

制約付きノンパラメトリック最尤推定に基づくソフトウェア信頼性評価について

広島大学大学院工学研究科 齋藤 靖洋 (Yasuhiro Saito) 土肥 正 (Tadashi Dohi)
Graduate School of Engineering, Hiroshima University

1 はじめに

近年, 多くのコンピュータシステムはソフトウェアによってコントロールされており, ソフトウェア信頼性の正確な把握は重要な課題となっている. 特に, ソフトウェア信頼性の定量化は製品の品質保証の観点からも極めて重要である. 定量的なソフトウェア信頼度は, ソフトウェアフォールトによって引き起こされるソフトウェア故障が任意の期間起こらない確率として定義され, ソフトウェア信頼性モデルはソフトウェア工学及び信頼性工学において広く研究されている. 過去 40 年間で, 多くのソフトウェア信頼性モデルが提案され, ソフトウェア信頼度の推定や定量的なソフトウェアテストの管理のために用いられている. 中でも, 非同次ポアソン過程に基づいたソフトウェア信頼性モデルは, 取り扱いの容易さやデータ適合度の高さから, 検出フォールト数の確率的な振る舞いを記述するために広く適用されている. Goel and Okumoto [4], Gokhale and Trivedi [5], Okamura ら [9] は代表的な NHPP に基づくソフトウェア信頼性モデルを提案した. 上記の NHPP モデルは, それを特徴付ける平均値関数や強度関数が既知であるというパラメトリックモデルに分類される. パラメトリックモデルはフォールト検出時刻の統計的性質に依存することから, パラメトリック NHPP モデルの選択はフォールト検出時間分布の選択と同じ意味を持つ. しかしながら, 過去 40 年間で報告された多くの研究成果から, あらゆるソフトウェアフォールトデータに最も適合するような唯一のパラメトリック NHPP モデルは存在しないことが知られている. そのため, 実際のソフトウェア開発プロジェクトにおいては, データへの適合度を調べることで数多くあるモデル候補から最良のパラメトリック NHPP モデルを選択する必要がある.

この問題点を克服すべく, 特定のフォールト検出時間分布を持たないノンパラメトリック NHPP モデルが考察されている. Miller and Sofer [8] は, NHPP の平均値関数や強度関数が未知であるという条件下で, 平均値関数についての線形区分推定量を提案し, その傾きを強度関数の推定量として定義した. Gandy and Jensen [3], Wang ら [11], Kaneishi and Dohi [6] はそれぞれ独立にカーネル関数に基づくソフトウェア強度関数の推定手法を提案した. しかしながら, カーネル関数を用いた推定手法ではバンド幅を慎重に決定する必要があり, 計算コストが極めて高くなる欠点を持つ. 本稿では, NHPP モデルを用いたソフトウェア信頼性評価を目的とし, 2 種類のノンパラメトリック最尤推定量 (NPMLE) を提案する. 一つ目として, Boswell [2] によって提案された典型的なノンパラメトリック最尤推定量をソフトウェアフォールト検出時刻データの解析に用いる. 二つ目として, 一般的な順序統計量の観点から, フォールト検出時刻データに Marshall/Proschan 推定量 [7] と呼ばれる故障率関数の推定量を適用する. これら 2 種類のノンパラメトリック最尤推定量を 6 個のソフトウェア開発プロジェクトデータに適用し, 代表的なパラメトリック NHPP モデルとの比較を行なう.

2 非同次ポアソン過程モデル

時刻 t までに観測された累積ソフトウェアフォールト数を $N(t)$ とおく. テスト開始前にソフトウェアに内在するフォールト数を N_0 (≥ 0) とし, 各ソフトウェアフォールトはパラメータ θ を持つ確率分布関数

$F(t; \theta)$ に従い、独立で同一に分布しているとする。この時、時刻 t までに n 個のソフトウェアフォールトが検出される確率は二項分布に従い、

$$\Pr\{N(t) = n\} = \binom{N_0}{n} F(t; \theta)^n (1 - F(t; \theta))^{N_0 - n} \quad (1)$$

と表される。更に、もしソフトウェアの初期残存フォールト数 N_0 が平均 $\omega (> 0)$ のポアソン分布に従うとした場合、時刻 t までに検出される総フォールト数が n である確率は

$$\Pr\{N(t) = n\} = \frac{\{\omega F(t; \theta)\}^n}{n!} \exp\{-\omega F(t; \theta)\} \quad (2)$$

によって与えられる。これは平均値関数 $\Lambda(t; \xi) = \omega F(t; \theta)$ を持つ非同次ポアソン過程 (NHPP) の確率関数と一致する。ここで、 $\xi = (\omega, \theta)$ である。このことから、NHPP に基づくソフトウェア信頼性モデル (NHPP モデル) はフォールト検出時間分布 $F(t; \theta)$ によって特徴付けることが出来る。更に、確率密度関数 $f(t; \theta) = dF(t; \theta)/dt$ が存在する場合、確率分布関数 $F(t; \theta)$ の故障率は

$$r(t; \theta) = \frac{f(t; \theta)}{1 - F(t; \theta)} \quad (3)$$

と定義される。式 (3) より、

$$f(t; \theta) = r(t; \theta) \exp \left\{ - \int_0^t r(x; \theta) dx \right\} \quad (4)$$

が成り立つことは明らかである。したがって、NHPP の強度関数 $d\Lambda(t; \xi)/dt = \lambda(t; \xi) = \omega f(t; \theta)$ は

$$\lambda(t; \xi) = \omega r(t; \theta) \exp \left\{ - \int_0^t r(x; \theta) dx \right\} \quad (5)$$

によって表される。

パラメータ $\xi = (\omega, \theta)$ を推定するために最も広く用いられている手法として、最尤法が挙げられる。テスト期間 $t \in (0, T]$ において、 $n (> 0)$ 個のフォールト検出時刻データ t_i ($i = 1, 2, \dots, n$) が観測されたとすると、NHPP モデルの対数尤度関数は

$$LLF(\xi | t_1, \dots, t_n, T) = \sum_{i=1}^n \log \lambda(t_i; \xi) - \Lambda(T; \xi) \quad (6)$$

で与えられる。この時、最尤推定量 $\hat{\xi}$ は式 (6) を最大化するパラメータとして定義される。

3 制約付きノンパラメトリック最尤推定量 (CNPMLE)

3.1 強度関数推定量

1 節で述べたように、あらゆるフォールト検出時刻データに適合するようなパラメトリック NHPP モデルは存在しないことが知られている。したがって、データに最も適合する最適なパラメトリック NHPP モデルを得るためには、フォールト検出時間分布 $F(t; \theta)$ を慎重に選択する必要がある。Okamura and Dohi [10] は指数分布、ガンマ分布、パレート分布、切断正規分布、対数正規分布、切断ロジスティック分布、対数ロジスティック分布、切断 Gumbel 分布、Frechet 型極値分布、Gompertz 分布 (最小値)、Weibull 分布といった 11 種類の基本的なフォールト検出時間分布を考慮すれば、実用上、十分であることを示唆した。しかしながら、上記 11 種類のパラメトリック NHPP モデルのいずれかが、これまでに提案されてきた他のモデルよりも常に高い有効性を発揮するとは限らない。データに最も適合するような平均値関数や対応す

るフォールト検出時間分布が分からないような不確実な状況の下では、NHPP モデルのノンパラメトリック推定量を考察することは有益であると言える。

はじめに、Boswell [2] が提案した強度関数の制約付きノンパラメトリック最尤推定量 (CNPMLE) を導入する。ここでの議論では、 $\Lambda(t; \xi) = \omega F(t; \theta)$ とは限らず、平均値関数が有界である NHPP モデル (すなわち、 $\lim_{t \rightarrow \infty} \Lambda(t; \xi) = \omega < \infty$) に限定する必要はない。強度関数 $\lambda(t)$ が時刻 $t \in (0, T]$ に対して非負な非増加関数であるとする。未知の強度関数 $\lambda(t)$ に対し、Boswell [2] は

$$\arg \max_{\lambda(t)} \sum_{i=1}^n \log \lambda(t_i) - \int_0^T \lambda(x) dx \quad (7)$$

で表される変分問題を考え、式 (7) の解がフォールト検出時刻のいくつかでのみ増加するような右連続の階段関数でなければならないことを示した。その解は

$$\hat{\lambda}(t_j) = \min_{1 \leq h \leq j} \max_{j \leq k \leq n} \frac{k - h}{t_k - t_h} \quad (8)$$

によって表され、区分的に線形な非連続関数として与えられる。本稿ではこれを強度関数推定量と呼ぶ。強度関数推定量はカーネル関数に基づく手法などに代表される他のノンパラメトリックモデルと比較して計算コストが非常に低いという特徴を持つ。

3.2 故障率推定量

3.1 節で紹介した CNPMLE では、強度関数の単調性を仮定した。しかしながら強度関数推定量を直接用いる場合には、式 (2) で表されるパラメトリック NHPP モデルと異なり、初期残存フォールト数 ω を推定することが出来ない。このため、いくつかの重要なソフトウェア信頼性尺度を推定することが困難となる。本節では、平均値関数が $\omega F(t)$ で表される任意の NHPP モデルについて、もう一つのノンパラメトリック最尤推定量を考える。式 (1) より、フォールト検出時刻データ $t_1 < t_2 < \dots < t_n$ は確率分布関数 $F(t)$ から得られた順序統計量と考えることが出来る。 t_i ($i = 1, 2, \dots, n$) を独立で同一な分布から得られるサンプルとみなした場合、対数尤度関数は

$$LLF(r(t_i), i = 1, 2, \dots, n) = \sum_{i=1}^n \log r(t_i) - \sum_{i=1}^n \int_0^{t_i} r(x) dx \quad (9)$$

となる。これを最大化する故障率を求めるために、Marshall and Proschan [7] は Boswell [2] と同様に $r(\cdot)$ に関する変分問題

$$\arg \max_{r(t_i), i=1, 2, \dots, n} LLF(r(t_i), i = 1, 2, \dots, n) \quad (10)$$

を定式化した。本稿では、ある制約条件の下で、式 (10) から得られるノンパラメトリック最尤推定量を故障率推定量と呼ぶ。具体的には、もしフォールト検出時間分布が IFR (Increasing Failure Rate) の場合、故障率の CNPMLE は

$$r(t) = \begin{cases} 0, & 0 \leq t < t_1, \\ r_k^{IFR}(t), & t_k \leq t < t_{k+1} \quad (k = 1, \dots, n-1), \\ r_{n-1}^{IFR}(t), & t = t_n, \\ \infty, & t_n < t \end{cases} \quad (11)$$

によって与えられる。ここで、

$$r_k^{IFR}(t) = \min_{v \geq k+1} \max_{u \leq k} \left[\frac{v - u}{J(u, v)} \right], \quad (12)$$

$$J(u, v) = \sum_{i=u+1}^v \{(n-i+1)(t_i - t_{i-1})\} \quad (13)$$

である。一方、フォールト検出時間分布が DFR (Decreasing Failure Rate) である場合には、故障率推定量は

$$r(t) = \begin{cases} r_0^{DFR}(t), & 0 \leq t \leq t_1, \\ r_k^{DFR}(t), & t_k < t \leq t_{k+1} \quad (k = 1, \dots, n-1), \\ 0, & t_n < t \end{cases} \quad (14)$$

と表され、

$$r_k^{DFR}(t) = \max_{v \geq k+1} \min_{u \leq k} \left[\frac{v-u}{J(u, v)} \right] \quad (15)$$

である。上記の結果から、フォールト検出時間分布の年齢特性 (IFR もしくは DFR) に合わせて故障率関数 $r(\cdot)$ が推定可能であることが分かる。

また、実際のソフトウェア開発プロジェクトデータでは、故障率が DFR から IFR へ変化する傾向を持つものが多く観測される。DFR から IFR への変化点を t_m とすると、この場合の CNPMLE は

$$r(t) = \begin{cases} r_0^{DFR}(t), & 0 \leq t \leq t_1, \\ r_k^{DFR}(t), & t_k < t \leq t_{k+1} \quad (k = 1, \dots, m-1), \\ r_k^{IFR}(t), & t_k < t \leq t_{k+1} \quad (k = m, \dots, n-1), \\ \infty, & t_n < t \end{cases} \quad (16)$$

と表すことが出来る。ここで、変化点 t_m はデータ点 $t_1, t_2, \dots, t_n$ から次節で紹介する総試験時間 (TTT) プロットを用いて見つけることが出来る。得られた故障率推定量を式 (5) に代入することで強度関数の推定量を得ることができ、それを用いて、式 (6) を最大化する ω は

$$\hat{\omega} = \frac{n}{1 - \exp\left(-\int_0^{t_n} r(t) dt\right)} = \frac{n}{F(t_n)} \quad (17)$$

となる。式 (11), (14), (16) で紹介した推定量は故障率の CNPMLE であり、式 (17) の推定量は ω について尤度関数を最大化したものである。フォールト検出時刻データ数 n が増えるにつれ、 $t_n \rightarrow \infty$ かつ $F(t_n) \rightarrow 1$ となり、 $\hat{\omega}$ は n に近づいていくことが分かる。

3.3 総試験時間 (TTT) プロット

3.2 節では、平均値関数が上限を持つ場合 ($\Lambda(t) \leq \omega$) の推定量を導出した。しかしながら、その推定手法は故障時間分布の年齢特性に基づいているため、フォールト検出時刻データの順序統計量から、確率分布関数 $F(t)$ の年齢特性を調べる必要がある。最も単純な手法として、標準総試験時間 (TTT) プロットを用いる方法が挙げられる。標準 TTT プロットは故障時間データの解析をするための手法として Barlow and Campo [1] によって導入された。

n 個のフォールト検出時刻データ $(t_1, t_2, \dots, t_n)$ に対して、総試験時間 (TTT) 統計量は

$$TTT_i = \sum_{j=1}^i (n-j+1)(t_j - t_{j-1}), \quad (i = 1, \dots, n) \quad (18)$$

によって定義される。式 (18) を正規化することによって、標準 TTT 統計量を

$$ST_i = \frac{TTT_i}{TTT_n}, \quad (i = 1, \dots, n; ST_0 = 0) \quad (19)$$

として得ることが出来る．更に，フォールト検出時刻データ $(t_1, t_2, \dots, t_n)$ に対する経験分布関数 $F_n(t)$ は

$$F_n(t) = \begin{cases} \frac{i}{n}, & t_i \leq t < t_{i+1} \ (i = 0, \dots, n; t_0 = 0), \\ 1, & t_n \leq t \end{cases} \quad (20)$$

と定義できる． $[0, 1] \times [0, 1]$ 平面上に $(i/n, ST_i)$ をプロットした上で，各点を線で結ぶことによって標準 TTT プロットを得ることが出来る．もし標準 TTT プロットの結果が 2 点 $(0,0)$ 及び $(1,1)$ を結ぶ 45 度線より常に上にある場合，フォールト検出時間分布は IFR であると判断することが出来る．一方，もし標準 TTT プロットが 45 度線を下回る場合，フォールト検出時間分布は DFR であると判定出来る．DFR から IFR に故障率が変化する場合には，標準 TTT プロットは 45 度線と交差する S 字型曲線となる．本稿では，それぞれの場合に対応する故障率推定量を，IFR 推定量，DFR 推定量， S 字型推定量と呼ぶ．この方法により，故障時間分布の年齢特性を調べることができ，更に式 (5) を用いて強度関数の推定値を得ることが出来る．

4 実データ解析

4.1 データ適合度

ここでは，パラメトリック NHPP モデルと本稿で提案した 2 種類の CNPMLE を持つノンパラメトリック NHPP モデルとのデータ適合度を比較する．比較対象として，SRATS (Software Reliability Assessment Tool on Spreadsheet) [10] 上に実装されている 11 種類のパラメトリック NHPP モデル (指数分布 (exp), ガンマ分布 (gamma), パレート分布 (pareto), 切断正規分布 (tnorm), 対数正規分布 (lnorm), 切断ロジスティック分布 (tlogis), 対数ロジスティック分布 (llogis), 切断 Gumbel 分布 (txvmax), Frechet 型極値分布 (lxvmax), Gompertz 分布 (最小値) (txvmin), Weibull 分布 (lxvmin)) を用いた．また，実際のソフトウェアテスト工程から得られた DS1 から DS6 までの 6 種類のフォールト検出時刻データセットを用いて，計算を行った．表 1 は解析に用いたデータセットを表しており，テスト工程で見つかった総フォールト数 n 及び標準 TTT プロットの結果を示している．また，“ベスト”は 11 種類の NHPP モデルの中で赤池情報量基準が最小となる最もデータ適合度が高いと言えるパラメトリックモデルを表す．図 1-2 はそれぞれ，DS1 及び DS5 に対して標準 TTT プロットを用いた結果を図示したものである．図 1 では，プロット結果が 45 度線と交差していることから，DS1 が S 字型傾向を持つことが分かる．一方，図 2 から DS5 が完全な IFR の傾向を示していることが分かる．

表 1: データセット

	n	年齢特性	ベスト
DS1	136	S 字型	lxvmin
DS2	129	S 字型	lxvmax
DS3	397	DFR	lnorm
DS4	278	S 字型	gamma
DS5	197	IFR	txvmin
DS6	207	S 字型	exp

図 3-4 は DS1 及び DS5 に対してベストパラメトリックモデルと 4 つのノンパラメトリックモデルを用いた平均値関数の推定結果を表したものである．これらの結果から，(i) ベストパラメトリックモデルは累積ソフトウェアフォールト数の平均的振る舞いを上手く表現出来てはいるが，微細な振る舞いを捉えることは出来ないこと，(ii) 強度関数推定量は常に累積ソフトウェアフォールト数を過大に推定していること，(iii)

故障率推定量は、データの傾向が正しく推定可能な場合にはよりデータに合致した推定結果を算出していること、つまり、DS1 に対しては S 字型推定量が最もデータに適合しており、IFR や DFR 推定量は累積フォールト数を過大もしくは過小に見積もっていることが分かる。(i) については、パラメトリック NHPP モデルは選択された確率分布関数の形状に強く依存する傾向があること、(iii) については、過去のデータ傾向を考慮することで、故障率の CNPML 推定は累積フォールト数の細かな振る舞いを推定することが可能であることが理由として挙げられる。また、テスト時間が比較的短い場合には、DFR 推定量は IFR 推定量よりも常に大きな値を取っているが、これら二つの推定量の差はソフトウェアテストの終盤では小さくなっていくことが分かる。

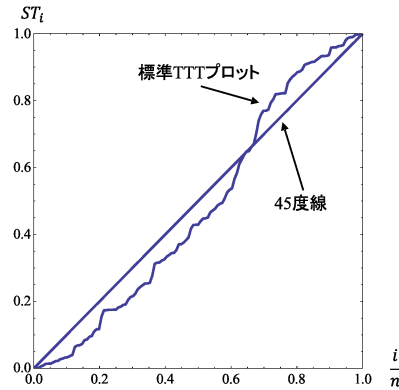


図 1: 標準 TTT プロット (DS1)

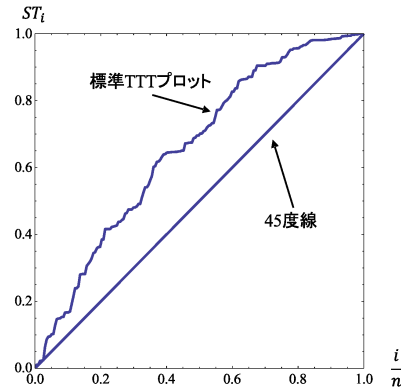


図 2: 標準 TTT プロット (DS5)

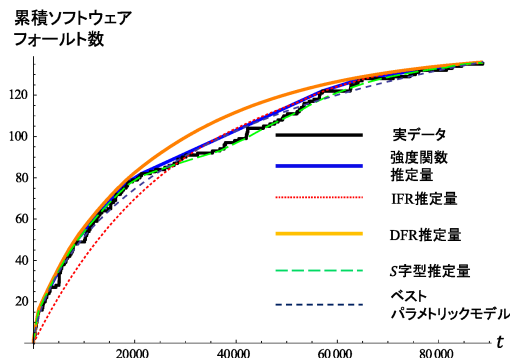


図 3: 平均値関数の振る舞い (DS1)

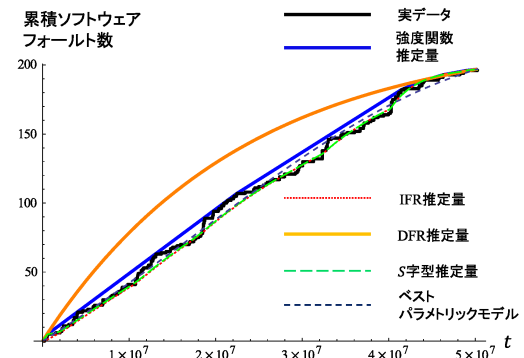


図 4: 平均値関数の振る舞い (DS5)

データへの適合性を定量的に測るために、適合度の指標として平均二乗誤差 (MSE) と最大対数尤度 (MLL) を用いる。

$$\text{MSE} = \frac{\sqrt{\sum_{i=1}^n \{\Lambda(t_i) - i\}^2}}{n}, \quad (21)$$

$$\text{MLL} = \sum_{i=1}^n \log \lambda(t_i) - \Lambda(t_n). \quad (22)$$

表 2 は適合度の計算結果を表したものである。 S 字型推定量は 3 つの年齢特性 (IFR, DFR 及び S 字型) の中で最小の MSE を示している。 S 字型推定量は強度関数推定量と比較しても、ほとんど全てのデータセットにおいて小さな MSE を示していることが分かる。 S 字型推定量とベストパラメトリックモデルとの比

表 2: 適合性指標

(i) MSE

	IFR	DFR	S 字型	強度関数	ベスト
DS1	0.701	0.520	0.145	0.196	0.222
DS2	1.181	0.637	0.430	0.417	0.534
DS3	2.116	0.764	0.540	0.374	0.814
DS4	0.360	2.138	0.256	0.624	0.630
DS5	0.291	2.098	0.263	0.443	0.281
DS6	0.296	1.262	0.236	0.284	0.228

(ii) MLL

	IFR	DFR	S 字型	強度関数	ベスト
DS1	-961.6	-963.9	-958.6	-953.4	-966.1
DS2	-900.2	-910.0	-891.4	-897.0	-908.0
DS3	-2367.7	-2341.8	-2325.5	-2317.9	-2355.4
DS4	-3616.8	-3662.2	-3613.7	-3628.0	-3651.4
DS5	-2611.4	-2668.4	-2625.4	-2638.4	-2645.3
DS6	-1077.3	-1099.4	-1076.4	-1082.5	-1093.5

較では、DS6 を除くすべての場合において、故障率推定量が最小二乗誤差の観点で最も優れたデータ適合性を示していることが分かる。また、DS6 についても、データ適合性に大きな差がないことが確認できる。一方、MLL を用いた場合の結果を見ると、 S 字型推定量及び強度関数推定量の 2 つの CNPML 全てのデータセットにおいて、ベストパラメトリックモデルよりも高いデータ適合性を示していることが分かる。CNPML の自由度はパラメトリック NHPP モデルよりも常に大きいことから、この結果は妥当なものであると言える。しかしながら実用上の観点から言えば、ノンパラメトリック最尤推定量を用いることにより、ソフトウェア信頼性尺度を推定する前にデータ適合性からのモデル選択を必要としない大きな利点を持つ。

4.2 ソフトウェア信頼性尺度

次に、CNPML と 11 種類のパラメトリック NHPP モデルを用いて、いくつかのソフトウェア信頼性尺度を推定する。強度関数推定量では、初期残存フォールト数 ω に依存する残存ソフトウェアフォールト数及びソフトウェア製品内にフォールトが残っていない確率を表すフォールトフリー確率を推定することは出来ない。確率分布関数 $F(t)$ に対して、任意の時刻 t における残存ソフトウェアフォールト数 $R(t)$ とフォールトフリー確率 $FFP(t)$ をそれぞれ

$$R(t) = \hat{\omega}\{1 - \hat{F}(t)\}, \quad (23)$$

$$FFP(t) = \exp(-\hat{\omega}\{1 - \hat{F}(t)\}) \quad (24)$$

と定義する。ここで、 $\hat{F}(t)$ はフォールト検出時間分布の推定値を表す。

表 3 は 6 種類のデータセットを用いたソフトウェア信頼性尺度の推定結果を表している。ここでは、最終フォールト検出時刻 t_n における残存ソフトウェアフォールト数及びフォールトフリー確率を推定した。初期残存フォールト数及び残存ソフトウェアフォールト数の推定結果から、故障率推定量では初期残存フォールト数の推定値は総ソフトウェアフォールト数 n に近い値を取ることが示されており、残存ソフトウェアフォールト数が 0 に近い結果となっていることが分かる。また、IFR 推定量と S 字型推定量が非常に近い。

推定結果を示している一方で、DFR 推定量はそれらと比べてやや大きめの推定結果を示す傾向があることが分かる。故障率推定量に基づくフォールトフリー確率は大きな値を取る傾向にあり、特に IFR 推定量においてその傾向が顕著であった。これに対して、ベストパラメトリックモデルを用いた場合、フォールトフリー確率は小さめの値を取る傾向があることが分かった。

表 3: ソフトウェア信頼性尺度 I

(i) 初期残存フォールト数 (時刻 $t = 0$)

	n	IFR	DFR	S 字型	ベスト
DS1	136	136.33	141.69	136.41	172.90
DS2	129	129.28	134.93	129.40	174.28
DS3	397	397.13	409.11	397.30	439.18
DS4	278	278.22	305.43	278.22	476.36
DS5	197	197.25	221.78	197.25	215.43
DS6	207	207.32	222.39	207.32	256.49

(ii) 残存ソフトウェアフォールト数 (時刻 $t = t_n$)

	IFR	DFR	S 字型	ベスト
DS1	0.33	5.69	0.41	36.88
DS2	0.28	5.93	0.40	8.64
DS3	0.13	12.11	0.30	42.20
DS4	0.22	27.43	0.22	198.08
DS5	0.25	24.78	0.25	18.36
DS6	0.32	15.39	0.32	49.47

(iii) フォールトフリー確率 (時刻 $t = t_n$)

	IFR	DFR	S 字型	ベスト
DS1	7.16E-01	3.38E-03	6.64E-01	9.61E-17
DS2	7.59E-01	2.65E-03	6.73E-01	2.21E-20
DS3	8.81E-01	5.51E-06	7.39E-01	4.71E-19
DS4	8.04E-01	1.22E-12	8.03E-01	9.43E-87
DS5	7.76E-01	1.73E-11	7.77E-01	1.07E-08
DS6	7.23E-01	2.08E-07	7.23E-01	3.26E-22

次に、以下ではフォールト検出時間に関わる 2 つの信頼性尺度を算出した。平均故障時間間隔 (MTBF) は、運用段階におけるソフトウェア故障の発生頻度を表す代表的な評価尺度の一つである。MTBF が長いほどフォールト検出の頻度は小さくなり、ソフトウェアの信頼性が向上しているとみなすことが出来る。しかしながら、有界な平均値関数を持つ NHPP モデルにおいては厳密な意味で MTBF を定義することが出来ないことから、本稿では代替的指標である累積平均故障時間間隔 $MTBF_C(t)$ 及び瞬間平均故障時間間隔 $MTBF_I(t)$ を用いる。

$$MTBF_C(t) = \frac{t}{\Lambda(t)}, \quad (25)$$

$$MTBF_I(t) = \frac{1}{\lambda(t)}. \quad (26)$$

累積 MTBF は単位時間当りの期待ソフトウェアフォールト数の逆数で表され、瞬間 MTBF は時刻 t にお

ける瞬間的な平均フォールト検出時間間隔を表す。

表 4: ソフトウェア信頼性尺度 II

(i) 累積 MTBF(時刻 $t = t_n$)

	IFR	DFR	S 字型	強度関数	ベスト
DS1	652	652	652	652	652
DS2	690	690	690	690	690
DS3	274	274	274	274	274
DS4	197602	197602	197602	197602	197404
DS5	255009	255009	255009	255009	254919
DS6	80	80	80	80	80

(ii) 瞬間 MTBF(時刻 $t = t_n$)

	IFR	DFR	S 字型	強度関数	ベスト
DS1	11307	5155	9221	4116	2303
DS2	276	5551	192	2571	3752
DS3	4855	3064	2028	1629	2255
DS4	9.3E+05	8.5E+05	9.3E+05	4.4E+05	3.4E+05
DS5	3.6E+06	9.3E+05	3.6E+06	1.4E+06	5.3E+05
DS6	663	409	663	332	205

表 4 は 6 種類のデータセットを用いて時刻 $t = t_n$ における累積 MTBF 及び瞬間 MTBF を推定した結果を表す。累積 MTBF に関しては、どのモデルにおいてもよく似た推定結果を示していることが分かる。これに対して、瞬間 MTBF はそれぞれのモデルにおいて非常に異なる値を示していることが分かる。もしデータに最も適合したモデルの推定結果が最も信頼出来る指標であると考えらるならば、S 字型推定量を用いたときの瞬間 MTBF を採用することが最も現実的であると考えられる。

5 まとめと今後の展望

本稿では、NHPP に基づくソフトウェア信頼性モデルに対して、2 つの制約付きノンパラメトリック最尤推定量 (CNPMLE) を提案した。強度関数推定量の主な特徴は尤度関数を最大化するよう区分的線形な強度関数を推定する点にある。また、故障率推定量ではフォールト検出時間分布の故障率をデータから直接的に推定出来る。実際のフォールト検出時刻データを用いた数値例の結果から、故障率の年齢特性を考慮した CNPMLE は尤度の観点から最も高いデータ適合度を示した。更に、ソフトウェア信頼性尺度を正確に推定するために S 字型推定量はほぼ全てのフォールト検出時刻データにおいて、最も有効であることが分かった。

今後の課題としては、ソフトウェア信頼性尺度の区間推定を行うことが挙げられる。具体的には、フォールト検出時刻データを複製するための代表的な統計的手法であるノンパラメトリックブートストラップ法を適用することが考えられる。

参考文献

- [1] R. E. Barlow and R. A. Campo, "Total time on test processes and applications to failure data analysis," *Reliability and Fault Tree Analysis* (R. E. Barlow, J. Fussell and N. D. Singpurwalla, eds.), pp. 451–481, SIAM (1975).
- [2] M. T. Boswell, "Estimating and testing trend in a stochastic process of Poisson type," *Annals of Mathematical Statistics*, **37** (6), pp. 1564–1573 (1966).
- [3] A. Gandy and U. Jensen, "A non-parametric approach to software reliability," *Applied Stochastic Models in Business and Industry*, **20** (3), pp. 3–15 (2004).
- [4] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Transactions on Reliability*, **R-28** (3), pp. 206–211 (1979).
- [5] S. S. Gokhale and K. S. Trivedi, "Log-logistic software reliability growth model," *Proceedings of 3rd IEEE International High-Assurance Systems Engineering Symposium (HASE-1998)*, pp. 34–41, IEEE CPS (1998).
- [6] T. Kaneishi and T. Dohi, "Software reliability modeling and evaluation under incomplete knowledge on fault distribution," *Proceedings of the 7th IEEE International Conference on Software Security and Reliability (SERE-2013)*, pp. 3–12, IEEE CPS (2013).
- [7] A. W. Marshall and F. Proschan, "Maximum likelihood estimation for distributions with monotone failure rate," *Annals of Mathematical Statistics*, **36** (1), pp. 69–77 (1965).
- [8] D. R. Miller and A. Sofer, "A non-parametric software reliability growth model," *IEEE Transactions on Reliability*, **R-40** (3), pp. 329–337 (1991).
- [9] H. Okamura, T. Dohi and S. Osaki, "Software reliability growth models with normal failure time distributions," *Reliability Engineering and System Safety*, **116**, pp. 135–141 (2013).
- [10] H. Okamura and T. Dohi, "SRATS: Software reliability assessment tool on spreadsheet," *Proceedings of The 24th International Symposium on Software Reliability Engineering (ISSRE 2013)*, pp. 100–117. IEEE CPS (2013).
- [11] Z. Wang, J. Wang, and X. Liang, "Non-parametric estimation for NHPP software reliability models," *Journal of Applied Statistics*, **34** (1), pp. 107–119 (2007).