

傾向曲線に基づいた ソフトウェア信頼性モデルに対するパラメータ推定

岡村 寛之[†] 古村 仁志[†] 土肥 正[†]

ソフトウェアのテスト工程において検出されるバグの累積総数を予測する方法の1つに、ゴンベルツ曲線やロジスティック曲線などの傾向曲線を基本とした回帰モデルがある。しかしながら、本来は確率的な計数過程として記述されるべきこれらのデータに対して、確定的な曲線とその観測誤差に基づいた回帰モデルを適用したとしても、バグ総数の推定精度が徐々に向上するとは限らない。一方、ソフトウェアの信頼度成長現象を柔軟に表現する計数過程として、非同次ポアソン過程 (NHPP) を用いた確率モデルが提案されているが、そのいくつかはパラメータ推定に関する厳密な議論がなされていない。そこで本論文では、ゴンベルツ曲線やロジスティック曲線などの回帰モデルと対応付けされるような NHPP を用いて、ソフトウェア信頼性モデルに対するパラメータ推定問題を再考する。特に、最尤推定値を EM アルゴリズムを用いて効率良く算出する手法の提案を行う。さらに、実際のテスト工程で観測されたデバッグデータを用いて、回帰モデルならびに NHPP モデルに対するパラメータ推定手法の比較を行うことで、提案する統計モデルおよび推定アルゴリズムの優位性を示す。

Parameter Estimation for Trend-curve-based Software Reliability Models

HIROYUKI OKAMURA,[†] HITOSHI FURUMURA[†] and TADASHI DOHI[†]

The regression model is one of the well-known methods to predict the cumulative number of faults detected in software development phase. In the regression models, it is assumed that the cumulative number of detected faults is described by a non-linear curve such as a Gompertz curve or a logistic curve. The regression model is essentially based on the squared errors between the observed data and the trend curve. However, the direct application of the regression model to the stochastic counting processes may cause the problem that the accuracy of prediction decreases. On the other hand, software reliability models based on non-homogeneous Poisson processes (NHPPs) are well known to represent various trend curves including the Gompertz and logistic curves. However, no effective parameter estimation method for these NHPP-based models has been developed. In this paper, we revisit the NHPP-based software reliability models with the mean value functions which draw the Gompertz and logistic curves. Especially, the estimation algorithm to compute the maximum likelihood estimates efficiently is developed by applying the EM algorithm. Furthermore, comparing between the regression models and the NHPP-based software reliability models, we show the significance of the NHPP-based models in the viewpoint of predictive accuracy.

1. はじめに

現在のように、コンピュータシステムとそれらを結ぶネットワークに強く依存した社会は、システムの障害に対する大きなリスクを負っているといっても過言ではない。このようなシステムを制御するソフトウェアには、非常に高い品質が要求される。ソフトウェアの品質特性としては、機能性、信頼性、使用性、効率性、保全性、移植性の6項目が標準化されているが、

システム障害を回避するための品質特性として「信頼性」は最も重要な要素として位置づけられる。しかしながら、ソフトウェアの開発は人的作業に基づいて行われているため、設計ミスなどによるプログラム中へのバグの混入は避けられない。このバグを取り除くために、開発工程の最終段階では、実行テストを繰り返し実施する必要があるが、実際にテスト段階において費やされるコストや労力が開発コストの大部分を占めているのが現状である。さらに、どれだけテストを行い、どれだけバグを修正・除去すればよいかという指標を定めることができなければ、多くの開発コストを無駄に消費することになる。つまり、定量的なソフトウェ

[†] 広島大学大学院工学研究科

Graduate School of Engineering, Hiroshima University

ア信頼性の評価基準を定めることは、現在もお重要な課題となっている。ソフトウェア信頼性の定量的評価尺度で、最も一般的なものとしてソフトウェア信頼度があげられる。これは、「ソフトウェアがある環境のもとで、一定期間中に障害を発生することなく動作する確率」として定義される。このように、ソフトウェア信頼度を評価するためには、必然的にソフトウェアバグの検出頻度を確率モデルに基づいて表現しなければならない。

一方、わが国では、ソフトウェア信頼度の概念が提案される以前から、ソフトウェアのテスト工程において発見されるバグの累積総数を予測する手法として、ゴンベルツ曲線やロジスティック曲線などの傾向曲線に基づいた回帰モデルが用いられており、今もなお、開発現場ではそれらのモデルが適用されている^{1),2)}。しかしながら、確率的な計数過程として記述されるべきこれらバグの検出データに対して、確定的な曲線とその観測誤差に基づいて記述された回帰モデルを適用した場合、前述のソフトウェア信頼度を算出することができないばかりか、バグ総数の推定値に関する精度が低下する危険性がある。他方、ソフトウェア信頼度を評価するための代表的な数理モデルとして、非同次ポアソン過程 (NHPP) を用いたソフトウェア信頼性モデルがある。これらのモデルは比較的よくバグの検出数データに適合し、バグの出現頻度に関する平均的な傾向を表す関数を与えるだけで様々なデバッグ状況をモデル化することが可能である。山田³⁾ や Ohba⁴⁾ は、バグの検出数に関する平均的な振舞いがゴンベルツ曲線やロジスティック曲線となる NHPP モデルを提案しているが、そこではパラメータ推定に関する厳密な議論がなされていない。そこで本論文では、ゴンベルツ曲線やロジスティック曲線による回帰モデルと対応付けが可能な傾向曲線に基づいた NHPP モデルに対して、パラメータ推定問題を再考する。特に、最尤推定値を EM アルゴリズムを用いて効率良く算出する方法を提案する。さらに、実際のデバッグデータを用いて、従来の回帰モデルによるパラメータ推定手法と比較することで、提案された NHPP モデルおよび推定アルゴリズムの優位性を示す。

2. 傾向曲線モデル

ソフトウェアの開発工程におけるバグ (フォールト) の検出数は、初期段階において少なく、中期において多く、そして最終段階においては再び少なくなる傾向にある。つまり、累積バグ数は S 字形の曲線を描くことが経験的に知られている。従来から、そのような S

字形曲線で表現される累積バグ数を表現するために、ゴンベルツ曲線やロジスティック曲線などの傾向曲線に基づいた統計モデルが用いられてきた¹⁾。ここでは、これらの傾向曲線に基づいたモデルを紹介するとともに、実データが与えられた際のパラメータ推定法について言及する。

2.1 ゴンベルツ曲線モデル

ゴンベルツ曲線は、成人の死亡率を表現する数理モデルとして、1825 年に Gompertz により提案された傾向曲線である。以来、伝染病の感染者数や人口減少数の予測などのほかに、ハードウェア製品の信頼度成長現象などの記述にも頻繁に利用されてきた。ソフトウェア信頼性の分野では、テスト工程において観測される累積バグ数を表現するためによく利用されている¹⁾。テスト時刻 t (≥ 0) におけるバグの累積検出数を $M_G(t)$ とすると、ゴンベルツ曲線は

$$M_G(t) = \omega a^{b^t} \quad (1)$$

によって与えられる。ここで、 $0 < a < 1$, $0 < b < 1$ は任意のパラメータであり、 $\lim_{t \rightarrow \infty} M_G(t) = \omega$ (> 0) はソフトウェアに潜在する初期のバグ数と見なすことができる。図 1 にゴンベルツ曲線の例を示す。図から分かるように、ゴンベルツ曲線は $t = -\log[-\log a]/\log b$ で変曲点を持つ S 字形曲線となる。また、 $t = 0$ のとき値 $M_G(0) = \omega a$ をとり、 0 とはならない。これは、テスト開始以前にすでに発見されているバグが存在することを意味しており、テスト工程におけるデバッグ現象に対する表現としては至って不自然である。各パラメータ ω , a , b の値は、テスト工程において検出されたバグ数のデータを用いて、回帰分析により推定する方法が一般的である。具体的には、少ない計算労力で推定値を算出できる三点推定法⁵⁾ やゴンベルツ曲線と実データの残差平方和を最小にする最小二乗法が用いられる。ここでは、最小二乗法による推定

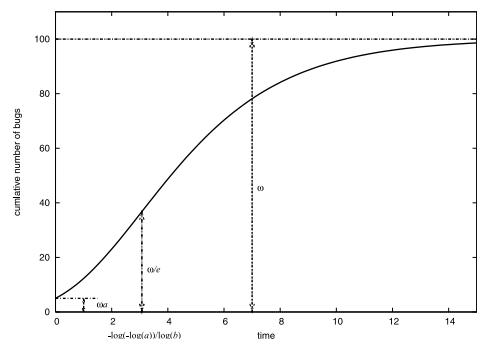


図 1 ゴンベルツ曲線 ($\omega = 100$, $a = 0.05$, $b = 0.7$)
Fig. 1 Gompertz curve ($\omega = 100$, $a = 0.05$, $b = 0.7$).

表 1 テスト工程におけるバグ数のデータ形式

Table 1 Data format of the number of bugs in testing phase.

time interval	number of software bugs
$[t_0, t_1)$	y_1
$[t_1, t_2)$	y_2
\vdots	\vdots
$[t_{K-1}, t_K)$	y_K

を紹介する.

いま, あるテスト時刻列 $0 < t_1 < \dots < t_K$ の時間間隔 $(t_{i-1}, t_i]$ ($i = 1, \dots, K$) における検出バグ数 y_i が与えられているものとする (表 1 参照). このとき, モデルパラメータ $\theta = (\omega, a, b)$ で表されるゴンベルツ曲線 $M_G(t; \theta)$ と実データとの残差平方和

$$RSS(\theta) = \sum_{i=1}^K \left\{ \sum_{j=1}^i y_j - M_G(t_i; \theta) \right\}^2 \quad (2)$$

を定義する. 回帰分析におけるパラメータの推定値は上式で定義された残差平方和 $RSS(\theta)$ を最小にする値として与えられる. 通常, これは非線形計画問題となるため, 実際のパラメータ推定値はガウス・ニュートン法などの数値解析手法を利用して算出することになる.

2.2 ロジスティック曲線モデル

ロジスティック曲線は, 1838 年に Verhulst により人口の成長を数理的に表現するモデルとして定式化され, 資源制約のある個体群成長の基本モデルとして知られている. 人口予測のモデル以外にも, 製品の需要曲線や産業の成長・発展・衰退の記述にも利用されている. ゴンベルツ曲線と同様に, ソフトウェア信頼性の分野では, テスト工程において検出される累積バグ数を表す曲線として用いられる. テスト時刻 $t (\geq 0)$ におけるバグの累積検出数を $M_L(t)$ とすると, ロジスティック曲線は

$$M_L(t) = \frac{\omega}{1 + ce^{-bt}} \quad (3)$$

によって与えられる. ロジスティック曲線においても $\lim_{t \rightarrow \infty} M_L(t) = \omega (> 0)$ は初期バグ数を表す. また, $b > 0, c > 0$ である. 図 2 に, ロジスティック曲線の例を示す. ロジスティック曲線の変曲点は $t = (\log m)/b$ であり, $t = 0$ のときはゴンベルツ曲線と同様に $M_L(0)$ が 0 とはならないことに注意する.

ロジスティック曲線によるパラメータ推定も, 同様に回帰分析で行う. つまり, 各パラメータ $\theta = (\omega, b, c)$ は $M_L(t; \theta)$ と実データの残差平方和

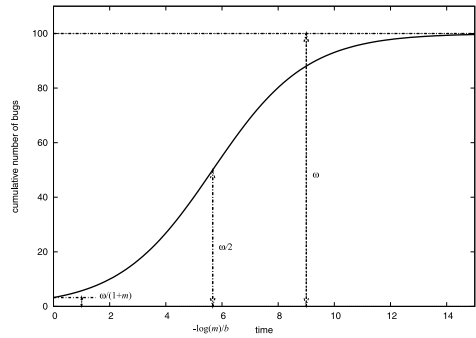


図 2 ロジスティック曲線 ($\omega = 100, m = 30, b = 0.6$)
Fig. 2 Logistic curve ($\omega = 100, m = 30, b = 0.6$).

$$RSS(\theta) = \sum_{i=1}^K \left\{ \sum_{j=1}^i y_j - M_L(t_i; \theta) \right\}^2 \quad (4)$$

を最小にする値として与えられる. もちろん, ゴンベルツ曲線と同じく三点推定法を用いることも可能であることを付記しておく.

3. NHPP モデル

ゴンベルツ曲線モデルとロジスティック曲線モデルは, 統計的には式 (1) や式 (3) で表される確定的な曲線とデータとの誤差を最小にする回帰モデルである. つまり, 確定的な曲線とその誤差を考慮した統計モデルと見なすことができる. このため, 実際の累積バグ数が時間的前後の値に関係なく増減するモデルと見なせる. しかしながら, 実際のテスト工程における累積バグ数は単調に増加する計数過程であるため, 単純に傾向曲線モデルをあてはめた場合には, 単調増加に関する情報を考慮していない分, 推定精度が低くなることが考えられる. そこで, 平均的に従来の傾向曲線と同じ振舞いをする計数過程を用いて, テスト工程における累積バグ数を表現することは有効であり²⁾, 具体的に, 次に示す非同次ポアソン過程 (NHPP) を用いることが一般的である.

テスト工程において, 時刻 $t (\geq 0)$ での累積バグ数を確率過程 $\{N(t), t \geq 0\}$ で表す. この $N(t)$ が次の条件

- 条件 1. $N(t) = 0,$
- 条件 2. $\{N(t), t \geq 0\}$ は独立増分を持つ,
- 条件 3. $\Pr\{N(t + \Delta t) - N(t) = 1\} = m(t)\Delta t + o(\Delta t),$

条件 4. $\Pr\{N(t + \Delta t) - N(t) \geq 2\} = o(\Delta t),$
を満たすならば, $\{N(t), t \geq 0\}$ は NHPP であるという. ここで $o(\Delta t)$ は, 微小時間 Δt の 2 次以上の高次項であり, 十分大きい t に対して時間の影響は無視

できることを表している．また $m(t)$ は強度関数と呼ばれ，バグの検出率を表す．この $m(t)$ を用いて，時刻 t における累積バグ数の期待値は

$$M(t) = \int_0^t m(x) dx \quad (5)$$

によって与えられる．また，時刻 t における累積バグ数の確率関数は

$$\Pr\{N(t) = n\} = \frac{M(t)^n}{n!} e^{-M(t)} \quad (6)$$

となる．式 (5) で与えられる $M(t)$ は NHPP の平均値関数と呼ばれ，この関数を特定することで様々な累積バグ数の振舞いを表現することができる．

次に，NHPP モデルにおけるパラメータ推定について言及する．NHPP モデルは傾向曲線モデルと異なり確率モデルであるため，最尤法と呼ばれる推定手続きを直接利用することができる．最尤法による推定量（最尤推定量）は漸近正規性や漸近有効性などの統計的に多くの良い性質を持つため，他の推定量に比べて利用範囲の広い推定量である．前節と同様に，表 1 のような実データが与えられているものとする．このとき，平均値関数 $M(t; \theta)$ (θ はパラメータ) に対する対数尤度関数

$$\begin{aligned} \text{LLF}(\theta) = & \sum_{i=1}^K y_i \log\{M(t_i; \theta) - M(t_{i-1}; \theta)\} \\ & - M(t_K; \theta) - \sum_{i=1}^K \log y_i! \quad (7) \end{aligned}$$

を定義する．最尤法における推定値は $\text{LLF}(\theta)$ を最大にする値によって与えられる．しかしながら，一般的に NHPP に対する対数尤度関数は残差平方和よりも非線形性の強い関数であり，多くの場合，多峰関数となる．このため，汎用的な数値解析手法がうまく機能しないことが経験的に知られている．つまり，一般的に最小二乗法よりも最尤法によるパラメータ推定は，より多くの計算労力が必要となる．

3.1 ゴンベルツ型 NHPP モデル

山田³⁾ は以下の平均値関数を持つ NHPP モデルを提案している．

$$M_G(t) = \omega(a^{b^t} - a). \quad (8)$$

これは，本来のゴンベルツ曲線が原点において $M_G(0) = 0$ となるように式 (1) の $M_G(t)$ を調整したものである．上記の平均値関数を持つ NHPP を考えることにより，フォールト検出過程をゴンベルツ曲線に準じた計数過程で表現することが可能である．そこでは，最尤法によるパラメータ推定値の算出に關

する労力を軽減するために，式 (1) によって与えられるゴンベルツ曲線の三点推定法や最小二乗法から得られる値を近似的に用いることが提案されている．

3.2 ロジスティック型 NHPP モデル

累積バグ数の平均がロジスティック曲線を描く NHPP モデルは習熟 S 字形信頼度成長モデル⁴⁾ と呼ばれる．これは，テストにより検出されるバグには，他のバグとは独立に検出されるものと，他のバグの検出が引き金となって検出されるものが混在するという仮定から導かれる．具体的に，平均値関数は

$$M_L(t) = \frac{\omega(1 - e^{-bt})}{1 + ce^{-bt}} \quad (9)$$

となり，本来のロジスティック曲線が， $t = 0$ で $M_L(0) = 0$ となるように修正されている．文献 4) では，推定手続きを簡単にするため，パラメータ c を既成的に与え，パラメータ ω と b だけを最尤法によって推定する手続きを紹介しているが，一般的な場合についてはいまだに解かれていない．

4. 有限デバッグモデル

文献 3), 4) におけるゴンベルツ型 NHPP モデルやロジスティック型 NHPP モデルにおける問題は，本来は実データから推定されるべきパラメータに対して最小二乗法による推定値を与えたり，既知定数を作為的に与えたりしている点にある．そこで本論文では，ゴンベルツ型 NHPP モデルやロジスティック型 NHPP モデルを有限デバッグモデル⁶⁾ の枠組みで再記述することで，統計的な観点からこれらのモデルの差異を明確にし，EM アルゴリズムに基づいたより簡便かつ厳密な推定アルゴリズムを構築する．

以下の仮定を設定する．

仮定 A：プログラム中に含まれる初期のバグ数 N_0 は平均 $\omega (> 0)$ のポアソン分布に従う．

仮定 B：テスト工程で検出されるバグはそれぞれ独立かつ時間に関してランダムに検出され，各々のバグの検出時間 X_i は確率分布 $F(t; \theta)$ ，密度関数 $f(t; \theta)$ を持つ非負の確率変数によって記述される (θ はパラメータ集合)．

上述の仮定の下で，時刻 $t (\geq 0)$ までに検出される累積バグ数の確率関数は

$$\begin{aligned} \Pr\{N(t) = n\} &= \sum_{k=n}^{\infty} \Pr\{N(t) = n | N_0 = k\} \Pr\{N_0 = k\} \\ &= \sum_{k=n}^{\infty} \binom{k}{n} F(t; \theta)^n \bar{F}(t; \theta)^{k-n} \frac{\omega^k}{k!} e^{-\omega} \end{aligned}$$

$$= \frac{\{\omega F(t; \theta)\}^n}{n!} e^{-\omega F(t; \theta)} \quad (10)$$

によって表現される．ここで $\bar{F}(t; \theta) = 1 - F(t; \theta)$ である．式 (10) は平均値関数 $\omega F(t)$ を持つ NHPP の確率関数と等価である．つまり，NHPP モデルにおける平均値関数の選択とバグの検出時間分布 $F(t)$ の選択は等価であることが分かる．

有限デバッグモデルで累積バグ数を記述する利点の 1 つとして，次に示す EM アルゴリズムによるパラメータ推定が可能となる点があげられる⁷⁾．EM アルゴリズムは，観測不可能な変数を含む推定問題において，期待対数尤度を最大にする値をパラメータの更新値とする繰返演算を実行することで，観測可能な変数に対する最尤推定値を求める手法である．いま，有限デバッグモデルにおいて，プログラム中に含まれるすべてのバグが検出されたものと仮定する．このとき，パラメータ推定問題はバグ検出時間分布のパラメータ θ だけに対する推定問題として簡略化されることに注目する．

検出バグ数に関する実データ D が与えられたもとで，暫定的なパラメータ値 ω' , θ' を用いて，推定値は

$$\omega := E[N_0 | D; \omega', \theta'], \quad (11)$$

$$\theta := \operatorname{argmax}_{\theta} \left\{ E \left[\sum_{i=1}^{N_0} \log f(X_i | \theta) \middle| D; \omega', \theta' \right] \right\} \quad (12)$$

によって与えられる．ここで， $\operatorname{argmax}\{\cdot\}$ は括弧の中の値が最大となる引数を返す関数である．また， $E[\cdot | D; \omega', \theta']$ は暫定パラメータ ω' と θ' を用いて算出される「実データ D が観測されたもとの期待値」を表す．有限デバッグモデルでは，式 (11) および式 (12) で表現される更新式を繰り返し計算することで，最尤推定値を効率良く求めることができる．

次に，具体的なバグ検出時間の確率分布を与えることで，ゴンベルツ型 NHPP モデルとロジスティック型 NHPP モデルに確率論的な解釈を与え，さらにそれらのパラメータ推定アルゴリズムについて述べる．

4.1 ゴンベルツ型 NHPP モデル

バグ検出時間の確率分布として，最大値に対する 2 重指数型の極値分布 (Gumbel 分布) を考える．

$$F(t) = \exp\{-e^{-(t-\mu)/\theta}\}, \quad -\infty < t < \infty. \quad (13)$$

ここで， $\mu > 0$, $\theta > 0$ である．Gumbel 分布は負の値を許容するので，非負の確率変数であるバグ検出時間分布としては不適當である．そこで，Gumbel 分布

に対して次の切断分布を考える．

$$F_{trunc}(t) = 1 - \bar{F}(t)/\bar{F}(0) \\ = \frac{a^{bt} - a}{1 - a}, \quad 0 \leq t < \infty. \quad (14)$$

ここで， $a = \exp(-e^{\mu/\theta})$, $b = e^{-1/\theta}$ である．モデルの自由度を考えると，式 (14) で与えられるバグ検出時間分布を適用したモデルは式 (8) で与えられるゴンベルツ型 NHPP モデルと本質的に等価であることが分かる．また，文献 8) では，非負の確率変数の与え方として，対数化による方法を考えることで，一般化指数型 NHPP モデル⁹⁾ と等価なモデルが表現可能であることも示している．

次に，表 1 の形式のデータに対してパラメータ推定アルゴリズムを導出する．いま，切断された時刻 $t = 0$ 以前におけるバグ検出時間と各時間間隔 $[t_{i-1}, t_i)$ におけるバグ検出時間に関するデータを完全データと見なすことで，式 (11) と式 (12) から表 1 の不完全データに対する EM アルゴリズムを構築する (付録 A.1 を参照)．具体的には，ゴンベルツ型 NHPP モデルに対する EM アルゴリズムによる更新式は

$$\omega := S_1 + \omega\{1 - \bar{F}(0) + \bar{F}(t_K)\}, \quad (15)$$

$$\mu := -\theta \log \left\{ \frac{S_2 + \omega e^{-\mu/\theta} (1 - \psi_0 + \psi_K)}{S_1 + \omega\{1 - \bar{F}(0) + \bar{F}(t_K)\}} \right\}, \quad (16)$$

$$\theta := \theta \frac{S_3 + \omega\{1 - \xi_0 + \xi_K\}}{S_1 + \omega\{1 - \bar{F}(0) + \bar{F}(t_K)\}}. \quad (17)$$

ここで

$$S_1 = \sum_{i=1}^K y_i, \quad (18)$$

$$S_2 = \sum_{i=1}^K \frac{y_i e^{-\mu/\theta} (\psi_{i-1} - \psi_i)}{\bar{F}(t_{i-1}) - \bar{F}(t_i)}, \quad (19)$$

$$S_3 = \sum_{i=1}^K \frac{y_i (\xi_{i-1} - \xi_i)}{\bar{F}(t_{i-1}) - \bar{F}(t_i)}, \quad (20)$$

$$\psi_i = \bar{F}(t_i) - \theta f(t_i), \quad (21)$$

$$\xi_i = \bar{F}(t_i) + (t_i - \mu) f(t_i) \quad (22)$$

である．ただし，ここで算出される ω の推定値は負の時刻で検出されるバグの個数も含んでいるため，最終的な推定値は $\omega := \omega \bar{F}(0)$ によって得られる．

4.2 ロジスティック型 NHPP モデル

バグ検出時間分布に以下のロジスティック分布を考える．

$$F(t) = \frac{e^{(t-\mu)/\phi}}{1 + e^{(t-\mu)/\phi}}, \quad -\infty < t < \infty. \quad (23)$$

極値分布を適用した場合と同様に，ロジスティック

分布も非負の値をとる確率変数であるために、次のような切断分布を考える。

$$F_{trunc}(t) = 1 - \frac{\overline{F}(t)}{\overline{F}(0)} = \frac{1 - e^{-bt}}{1 + ce^{-bt}}, \quad 0 \leq t < \infty. \quad (24)$$

ここで、 $b = 1/\phi$ 、 $c = e^{\mu/\phi}$ である。これをバグ検出時間分布とする有限デバッグモデルを考えることで、ロジスティック型 NHPP モデルが表現可能であり、式 (9) のロジスティック型 NHPP モデル (習熟 S 字形モデル) と等価である。また、式 (23) の分布の対数化を行ったバグ検出時間分布は対数ロジスティック分布と呼ばれ、対応する対数ロジスティック NHPP モデルを導出することも可能である¹⁰⁾。

ゴンペルツ型 NHPP モデルと同様に、切断時刻 $t = 0$ までのバグ検出時刻と各時間間隔におけるバグ検出時刻を完全データとすることで次の更新式を得る。

$$\omega := S_1 + \omega\{1 - \overline{F}(0) + \overline{F}(t_K)\}, \quad (25)$$

$$\mu := \mu + \phi \log \left\{ \frac{S_1 + \omega\{1 - \overline{F}(0) + \overline{F}(t_K)\}}{S_2 + \omega(1 - \psi_0 + \psi_K)} \right\}, \quad (26)$$

$$\phi := \phi \frac{S_3 + \omega\{1 - \xi_0 + \xi_K\}}{S_1 + \omega\{1 - \overline{F}(0) + \overline{F}(t_K)\}}. \quad (27)$$

ここで

$$S_1 = \sum_{i=1}^K y_i, \quad (28)$$

$$S_2 = \sum_{i=1}^K \frac{y_i(\psi_{i-1} - \psi_i)}{\overline{F}(t_{i-1}) - \overline{F}(t_i)}, \quad (29)$$

$$S_3 = \sum_{i=1}^K \frac{y_i(\xi_{i-1} - \xi_i)}{\overline{F}(t_{i-1}) - \overline{F}(t_i)}, \quad (30)$$

$$\psi_i = \frac{1}{(1 + e^{(t_i - \mu)/\phi})^2}, \quad (31)$$

$$\xi_i = \frac{1 + (1 + (t_i - \mu)/\phi)e^{(t_i - \mu)/\phi}}{(1 + e^{(t_i - \mu)/\phi})^2} \quad (32)$$

である。また、ゴンペルツ型 NHPP モデルと同様にパラメータ ω は負の時刻で検出されるバグの個数を含んでいるため、上記の更新式で算出した後に $\omega := \omega \overline{F}(0)$ によってパラメータ ω の推定値を修正する必要がある。

5. 数 値 例

ここでは、ゴンペルツ曲線およびロジスティック曲線による回帰モデルと、ゴンペルツ型 NHPP モデルおよびロジスティック型 NHPP モデルの予測精度の比較を行う。つまり、実際のソフトウェアテスト工程において、ある時刻までに検出されたバグ数のデータ

を用いて将来のバグ検出数の振舞いに関する予測を行う。このとき、回帰モデルと NHPP に基づいたモデルでは予測方法が異なることに注意しなければならない。回帰モデルはデータの時系列を考慮したモデルではなく、各時刻における検出バグ数の平均的な傾向とばらつきをモデル化したものである。そのため、将来の $(T, T + s]$ 時間における検出バグ数の振舞い $\hat{M}(t)$ は、最小二乗法で推定されたパラメータ $\hat{\theta}$ を用いて、

$$\hat{M}(t) = M_r(t; \hat{\theta}), \quad T < t \leq T + s \quad (33)$$

となる。ここで、 \mathcal{D}_T と $M_r(t)$ は観測時刻 T までの検出されたバグ数のデータと傾向曲線である。一方、NHPP に基づいたモデルは検出バグ数の時系列データを考慮している。つまり、将来の $(T, T + s]$ 時間における検出バグ数の平均的な振舞いは、最尤法で推定されたパラメータ $\hat{\theta}$ に加えて、観測時間までのバグ検出数の情報を用いて、

$$\begin{aligned} \hat{M}(t) &= E[N(t)|\mathcal{D}_T; \hat{\theta}] \\ &= N(T) + (M_N(t; \hat{\theta}) - M_N(T; \hat{\theta})), \\ &\quad T < t \leq T + s \end{aligned} \quad (34)$$

となる。ここで、 $N(T)$ および $M_N(t)$ は時刻 T におけるバグの検出数と NHPP モデルにおける平均値関数である。つまり、現在までに検出されたバグ数と将来の期待増分の和として表現される。ここでは、これらの平均的な予測値を用いてモデルの性能評価を行う。

使用されるデータは実際のソフトウェア開発のテスト工程において検出された 3 種類のデータ (DS1, DS2, DS3) である¹¹⁾。これらはそれぞれ、181 週、41 週、114 週にわたるテストで検出されたバグ数を記録したものである。これらのデータに対して、DS1 では 150 週、DS2 では 30 週、DS3 では 90 週までのデータを用いて将来の予測精度を検証する。

図 3、図 4、図 5、図 6、図 7、図 8 は、各データに対してゴンペルツ曲線による回帰モデルとゴンペルツ型 NHPP モデル、ロジスティック曲線による回帰モデルとロジスティック型 NHPP モデルを適用した結果を比較している。特に回帰モデルでは傾向曲線の推定値を、NHPP に基づいたモデルでは推定された

三点推定法は計算機があまり発達していない時代の簡易的な推定手法であると考えられる。実際に計算してみると予測に関して明らかに精度が悪かったため、ここでは最小二乗法で推定されたパラメータを用いた場合の予測精度のみを示す。たとえば DS1 の 100 週まではテストが十分にこなされていないため、バグ数が依然として急激に増えている。バグ検出データしか利用しないソフトウェア信頼性モデルでは、100 週までのようなデータを利用してうまく機能しないことが知られている。

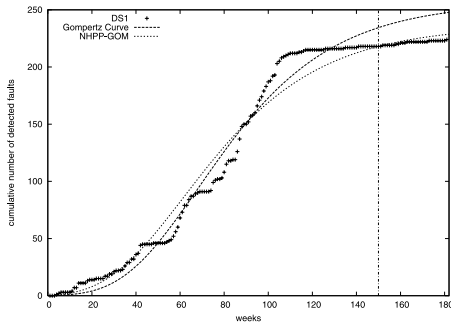


図 3 ゴンペルツ曲線とゴンペルツ型 NHPP モデルの平均値関数 (DS1)

Fig. 3 Gompertz curve and mean value function of Gompertz NHPP model (DS1).

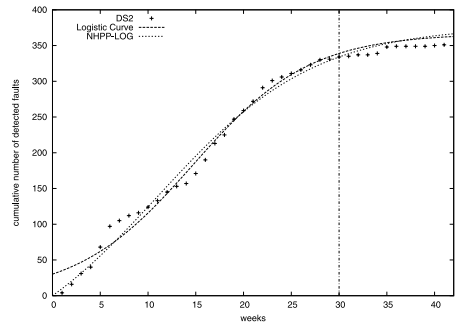


図 6 ロジスティック曲線とロジスティック型 NHPP モデルの平均値関数 (DS2)

Fig. 6 Logistic curve and mean value function of Logistic NHPP model (DS2).

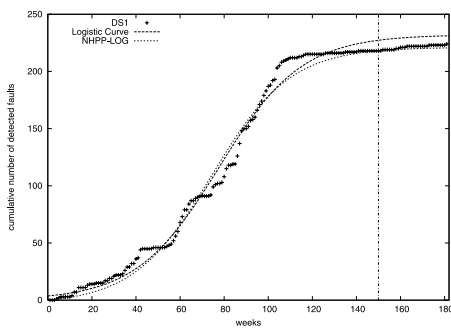


図 4 ロジスティック曲線とロジスティック型 NHPP モデルの平均値関数 (DS1)

Fig. 4 Logistic curve and mean value function of Logistic NHPP model (DS1).

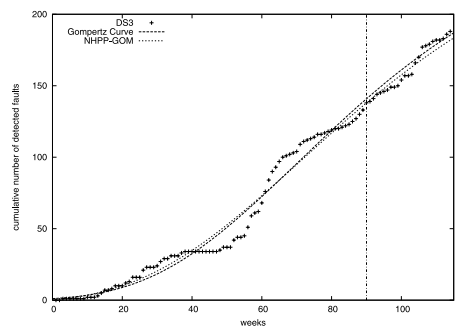


図 7 ゴンペルツ曲線とゴンペルツ型 NHPP モデルの平均値関数 (DS3)

Fig. 7 Gompertz curve and mean value function of Gompertz NHPP model (DS3).

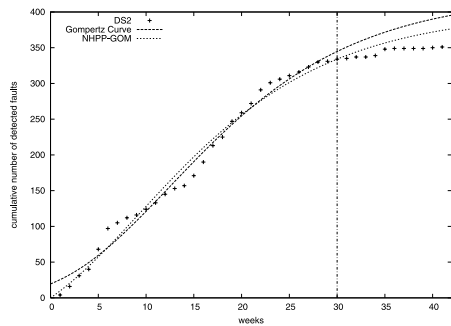


図 5 ゴンペルツ曲線とゴンペルツ型 NHPP モデルの平均値関数 (DS2)

Fig. 5 Gompertz curve and mean value function of Gompertz NHPP model (DS2).

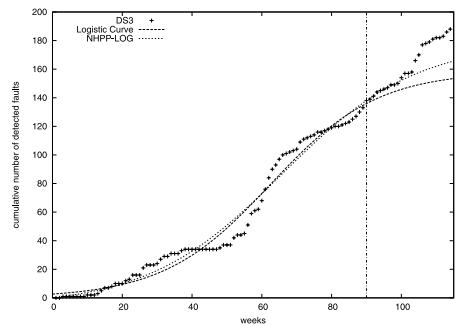


図 8 ロジスティック曲線とロジスティック型 NHPP モデルの平均値関数 (DS3)

Fig. 8 Logistic curve and mean value function of Logistic NHPP model (DS3).

平均値関数をそれぞれ描いている。また図中の縦線は観測時間、すなわち、縦線以前のデータを用いてパラメータの推定を行ったことを示している。これらの図において、回帰モデルより NHPP に基づいたモデルの方が、平均的な傾向において実際のデータによく適合していることが見てとれる。

次に、表 2 は観測時間までのデータに基づいて最小

残差平方和 (LRSS) と最大対数尤度 (MLL) を求めた結果を示している。ゴンペルツ曲線 (GOM) とロジスティック曲線 (LOG)、あるいはゴンペルツ型 NHPP モデル (NHPP-GOM) とロジスティック型 NHPP モデル (NHPP-LOG) はともにモデルの自由度 (パラメータ数) が同じであるため、これらを単純に比較することによってモデルの適合性をみる事ができる。

表 2 モデルの比較
Table 2 Model comparison.

Data	Model	LRSS	MLL	PSS
DS1	GOM	17298.37	—	13452.95
	NHPP-GOM	—	-232.62	296.57
	LOG	9024.66	—	2188.33
	NHPP-LOG	—	-214.35	119.33
DS2	GOM	4452.63	—	10295.22
	NHPP-GOM	—	-114.48	2545.60
	LOG	5001.07	—	1085.15
	NHPP-LOG	—	-113.28	1059.28
DS3	GOM	4815.04	—	635.36
	NHPP-GOM	—	-131.24	553.47
	LOG	3920.94	—	9565.56
	NHPP-LOG	—	-130.40	3849.31

また、同時に予測精度の評価として予測平方和 (Predictive Squared Sum: PSS) を算出した。これは各モデルにおける平均傾向の予測値 $\hat{M}(t)$ を用いて

$$PSS = \sum_{k=n}^K \left\{ \sum_{j=1}^k y_j - \hat{M}(t_k) \right\}^2 \quad (35)$$

と定義される。式 (35) において、 n は観測時刻 $T < t_n$ となる最小の値であり、PSS は観測時間後の二乗誤差を表している。この結果において、予測精度の観点から NHPP に基づいたモデルに対して最尤法を用いて推定を行ったものが最も良い結果となることが分かる。すなわち、バグの検出事象のように時系列で観測されるデータに対して単純な回帰モデルを適用することは、予測誤差の増大を招くことが分かる。

6. ま と め

本論文では、ゴンベルツ曲線やロジスティック曲線と平均的に同等な振舞いを示す NHPP モデルについて再考した。特に、有限デバッグモデルの枠組みにおいてゴンベルツ型 NHPP モデルとロジスティック型 NHPP モデルに関する確率的な違いを明らかにし、EM アルゴリズムを適用することで、従来まで困難であった最尤推定値を算出するための問題点を克服することができた。また、ゴンベルツ曲線あるいはロジスティック曲線を用いた回帰モデルと NHPP モデルを比較することで、最尤推定値を用いて厳密に信頼性の予測を行うことの優位性を示した。

謝辞 本研究の一部は文部省科学研究費萌芽研究 15651076 (2003–2005) および基盤研究 (B) Grant No.1631011600 (2004–2006) による助成の下で行われたものである。

参 考 文 献

- 1) 坂田：ソフトウェアの生産管理における予測技法の定式化—静的な予測および故障率推定モデル、電子通信学会論文誌, Vol.57-D, No.5, pp.277–283 (1974).
- 2) 山田：ソフトウェア信頼性モデル—基礎と応用、日科技連出版社 (1994).
- 3) 山田：ゴンベルツ曲線を用いた確率的ソフトウェア信頼度成長モデル、情報処理学会論文誌, Vol.33, No.7, pp.964–969 (1992).
- 4) Ohba, M.: Inflection S-shaped software reliability growth model, *Stochastic Models in Reliability Theory*, Osaki, S. and Hatoyama, Y. (Eds.), pp.144–165, Springer-Verlag, Berlin, Germany (1984).
- 5) 菱谷：回帰分析のはなし、東京図書 (1985).
- 6) Langberg, N. and Singpurwalla, N.D.: Unification of some software reliability models, *SIAM J. Sci. Comput.*, Vol.6, pp.781–790 (1985).
- 7) Okamura, H., Watanabe, T. and Dohi, T.: An iterative scheme for maximum likelihood estimation in software reliability modeling, *Proc. 14th Int'l Sympo. Software Reliab. Eng.*, pp.246–256 (2003).
- 8) Okamura, H. and Dohi, T.: EM algorithms for extreme-value software reliability models, *CD-ROM Proc. 4th Int'l Conf. on Mathematical Methods in Reliab.* (2004).
- 9) Goel, A.L.: Software reliability models: Assumptions, limitations and applicability, *IEEE Trans. Softw. Eng.*, SE-11, pp.1411–1423 (1985).
- 10) Okamura, H., Dohi, T. and Osaki, S.: EM algorithms for logistic software reliability models, *Proc. IASTED Int'l Conf. on Software Eng.*, pp.263–268 (2004).
- 11) Lyu, M.E. (Ed.): *Handbook of Software Reliability Engineering*, McGraw-Hill, New York (1996).

付 録

A.1 式 (15)–(17) の導出

ここでは式 (15)–(17) の導出を行う。いま、ゴンベルツ型 NHPP モデルに対する完全なバグ検出時刻データ $X_{-M+1} < \dots < X_{-1} < X_0 < X_1 < \dots < X_{N-M}$ が与えられているものとする。ここで、 N はソフトウェア内に潜在しているバグの総数、 M は時刻 $t = 0$ 以前 (負の時刻) に検出されるバグの総数とする。ゴンベルツ型 NHPP モデルは Gumbel 分布の切断分布として構築されている点と、有限デバッグモデルの性質から、 $\{X_i; i = -(M-1), \dots, N-M\}$ は単純に

Gumbel 分布からの標本であると考えられることができる。したがって、もとの Gumbel 分布のパラメータ μ と θ は次の尤度方程式の解として導出される。

$$\hat{\mu} = -\hat{\theta} \log \left\{ \frac{1}{N} \sum_{i=-M+1}^{N-M} \exp(-X_i/\hat{\theta}) \right\}, \quad (36)$$

$$N = \sum_{i=-M+1}^{N-M} \left(\frac{X_i - \hat{\mu}}{\hat{\theta}} \right) \times \left\{ 1 - \exp \left(-\frac{X_i - \hat{\mu}}{\hat{\theta}} \right) \right\}. \quad (37)$$

また、パラメータ ω は

$$\hat{\omega} = N \quad (38)$$

となる。しかしながら、実際は $X_1 < \dots < X_K$, $K < N - M$ のみが観測されるので、上記の尤度方程式を観測データに対する期待値を用いて書きかえることで推定アルゴリズムが得られる。たとえば、パラメータ ω に関しては

$$\begin{aligned} \hat{\omega} &= E[N|D] \\ &= E[N|t_{i-1} < X_{s_{i-1}+z_i} < t_i; \\ &\quad z_i = 1, \dots, y_i, \quad i = 1, \dots, K]. \end{aligned} \quad (39)$$

ここで $s_i = \sum_{j=0}^i y_j$, $t_0 = 0$, $y_0 = 0$ である。さらに上記の期待値は、任意の可測関数 $h(\cdot)$ に対して

$$\begin{aligned} E \left[\sum_{i=-M+1}^{N-M} h(X_i) \middle| t_{i-1} < X_{s_{i-1}+z_i} < t_i, \right. \\ \left. z_i = 1, \dots, y_i, \quad i = 1, \dots, K \right] \\ = \sum_{i=1}^K \frac{y_i \int_{t_{i-1}}^{t_i} h(u) f(u) du}{\int_{t_{i-1}}^{t_i} f(u) du} \\ + \omega \int_{-\infty}^0 h(u) f(u) du \\ + \omega \int_{t_K}^{\infty} h(u) f(u) du \end{aligned} \quad (40)$$

なる関係を用いて算出ができる（詳細は文献 7）を参照）。ここで $f(u)$ は Gumbel 分布の密度関数である。これより式 (15) を得る。式 (15) では、第 1 項が観測されたバグ数、第 2 項が未観測のバグ数を表している。

他の式についても期待値を算出することで更新式が得られる。また、ロジスティック型 NHPP モデルについても同様の手続きである。

(平成 17 年 5 月 25 日受付)

(平成 17 年 12 月 2 日採録)



岡村 寛之 (正会員)

昭和 48 年生。平成 7 年広島大学工学部第二類卒業。平成 9 年同大学大学院工学研究科システム工学専攻博士課程前期修了。同年 CSK (株) 入社。平成 10 年広島大学工学部助手。

平成 15 年広島大学大学院工学研究科助教授。博士 (工学)。主として、待ち行列システム、信頼性・安全性システムの研究に従事。日本 OR 学会、電子情報通信学会、日本応用数理学会、IEEE 各会員。



古村 仁志

昭和 56 年生。平成 16 年広島大学工学部第二類卒業。平成 16 年同大学大学院工学研究科情報工学専攻博士課程前期入学。現在、ソフトウェア信頼性に関する研究に従事。



土肥 正

昭和 40 年生。平成元年広島大学工学部第二類卒業。平成 4 年同大学大学院工学研究科システム工学専攻博士課程後期中途退学。同年広島大学工学部助手。平成 8 年同助教授。

平成 14 年広島大学大学院工学研究科教授。博士 (工学)。平成 4 年プリティッシュ・コロンビア大学 (カナダ) 客員研究員。平成 12 年デューク大学 (アメリカ) 客員研究員。主として、信頼性理論および数理システムの研究に従事。日本 OR 学会、日本応用数理学会、電子情報通信学会、システム制御情報学会、計測自動制御学会、IEEE 各会員。