virene의 방법으로 추정되었다. 제안된 모델은 평균 상대 예측 오차에 기반하여 성능을 비교하였다. 실험 결과 제안된 모델은 대수형 Gompertz 성장 곡선에 기반한 모델보다 좋은 성능을 보였다.

A Software Reliability Growth Model Based on Gompertz Growth Curve

Seok-Gyu Park* · Sang-Un Lee**

ABSTRACT

Current software reliability growth models based on Gompertz growth curve are all logarithmic type. Software reliability growth models based on logarithmic type Gompertz growth curve has difficulties in parameter estimation. Therefore this paper proposes a software reliability growth model based on the logistic type Gompertz growth curve. Its usefulness is empirically verified by analyzing the failure data sets obtained from 13 different software projects. The parameters of model are estimated by linear regression through variable transformation or Virene's method. The proposed model is compared with respect to the average relative prediction error criterion. Experimental results show that the proposed model performs better the models based on the logarithmic type Gompertz growth curve.

키워드 : 소프트웨어 신뢰성 성장 모델(Software Reliability Growth Model), 성장곡선(Growth Curve), Gompertz 모델(Gompertz Model), 대수형(Logarithmic Type), 로지스틱형(Logistic Type)

1. 서 론

일반적으로 소프트웨어의 품질을 정량적으로 표현하는 속성중의 하나가 소프트웨어 신뢰성(Software Reliability)이다. 소프트웨어 신뢰성은 주어진 환경하에서 주어진 시간 동안 소프트웨어가 고장 없이 작동할 확률로 정의된다[1]. 소프트웨어 개발 과정 중 시험단계와 운영단계에서 고장(Failure)의 원인인 소프트웨어에 내재된 결함(Fault)이 검출되고 제거된다. 그 결과 시험이 진행될수록 소프트웨어 시스템에 남아 있는 결함의 수는 감소되어 고장 발생 확률이 감소하며, 고장 발생 시간 간격(Time Between Failures)이 보다 길어진다. 이로 인해 소프트웨어에 대한 신뢰성은 성장하게 된다.

소프트웨어 신뢰성을 평가하기 위해 고장 발생 현상에

대한 통계적 모델이 일반적으로 사용되고 있다. 소프트웨어 인도 일자나 자원을 할당하는 의사결정 문제가 모델의 예측 결과에 크게 의존하기 때문에 가장 적합한 신뢰성 평가 모델의 선택 문제는 매우 중요한 문제이다.

지금까지 많은 소프트웨어 신뢰성 성장 모델(SRGMs : software reliability growth models)이 제안되었지만 모든 소프트웨어의 다양한 환경에 적합한 모델이 아직 연구되지 못한 실정이다[2, 3]. 이와 같이 제안된 다양한 모델들에 대해 Yamada[4], Wood[5]와 Huang et al.[6]에서 요약하여 제시하고 있다. 일반적으로 발견되는 소프트웨어의 누적 고장 수에 대한 모델은 Goel-Okumoto의 지수적 형태 모델을 적용하고 있으며, 성장곡선(Growth Curve)에 기반한 모델들도 또한 한 부류를 차지하고 있다. 성장곡선에 기반한 소프트웨어 신뢰성 추정 모델들은 지연된 S자형, 굴절 S자형, Gompertz S자형, 로지스틱 S자형, Weibull S자형, Rayleigh S자형 등 다양한 형태가 존재한다. 일반적인 생물체의 성장현상은 대칭 로지스틱과 일반화된 로지스틱 S자형이, 기

* 정 회 원 : 경상대학교 대학원 컴퓨터학과

** 정 회 원 : 국립 원주대학 여성교양과 교수

논문접수 : 2004년 6월 9일, 심사완료 : 2004년 8월 27일

솔루션 분야에는 Pearl 또는 Gompertz 곡선이 주로 적용되고 있다. Pearl 곡선은 인구예측분야에 많이 적용하고 있으며, 최대 시간의 중간 지점에서 곡선의 상·하가 대칭이 되는 반면에 Gompertz 곡선은 대칭이 되지 않는 특징을 갖고 있다. 현실적으로 소프트웨어의 시험 중에 발견되는 고장의 수는 대칭이 되지 않기 때문에 소프트웨어 신뢰성 추정에는 Gompertz 성장곡선이 적용되고 있다.

Gompertz 성장곡선은 대수형(Logarithmic Type)과 로지스틱형(Logistic Type)이 있으며, Gompertz 성장곡선에 기반한 신뢰성 성장 모델에는 Yamada[4], Shigeru[7], Mitsuhashi[8], Xie[9], Virene[10]와 Kececioglu[11] 등이 있다. 이들 모델들은 모두 대수형 Gompertz 성장곡선에 기반하고 있다. 대수형 Gompertz 성장곡선에 기반한 신뢰성 성장 모델의 경우 모수 추정이 어렵다는 문제가 있는 것으로 알려져 있다.

본 논문은 로지스틱형 Gompertz 성장곡선에 기반한 소프트웨어 신뢰성 추정 모델을 제안한다. 2장에서는 성장곡선과 Gompertz 성장곡선에 기반한 기존 소프트웨어 신뢰성 성장모델의 문제점을 고찰해 본다. 3장에서는 로지스틱형 Gompertz 성장곡선에 기반한 신뢰성 성장모델을 제안하고 모델의 적합성을 검증한다.

2. 소프트웨어 신뢰성 성장 모델

2.1 소프트웨어 신뢰성

$N(t)$ 를 시간 t 까지 검출된 소프트웨어의 누적 고장 수, $m(t)$ 를 기대치를 나타내는 평균값 함수(MVF, Mean Value Function)로 시간 $(0, t]$ 에서 발견이 기대되는 누적 고장 수(Cumulative Number of Failures), $\lambda(t)$ 를 강도 함수(Intensity Function)로 시간 t 에서의 순간 결함 검출율이라 하자. 가장 널리 사용되는 비동차 포아송 프로세스(NHPP, Non-Homogeneous Poisson Process)모델은 누적 고장 수를 확률변수 $N(t)$ 로 모델링하고 $N(t)$ 가 식 (1)의 포아송 분포를 따른다고 가정한다.

$$P_r\{N(t) = n\} = \frac{\{m(t)\}^n}{n!} e^{-m(t)}, \quad n = 1, 2, \dots \quad (1)$$

여기서 $m(t) = \int_0^t \lambda(x)dx$ 이다. $a \equiv m(\infty)$ 를 초기 예상 결함 수 또는 소프트웨어 시스템에 잠재하고 있는 총 결함 수로 가정하면 시간 t 에서 시스템에 남아 있는 기대 결함 수 $n(t)$ 는 $a - m(t)$ 로 얻어진다. 시간 t 까지 시험이 진행되고 $(t, t+s], (s \geq 0)$ 시간 간격에서 고장이 발생되지 않을 확률

인 소프트웨어 신뢰성은 식 (2)로 얻어진다.

$$R(t|s) = e^{-[m(t) - m(t+s)]}, \quad t \geq 0, s \geq 0 \quad (2)$$

소프트웨어 신뢰성을 실제로 평가하기 위해서는 평균 값 함수 $m(t)$ 를 정의하는 것이 필요하다. 기존 NHPP 모델들은 서로 다른 소프트웨어 시험환경에 대한 $m(t)$ 를 제안하는 것으로 생각할 수 있다. Yamada[4], Wood[5]와 Huang et al.[6]에서 제시된 기존 NHPP 모델들의 평균값 함수를 <표 1>에 요약하였다.

<표 1> NHPP 모델 형태

형 태	NHPP 모델
G-O Exponential (Concave)	$m(t) = a(1 - e^{-bt})$
Modified Exponential (Two-error Type)	$m_p(t) = a \sum_{i=1}^2 P_i (1 - e^{-b_i t})$
Yamada Delayed S-shaped	$m(t) = a[1 - (1 + bt)e^{-bt}]$
Inflection S-shaped, (Hossain-Dahiya/G-O S-shaped)	$m(t) = \frac{a(1 - e^{-bt})}{1 + ce^{-bt}}$
Logarithmic Poisson Execution Time Model	$m(t) = \frac{1}{\theta} \ln(\lambda_0 \theta t + 1)$
Weibull Testing-Effort Dependent	$m(t) = a[1 - e^{-ba(1 - e^{-at})}]$
Testing-Domain Dependent	$m(t) = a[1 - \frac{1}{v-b}(ve^{-bt} - be^{-vt})]$
G-O S-shaped	$m(t) = a[1 - (1 + bt)e^{-bt}]$
Gompertz Growth Curve (S-shaped)	$m(t) = ab^{e^t}$
Logistic Growth Curve (S-shaped)	$m(t) = \frac{a}{(1 + be^{-ct})}$
Weibull S-shaped (Generalized Goel NHPP)	$m(t) = a(1 - e^{-bt^c})$
Weibull more S-shaped	$m(t) = a[1 - (1 + bt^c)e^{-bt^c}]$
Yamada Exponential Concave	$m(t) = a[1 - e^{-b(1 - e^{-at})}]$
Yamada Rayleigh S-shaped	$m(t) = a[1 - e^{-b(1 - e^{-t^2})}]$
Modified Duane	$m(t) = a[1 - (\frac{b}{b+t})^c]$

2.2 성장곡선

일반적으로 결함 발견 현상에서 누적 고장 수는 초기에는 점진적으로 증가하다가 중간 구간에서 보다 빠르게 증가하고 한계점으로 도달하면서 성장 속도가 느려지며 일정한 시간 구간 후에는 수렴하는 S자 패턴을 보이는 것으로

알려져 있다[12]. 이는 생명체의 성장과정과 유사하며 성장곡선을 기반으로 하는 SRGM이 개발되었다.

대표적인 성장곡선으로 대칭 로지스틱 성장 곡선, 일반화된 로지스틱 곡선, Pearl 곡선과 Gompertz 곡선 등이 있다. a 를 $m(t)$ 의 상한 점근선, t 를 시간, b 를 곡선의 위치를 결정하는 하한 점근선, c 를 곡선의 모양을 결정하는 값인 성장률이라 하자. 대칭 로지스틱 성장곡선은 생물체의 성장 패턴을 모형화하는데 많이 사용되고 있으며 식 (3)으로 표현된다. 이 함수는 중간지점인 $a/2$ 를 중심으로 상호 대칭이 되며, $a/2$ 지점부터 성장률이 감소하기 시작하지만 최대 값 점근선인 a 에 도달할 때까지 계속 성장한다.

$$m(t) = \frac{a}{(1 + e^{-b+ct})} \quad (3)$$

일반화된 로지스틱 곡선은 Richard 곡선이라고도 불리며, 성장을 모형화하는데 널리 사용되고 있으며 식 (4)로 표현된다. 여기서 t_1 는 최대 성장이 발생하는 점을 결정한다.

$$m(t) = b \frac{a}{(1 + t_1 e^{-c(t-M)})^{1/t_1}} \quad (4)$$

Pearl 곡선은 미국 통계학자인 Raymond Pearl[13]이 제안한 것으로 인구예측에 사용하였으며, 로지스틱 곡선으로 잘 알려져 있다. 표준화된 Pearl 곡선은 식 (5)의 형태를 취한다. 이 함수는 $-\infty$ 시간에서는 초기값 0을, $+\infty$ 시간에서는 a 값을 가진다. 만약 초기 값이 0가 아닌 오프셋(Offset) 값을 갖고 있으면 상수로서 위 공식에 더해진다. 즉 인구예측에서 어느 한 도시에 기존에 거주한 인구가 오프셋 값이 되며, 이 인구가 어떻게 성장하는가를 표현하는 모델이 된다. $t = \ln(b)/c$ 시점에서 $m(t) = a/2$ 일 때 곡선이 굴곡된다. 따라서, 이 곡선은 $m(t) = a/2$ 가 중간지점이 되므로 이 지점을 중심으로 위쪽과 아래쪽이 대칭이 된다.

$$m(t) = \frac{a}{1 + be^{-ct}} \quad (5)$$

Gompertz 곡선은 영국의 공중인이자 수학자인 Benjamin Gompertz[14]에 의해 제안되었으며, 식 (6) 또는 식 (7)로 표현된다. 식 (6)을 로지스틱형, 식 (7)을 대수형 Gompertz 성장곡선이라 부른다. 이 곡선은 $-\infty$ 시간에서는 초기 값 0을, $+\infty$ 시간에서는 a 값을 가진다는 점에서 Pearl 곡선과 닮았지만, 곡선이 굴절되는 지점은 $t = \ln(b)/c$ 시점으로 비대칭이 되는 특징을 갖고 있다.

$$m(t) = ae^{-be^{-ct}} \quad (6)$$

$$m(t) = ab^{c^t} \quad (7)$$

성장곡선을 이용하기 위해서는 다음 가정에 기반하여야 한다.

- ① 선택된 성장곡선이 수집된 과거 이력 데이터에 적합한 것이어야 한다.
- ② 수집된 이력 데이터는 선택된 성장 곡선 함수의 계수를 적절히 결정하게 해주어야 한다.

성장곡선을 이용하여 미래의 행위를 예측하는 과정은 다음과 같다.

- ① 성장 곡선 모델이 주어진 데이터에 적절한지 결정한다.
- ② 적절한 성장곡선 모델을 선택한다(Gompertz 또는 Pearl 곡선).
- ③ MSE(Mean Squared Error)를 최소화하거나 변수변환으로 얻는 선형회귀 방정식을 이용하여 계수를 결정한다.

성장곡선 중에서 기술 예측에 가장 많이 적용하고 있는 성장곡선으로는 Pearl 곡선과 Gompertz 곡선이다. 이 두 곡선 중에서 적절한 것을 선택하는 기준은 다음 조건에 기반한다.

- ① 얻고자 하는 데이터의 성장 곡선이 대칭인가? 만약 대칭이 될 경우에는 Pearl 곡선을, 비대칭일 경우에는 Gompertz 곡선을 선택한다.
- ② 성장과정에서 이미 오프셋 요인이 있는가? 만약 오프셋 요인이 있는 경우에는 Pearl 곡선을, 오프셋 요인이 없으면 Gompertz 곡선을 선택한다.

소프트웨어의 시험 시간에 따른 누적 고장 수 그래프는 일반적으로 $a/2$ 지점을 중심으로 대칭이 되지 않으며, 소프트웨어 시험을 시작하는 시점 ($t=0$)에서의 발견된 고장은 없기 때문에 오프셋 값이 0가 된다. 따라서, 소프트웨어 신뢰성 추정 분야에는 Gompertz 성장곡선이 다른 성장곡선보다 더 적합하다고 할 수 있다.

3. 로지스틱형 Gompertz 성장곡선 SRGM

3.1 모델 제안

기존에 제안된 성장곡선 기반 SRGM은 Yamada[4], Wood[5], Shigeru[7], Mitsuhashi[8], Xie[9], Virene[10]와 Kececioglu[11] 등에서 거론되고 있으며, 식 (8)과 같이 모두 대수형이나

오프셋 값을 갖고 있는 식 (9)의 수정된 대수형을 적용하고 있다.

$$m(t) = ab^{ct}, 0 < a, 0 < b < 1, 0 < c < 1 \quad (8)$$

$$m(t) = d + ab^{ct}, 0 < a + d \leq 1, 0 < b < 1, 0 < c < 1, t \geq 0 \quad (9)$$

이 부류의 모델들을 적용하기 위해 모수 a, b, c 를 추정하는 방법은 최소자승법(LMS, Least Mean Squares)이나 최우추정법(MLE, Maximum Likelihood Estimation) 등 다양한 방법을 이용할 수 있다. 그러나 이 모델들은 비선형으로 초기 값을 적절히 설정하는 어려움이 있다. 비선형 회귀 분석방법의 초기 값 선택은 쉬운 작업이 아니다[15]. 초기 값의 나쁜 선택은 많은 반복을 가진 장황한 계산이 요구되며, 발산 또는 지역 최소점(Local Minimum)으로 수렴하는 경우가 발생할 수 있다. 만약 다수의 최소점들이 존재한다면 좋은 초기 값은 적은 반복으로 빠른 계산을 수행할 수 있으며, 전역 최소점(Global Minimum)에 도달하여 적절한 해법을 찾을 수 있다. 또 다른 방법은 식 (10)과(11)과 같이 변수변환으로 선형회귀분석을 수행한다. 이 방법도 c' 의 값으로부터 모수 c 의 값을 다시 얻어야 하는 어려움이 있다.

$$\log m(t) = \log a + c' \log b \quad (10)$$

$$\ln[m(t) - d] = \ln a + c' \ln b \quad (11)$$

회귀 모수에 대한 초기값을 얻기 위해 다양한 방법들이 개발되었으며 이들 방법 중에서 Virene 방법[10]이 가장 단순하여 선호된다.

본 장에서는 기존의 대수형이 아닌 로지스틱(Logistic), S자형(S-shaped) 또는 시그모이드(Sigmoidal) 곡선으로 불리는 식 (12) 모델을 소프트웨어 신뢰성을 추정하는데 적용한다.

$$m(t) = ae^{-be^{-ct}}, 0 < a, 0 < b, 0 < c \quad (12)$$

이 모델은 대수적 Gompertz 성장곡선 SRGM 보다 간단한 방법으로 모수를 추정할 수 있는 장점을 갖고 있다. 즉, 식 (13)과 같이 두 번의 자연로그 변수변환을 거치면 선형 방정식이 얻어지며, 상용 소프트웨어 도구를 이용하여 선형 회귀분석을 수행하면 모수들의 값을 구할 수 있다.

$$\ln \left[- \ln \left(\frac{m(t)}{a} \right) \right] = \ln b - ct \quad (13)$$

본 모델의 모수를 추정하는데 있어 한가지 어려운 점은 모수 a 의 적절한 값을 결정하는 점이다. 소프트웨어의 전

체 고장 수를 나타내는 a 는 신뢰성 성장 모델에서 가장 중요한 모수이므로 이론적인 근거 없이 시행착오적 방법으로 추정할 수는 없다. 따라서, 최소 자승법이나 최우 추정법으로 추정할 수도 있을 것이다. 본 논문은 이의 대안으로 누적 고장 수를 C_i 라 할 때, $\ln \left[- \ln \left(\frac{C_i}{a} \right) \right]$ 와 시간 t 사이의 상관계수가 최대가 되도록 데이터를 가급적 직선 형태로 만드는 방법을 적용하였다. 이 방법이 식 (13)을 이용하는 가장 합리적인 추정법이라 할 수 있다.

3.2 실험에 적용된 데이터

모델을 평가하기 위해 <표 2>와 같이 13개의 데이터를 적용한다. 기존의 신뢰성 추정 모델들의 특징은 고장 수집 시간(또는 시험 시간) 간격이 일정한 경우에 한해서만 모델들이 적용될 수 있는 단점을 갖고 있다. 그러나 실제적으로는 고장 수집 시간이 일정하거나 가변적인 경우가 발생할 수 있다. 고장 수집시간이 일정한 경우는 단위시간 간격을 일정하게 하고 이 시간 간격 내에서 발견된 고장 수를 기록하는 방법이다. 반면에 가변적인 경우는 고장이 발생한 시간 간격 데이터를 그룹화한 그룹 데이터(Grouped data) 형태로 형태는 고장 수 데이터와 동일하나 다른 점은 가변적인 시험 간격(Variable time interval of Testing) 형태를 취한다는 점이다.

<표 2> 실험에 적용된 데이터

데이터명	고장 수집 시간 간격		참고 문헌	고장 수	데이터 크기
	일 정	가 변 적			
Data 1	○		[16]	27	14
Data 2	○		[17]	136	25
Data 3	○		[18]	46	21
Data 4		○	[18]	328	17
Data 5		○	[18]	279	10
Data 6	○		[19]	3,207	13
Data 7	○		[20]	535	109
Data 8	○		[21]	481	111
Data 9	○		[21]	55	169
Data 10	○		[21]	198	16
Data 14	○		[20]	266	46
System T1		○	[1]	136	21
System T38		○	[1]	32	11

본 논문에서 모델을 평가하기 위해 수집된 데이터를 살펴보면 고장 수집 시간 간격이 일정한 경우가 9개, 고장 수

집 시간 간격이 가변적인 경우가 4개 데이터이다. 따라서, 제안된 모델이 고장 수집 시간 간격이 일정한 데이터는 물론 가변적인 경우에도 적합하지 검증한다.

3.3 모수 추정 결과 얻어진 모델

대수형 Gompertz 성장곡선 모델의 모수 추정은 Virene 방법[10]을, 로지스틱형 Gompertz 성장곡선 모델에 대해서는 식 (13)의 변수변환을 통한 선형회귀 방법을 적용하였다.

Virene[10]의 방법은 누적 고장 수 데이터의 개수를 m 이라 할 때, 누적 고장 수에 자연 로그를 취한 $\ln C_i, i=1, 2, \dots, m$ 을 얻는다. 이 데이터를 $m=3n$ 과 같이 세 부분으로 분할한다.

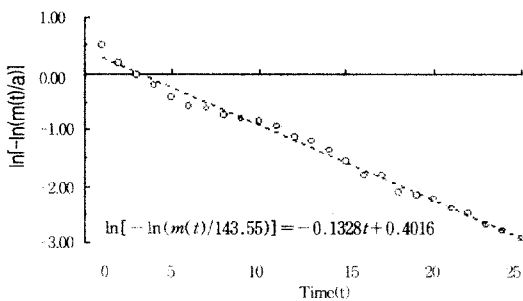
분할된 데이터 각각에 대해 $\ln C_i$ 를 누적시킨 $S_1 = \sum_{i=0}^{n-1} \ln C_i, S_2 = \sum_{i=n}^{2n-1} \ln C_i, S_3 = \sum_{i=2n}^{m-1} \ln C_i$ 을 얻는다. 이로부터 모델의 모수를 다음과 같이 얻는다.

$$c = \frac{S_3 - S_2}{S_2 - S_1}, a = e^{\frac{1}{n} \left(S_1 + \frac{S_2 - S_1}{1 - c^n} \right)}, b = e^{\frac{(S_2 - S_1)(c - 1)}{(1 - c^n)^2}}$$

이와 같은 방법으로 Data 2에 대해 식 (8)의 대수형 Gompertz 성장곡선 모델의 모수는 $\hat{a} = 149.1683, \hat{b} = 0.2715, \hat{c} = 0.8902$ 를 얻었다. 따라서, 누적 고장 수는 $m(t) = 149.1683 \cdot 0.2715^{0.8902t}$ 모델로 추정될 수 있다.

로지스틱형 Gompertz 성장곡선 모델에 대해서는 식 (12)에 자연로그를 2번 취한 식 (13)에서 상관계수가 최대가 되었을 때의 a 값은 143.55를 얻었다. (그림 1)의 데이터로부터 선형 회귀분석을 수행한 결과 식 (14)를 얻었다.

$$\ln[-\ln(m(t)/143.55)] = -0.1328t + 0.4016 \quad (14)$$



(그림 1) 로지스틱형 Gompertz 성장곡선의 모수 추정

식 (14)에서 $\ln(b) = 0.4016$ 로 부터 $b = e^{\ln b} = 1.4942$ 을 얻을 수 있다. 그러므로 누적 고장 수에 대한 추정 값은 $m(t) = 143.55e^{-1.4942e^{-0.1328t}}$ 모델로 계산된다.

이와 동일한 방법으로 13개 데이터에 대해 누적 고장 수를 추정하는 제안된 로지스틱형 Gompertz 성장 곡선 모델과 대수형 Gompertz 성장 곡선 모델에 대한 결과는 <표 3>에 제시하였다.

<표 3> 누적 고장 수 추정 모델

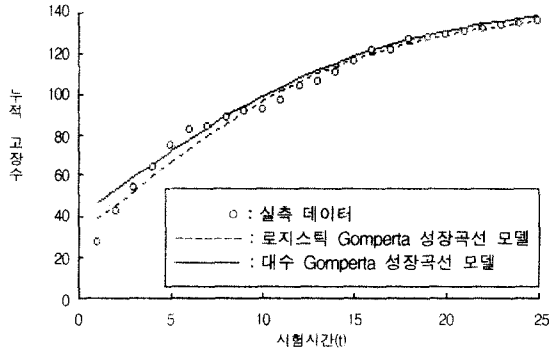
데이터 명	누적 고장 수 추정 Gompertz 성장곡선 모델	
	$m(t) = ae^{-be^{-ct}}$ (로지스틱형)	$m(t) = ab^{c^t}$ (대수형)
Data 1	$m(t) = 53.5e^{-2.4337e^{-0.0944t}}$	$m(t) = 32.4928 \cdot 0.0884^{0.8575^t}$
Data 2	$m(t) = 143.55e^{-1.4942e^{-0.1328t}}$	$m(t) = 149.1683 \cdot 0.2715^{0.8902^t}$
Data 3	$m(t) = 61.5e^{-4.1054e^{-0.1311t}}$	$m(t) = 62.3032 \cdot 0.0272^{0.8833^t}$
Data 4	$m(t) = 400.08e^{-2.8471e^{-0.1924t}}$	$m(t) = 441.8575 \cdot 0.0645^{0.8655^t}$
Data 5	$m(t) = 279.5e^{-9.2442e^{-0.6560t}}$	$m(t) = 292.1133 \cdot 0.0594^{0.6579^t}$
Data 6	$m(t) = 3485e^{-3.1664e^{-0.7964t}}$	$m(t) = 5234.1953 \cdot 0.0866^{0.8435^t}$
Data 7	$m(t) = 541.52e^{-3.9527e^{-0.1033t}}$	$m(t) = 545.9833 \cdot 0.0256^{0.9505^t}$
Data 8	$m(t) = 483.2e^{-2.8795e^{-0.1517t}}$	$m(t) = 478.9724 \cdot 0.0210^{0.9320^t}$
Data 9	$m(t) = 68.18e^{-4.1351e^{-0.1417t}}$	$m(t) = 68.2210 \cdot 0.0173^{0.9827^t}$
Data 10	$m(t) = 214.55e^{-5.0993e^{-0.2756t}}$	$m(t) = 272.5491 \cdot 0.0302^{0.8360^t}$
Data 14	$m(t) = 280.1e^{-3.3188e^{-0.2905t}}$	$m(t) = 334.8372 \cdot 0.0569^{0.9384^t}$
System T1	$m(t) = 138.65e^{-2.6988e^{-0.1166t}}$	$m(t) = 317.1547 \cdot 0.0021^{0.8972^t}$
System T38	$m(t) = 35.5e^{-2.7433e^{-0.1972t}}$	$m(t) = 30.0977 \cdot 0.1192^{0.6664^t}$

3.4 모델 평가

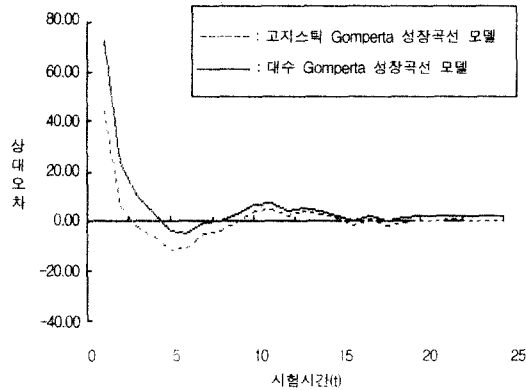
<표 3>과 같이 구해진 모델이 Data 2에 대해 실측 누적 고장 수를 얼마나 잘 추정하는지를 살펴보자. 실측 데이터에 대해 로지스틱과 대수형 Gompertz 성장곡선 모델의 추정 결과를(그림 2)에 제시하였다. 시험 시간이 초기 부분에서는 두 모델 모두 적절히 표현하지 못하고 있으며, 특히 대수형 Gompertz 성장곡선 모델이 보다 나쁜 결과를 나타내고 있다.

모델을 평가하는 기준으로 소프트웨어 신뢰성 분야 또는 추정 분야에서는 MMRE(Mean Magnitude Relative Error)를 대부분 채택하고 있다. 따라서 본 논문에서도 MMRE를 적용한다. MMRE는 다음과 같이 측정된다. 상대오차(RE, Relative Error) = (추정치 - 실측치)/실측치, MRE(Magnitude of RE) = |RE|, MMRE(Mean MRE) = $100/n * \sum_{i=1}^n MRE_i$. MMRE가 작은 값이면 평균적으로 좋은 모델임을 알 수 있다. Data 2에 대해 <표 3>의 로지스틱형과 대수형 Gom-

pertz 성장곡선 모델의 상대오차를 (그림 3)에 제시하였다. 그림에서 로지스틱형이 대수형 Gompertz 성장곡선 모델에 비해 상대오차가 적음을 알 수 있다.



(그림 2) 로지스틱형과 대수형 Gompertz 성장곡선 모델의 누적 고장 수 추정 결과(Data 2)



(그림 3) 로지스틱형과 대수형 Gompertz 성장곡선 모델 상대오차 비교(Data 2)

13개 데이터에 대해 제안된 로지스틱형과 기존의 대수형 Gompertz 성장곡선 모델을 이용하여 추정된 데이터의 최대값과 MMRE는 <표 4>에 제시하였다. 표에서 시간간격이 일정한 경우의 9개 데이터 중 8개 데이터에서 제안된 로지스틱형 Gompertz 성장 곡선 모델이 성능이 좋으며, 단지 Data 8에서만 대수형 Gompertz 성장 곡선 모델이 좋음을 보였다.

그러나 9개 데이터 모두에서 데이터의 최대 값을 추정할 수 있는지를 살펴본 결과 대수형 Gompertz 성장 곡선 모델은 모두 최대 값을 정확히 추정하지 못하는 반면 로지스틱형 Gompertz 성장 곡선 모델은 모두 최대 값을 정확히 추정하는 결과를 보이고 있다.

시간 간격이 가변적인 4개 데이터에 대해서도 4개중 3개

데이터에서 로지스틱형 Gompertz 성장 곡선 모델의 MMRE가 대수형 Gompertz 성장 곡선 모델보다 좋은 결과를 보였다. 또한, 최대 값 추정 성능면에서 로지스틱형 Gompertz 성장 곡선 모델이 4개 데이터 모두에서 거의 정확히 추정하는 능력을 갖고 있음을 알 수 있다. 결론적으로, Gompertz 성장 곡선을 이용한 소프트웨어 신뢰성 추정 분야에서 기존의 대수형 모델보다는 로지스틱형 모델이 모수 추정 방법이 간단하고 주어진 데이터의 최대값도 보다 정확히 추정할 수 있으며 또한 모델의 성능도 좋은 장점을 갖고 있음을 알 수 있다.

<표 4> Gompertz 성장 곡선 모델의 성능

고장 수집 시간 간격	데이터명	최대값	모델 추정 성능			
			$m(t) = ae^{-be^{-t}}$ (로지스틱형)		$m(t) = ab^{c^t}$ (대수형)	
			최대값	MMRE	최대값	MMRE
일정	Data 1	27	27.0052	8.0098	24.5101	13.5472
	Data 2	136	136.0012	4.5399	138.9187	6.6373
	Data 3	46	45.9905	7.0823	47.7477	18.4109
	Data 6	3,207	3207.022	5.9339	3902.142	22.5732
	Data 7	535	535.0003	10.5597	538.1250	11.8726
	Data 8	481	480.9547	19.6949	478.2250	8.3077
	Data 9	55	55.0003	15.7630	55.1651	16.2217
	Data 10	198	198.0097	11.5956	223.2610	26.8037
	Data 14	266	266.0019	41.3624	287.1553	58.9450
가변적	Data 4	328	328.0066	13.8938	439.3193	107.3673
	Data 5	279	279.0075	33.7223	288.5648	44.4128
	Ststem T1	136	135.9951	89.8568	213.1607	70.4292
	System T38	32	31.9993	74.3348	30.0977	495.0542

4. 결론 및 향후 연구과제

지난 30여 년 동안 소프트웨어의 신뢰성을 추정하기 위한 다양한 시험환경을 고려한 많은 모델들이 제안되었다. 이들 모델들 중에서 성장곡선을 이용한 SRGM들도 한 부류를 차지하고 있다. 본 논문은 이들 SRGM 들 중에서 Gompertz 성장곡선 모델에 대해 고찰하였다. Gompertz 성장곡선 모델은 대수형과 로지스틱형으로 분류된다. Gompertz 성장곡선에 기반한 기존의 모든 SRGM들은 대수형을 도입

하고 있다. 그러나 이 모델은 모수 추정의 어려움과 주어진 데이터의 최대 값을 적절히 표현하지 못하는 단점을 갖고 있다.

본 논문에서는 기존의 대수형 Gompertz 성장곡선 모델의 단점을 보완한 로지스틱형 Gompertz 성장곡선 모델을 제안하였다. 제안된 모델의 적합성을 검증하기 위해 실제 수집된 13개의 소프트웨어 고장 데이터를 적용하였다. 기존의 모델들은 고장 수집 시간이 일정한 경우에 적합한 모델들이다. 이러한 단점을 극복하기 위해 본 논문에서는 고장 수집시간이 일정한 경우와 가변적인 경우도 고려하였다. 제안된 모델은 고장 수집 시간이 일정한 경우와 가변적인 경우 모두 대부분의 데이터에서 MMRE와 데이터의 최대 값을 적절히 표현하는 성능을 보임을 입증하였다.

본 논문에서는 고장 수 데이터에 한정하여 로지스틱형 Gompertz 성장곡선 모델의 적합성을 검증하였다. 그러나 수집되는 고장 데이터의 형태에는 고장 시간 데이터도 존재한다. 고장시간 데이터는 일반적으로 고장 수 데이터와 다른 분포 형태를 따른다. 그럼에도 불구하고 고장 시간 데이터에도 로지스틱형 Gompertz 성장곡선 모델을 적용할 수 있는지 연구를 수행할 것이다.

참 고 문 헌

- [1] M. R. Lyu, "Handbook of Software Reliability Engineering," IEEE Computer Society Press, 1996.
- [2] R. H. Hou and S. Y. Kuo, "Applying Various Learning Curves to Hypergeometric Distribution Software Reliability Growth Model," IEEE, 1994.
- [3] F. Poptentiu and D. N. Boros, "Software Reliability Growth Supermodels," Microelectron. Reliab, Vol.36, No.4, pp.485-491, 1996.
- [4] S. Yamada, "Software Reliability Models and Their Applications : A Survey," International Seminar on Software Reliability of Man-Machine Systems - Theories, Methods and Information Systems Applications, Kyoto University, Kyoto, Japan, 2000.
- [5] A. Wood, "Predicting Software Reliability," IEEE Computer, Vol.29, No.11, pp.69-77, 1996.
- [6] C. Y. Huang, M. R. Lyu and S. Y. Kuo, "Unified Scheme of Some Nonhomogeneous Poisson Process Models for Software Reliability Estimation," IEEE Trans. on Software Eng., Vol.29, No.3, 2003.
- [7] Y. Shigeru, "A Stochastic Software Reliability Growth Model with Gompertz Curve," IPSJ Journal, Vol.33, No.07, 1992.
- [8] T. Mitsuhashi, "A Method of Software Quality Evaluation in Japanese," JUSE Press, Tokyo, 1981.
- [9] M. Xie, "Software Reliability Modeling," World Scientific Publishing, 1991.
- [10] E. P. Virene, "Reliability Growth and Its Upper Limit," Proceedings of Annual Symposium on Reliability, pp. 265-270, 1968.
- [11] J. S. Kececioglu, B. Dimitri, and V. Pantelis, "Modified Gompertz," Proceedings of the Annual Reliability and Maintainability Symposium, 1994.
- [12] C. Henry, "The Growth Curve," <http://www.anzpug.org/jsp/index.jsp>, PRIMAVERA Users Groups, Technology and Operations Management, California Polytechnic and State University,
- [13] R. Pearl, "The Biology of Population Growth," New York: Knopf, 1978.
- [14] B. Gompertz, "On The Nature of The Function Expressive of The Law of Human Mortality, and on a New Mode of Determining the Value of Life Contingencies," Phil. Trans. Roy. Soc. London. Vol.123, pp.513-585, 1832.
- [15] Weibull.com, "Software Reliability Growth Model," <http://www.weibull.com/relgrowthwebcontents.html>
- [16] K. Matsumoto, T. Inoue, T. Kikuno, and K. Torii, "Experimental Evaluation of Software Reliability Growth Models," Proc. IEEE Conf. FTCS-18, pp.148-153, 1988.
- [17] J. D. Musa, A. Lannino, and K. Okumoto, "Software Reliability : Measurement, Prediction, Application," McGraw-Hill, New York, 1987.
- [18] M. Ohba, "Software Reliability Analysis Models," IBM H. Res. Develop., Vol.28, pp.428-443, 1984.
- [19] M. L. Shooman, "Probabilistic Models for Software Reliability Prediction," Statistical Computer Performance Evaluation, New York Academic, pp.485-502, 1972.
- [20] Y. Thoma, H. Yamano, M. Ohba, and R. Jacoby, "Parameter Estimation of the Hyper-Geometric Distribution Model for Real Test/Debug Data," Dept. Computer Science, Tokyo Inst. Tech., Tech. REP. 901002, 1990.
- [21] Y. Thoma, K. Tokunaga, S. Nagase, and Y. Murata, "Structural Approach to the Estimation of the Number of Residual Software Faults Based on the Hyper-Geometric Distribution," IEEE Trans. on Software Eng., Vol.15, pp.345-355, 1989.

박석규



e-mail : sgpark@gw.ac.kr

1988년~2001년 진주산업대학교 전산실장

1992년 경남대학교 대학원 컴퓨터공학과 석사

2001년~현재 도립 강원전문대학 컴퓨터 응용과 조교수

2002년~현재 경상대학교 대학원 컴퓨터학과 박사과정

관심분야 : 소프트웨어 신뢰성, 시스템 분석 및 설계, 멀티미디어

이상운



e-mail : sulee@sky.wonju.ac.kr

1983년~1987년 한국항공대학교 항공전자 공학과 학사

1995년~1997년 경상대학교 컴퓨터과학과 석사

1998년~2001년 경상대학교 컴퓨터과학과 박사

1992년~2003년 국방품질관리소 항공전자장비 및 소프트웨어 품질보증 담당

2003년 도립 강원전문대학 컴퓨터응용과 전임강사

2004년~현재 국립 원주대학 여성교양과 전임강사

관심분야 : 프로젝트 관리, 소프트웨어 개발 방법론, 소프트웨어 Metrics, 소프트웨어 시험, 소프트웨어 신뢰성, 아키텍처, 신경망, 뉴로-퍼지