

離散化ソフトウェア信頼度成長モデルに基づいたブートストラップ信頼性評価法

鳥取大学・大学院工学研究科 井上 真二, 山田 茂
Shinji Inoue and Shigeru Yamada
Graduate School of Engineering,
Tottori University

1 はじめに

定量的なソフトウェア信頼性評価は、高信頼性ソフトウェアを開発するために管理的側面から支援する主要な技術の1つとして知られている。特に、ソフトウェア信頼度成長モデル (software reliability growth model: SRGM) [1-3] は、ソフトウェアの実行に伴い観測されるソフトウェア故障発生現象を記述する確率モデルであり、このモデルから導出される種々の信頼性評価尺度に基づいて定量的なソフトウェア信頼性評価を行うことができる。SRGMを適用する際には、一般的に、ソフトウェア信頼度成長過程を特徴づけるモデルパラメータを、観測したソフトウェア故障発生時間データやフォールト発見数データから点推定に基づいて推定する必要がある。しかし、実際のソフトウェアテスト環境では、納期やコストの制約を受け、従来から多く行われている点推定に基づいた SRGM による信頼性評価手法に対して十分な信頼性データが得られない場合があり、このようなデータから得られた信頼性評価結果が実際の状況に対してどれだけ信憑性をもつ結果なのか問題視されることも少なくない。また、このように観測データが少ない状況では、パラメータやソフトウェア信頼性評価尺度に関する推定量の確率分布を考慮しながら区間推定を行うべきであるが、近似的なアプローチを用いても理論や数式に基づく複雑な解析的アプローチを用いる必要があると共に、実際に有効な推定値を得られない場合も考えられる [4]。本研究では、フォールト発見数データの収集方法と整合性をもつ離散化非同次ポアソン過程 (nonhomogeneous Poisson process: NHPP) モデル [5] を一例に取り上げ、計算統計手法の1つであるノンパラメトリックブートストラップ法 [4, 6, 7] に基づいたモデルパラメータや信頼性評価尺度の区間推定法について議論する。

2 離散化指数形 SRGM

本研究では、離散化 NHPP モデル [5] の中でも最もシンプルな構造をもつ離散化指数形 SRGM を取り上げて議論する。まず、離散化 NHPP モデルとは、テスト開始後 n 期目までに発見された総フォールト数を表す離散時間計数過程 $\{N_n; n = 0, 1, 2, \dots\}$ が、

$$\Pr\{N_n = x \mid N_0 = 0\} = \frac{\{\Lambda_n\}^x}{x!} \exp[-\Lambda_n] \quad (x = 0, 1, \dots), \quad (1)$$

に従うものと仮定する SRGM である。式 (1) において、 Λ_n は離散時間 NHPP の平均値関数と呼ばれ、テスト開始後 n 期目までに発見された総期待フォールト数を表す。式 (1) より、離散時間計数過程 $\{N_n; n = 0, 1, 2, \dots\}$ はその平均値関数 Λ_n によって具体的に特徴付けられることがわかる。

離散化指数形 SRGM は、微分方程式で記述される従来の連続時間指数形 SRGM [8] の基本的仮定に基づいて導出される $\Lambda_n \equiv H_n$ に関する差分方程式：

$$H_{n+1} - H_n = \delta\beta(\omega - H_n) \quad (\omega > 0, \beta > 0), \quad (2)$$

によってソフトウェア故障発生現象の平均的挙動が特徴づけられる。ここで、 H_n はテスト開始後 n 期目までに発見された総期待フォールト数、 ω はテスト開始前にソフトウェア内に潜在する総フォールト数の期待値、 β は1個当りのフォールト発見率を表す。また、式 (2) は厳密解 H_n をもち、

$$H_n = \omega[1 - (1 - \delta\beta)^n], \quad (3)$$

のように導出される。なお、 $\delta \rightarrow 0$ のとき、式 (2) と式 (3) はそれぞれ、従来の連続時間指数形 SRGM の基本的仮定を表現する微分方程式およびその厳密解へ収束する。

離散化 NHPP モデルに含まれるパラメータは、差分方程式から導出される回帰式に基づいて最小二乗法を用いて推定される。いま、フォールト発見数データ： $(n, y_n)(n = 1, 2, \dots, N)$ が観測されたとき、離散化指数形 SRGM を適用した場合、式 (2) の差分方程式から単回帰式：

$$C_n = \alpha_0 + \alpha_1 D_n, \quad (4)$$

を得る。ここで、 y_n はテスト開始後 n 期目までに発見された総フォールト数を表し、

$$\begin{cases} C_n = H_{n+1} - H_n \equiv y_{n+1} - y_n, \\ D_n = H_n \equiv y_n, \\ \alpha_0 = \delta\omega\beta, \\ \alpha_1 = -\delta\beta, \end{cases} \quad (5)$$

である。式 (5) より、離散化指数形 SRGM のパラメータは、それぞれ、推定された回帰パラメータ $\hat{\alpha}_0$ および $\hat{\alpha}_1$ から、

$$\begin{cases} \hat{\omega} = -\hat{\alpha}_0/\hat{\alpha}_1, \\ \hat{\delta}\hat{\beta} = -\hat{\alpha}_1, \end{cases} \quad (6)$$

のように導出される。

また、式 (1) と式 (3) から、定量的なソフトウェア信頼性評価に有効な種々の信頼性評価尺度を導出することができる。なお、本研究では、次に示す 2 つの信頼性評価尺度について議論する。期待残存フォールト数 M_n は、テスト開始後 n 期目においてソフトウェア内に残存する総期待フォールト数を表し、式 (1) と式 (3) から、

$$\begin{aligned} M_n &\equiv E[N_\infty - N_n] = \omega - \Lambda_n \\ &= \omega(1 - \delta\beta)^n, \end{aligned} \quad (7)$$

のように導出することができる。さらに、離散時間ソフトウェア信頼度関数 $R(n, h)$ は、テストが開始後 n 期目まで進行しているとき、その後の時間区間 $(n, n+h](h = 1, 2, \dots)$ においてソフトウェア故障がしない確率として定義され、

$$\begin{aligned} R(n, h) &\equiv \Pr\{N_{n+h} - N_n = 0 \mid N_n = x\} \\ &= \exp[-\{\Lambda_{n+h} - \Lambda_n\}] \\ &= \exp[-H_h(1 - \delta\beta)^n], \end{aligned} \quad (8)$$

のように導出される。

3 ノンパラメトリックブートストラップ法の適用

離散化 NHPP モデルに基づいた信頼性解析へのブートストラップ法の適用について、前述した離散化指数形 SRGM を取り上げ議論する。離散化指数形 SRGM を適用した場合におけるノンパラメトリックブートストラップ推定手順を以下に示す：

- S1 観測データ $(n, y_n)(n = 1, 2, \dots, N)$ に基づいて回帰パラメータ α_0 と α_1 を推定し、それら推定結果を、それぞれ、 $\hat{\alpha}_{0(0)}$ と $\hat{\alpha}_{1(0)}$ と表記する。
- S2 各観測時点における残差 $\hat{d}_i$ を、 $\hat{d}_i = C_i - (\hat{\alpha}_{0(0)} + \hat{\alpha}_{1(0)} D_i)(i = 1, 2, \dots, N-1)$ に基づいて算出する。
- S3 残差 $\{\hat{d}_1, \hat{d}_2, \dots, \hat{d}_{N-1}\}$ がそれぞれ独立かつ同一の分布に従うものと仮定すると共に、これらを昇順に並べ、経験分布関数 $\hat{F}$ を生成する。
- S4 ブートストラップ反復回数の上限值 B を設定する。また、 $b(b = 1, 2, \dots, B)$ を反復回数とする。

- S5 経験分布関数 $\hat{F}$ から重複を許しながら残差に関するブートストラップ標本（ブートストラップ残差）， $\hat{d}_{(b)}^* = \{\hat{d}_{b,1}^*, \hat{d}_{b,2}^*, \dots, \hat{d}_{b,N-1}^*\}$ を生成する。
- S6 ブートストラップ残差から， $C_{b,i}^* = \hat{\alpha}_{0(0)} + \hat{\alpha}_{1(0)} D_i + \hat{d}_{b,i}^*$ に基づいて回帰分析に必要なブートストラップ標本 $z_{(b)}^* = \{(y_1, C_{b,1}^*), (y_2, C_{b,2}^*), \dots, (y_{N-1}, C_{b,N-1}^*)\}$ を生成する。
- S7 ブートストラップ標本 $z_{(b)}^*$ から $\alpha_{0(b)}^*$ と $\alpha_{1(b)}^*$ を推定する。
- S8 $\hat{\alpha}_{0(b)}^*$ と $\hat{\alpha}_{1(b)}^*$ を用いて， $\hat{\omega}_{(b)}^* = -\hat{\alpha}_{0(b)}^* / \hat{\alpha}_{1(b)}^*$ ， $\hat{\delta\beta}_{(b)}^* = -\hat{\alpha}_{1(b)}^*$ を求める。
- S9 各種ソフトウェア信頼性評価尺度を推定する。
- S10 $b < B$ ならば $b = b + 1$ として，S5 へ移動する。
- S11 $\hat{\omega}$ ， $\hat{\delta\beta}$ ，および各種ソフトウェア信頼性評価尺度に関する B 組のデータを得る。

4 ブートストラップ信頼区間

今回議論した手法によって得られたブートストラップ分布に基づいて，モデルパラメータや各種ソフトウェア信頼性評価尺度の区間推定を行う。なお，ここでは， θ を興味のある一次元パラメータ， $\hat{\theta}$ を θ の推定量として，以下の3つのブートストラップ信頼区間を取り上げて議論する。

まず，基本ブートストラップ信頼区間 [9] は， $\hat{\theta} - \theta$ の分布をブートストラップ統計量 $\hat{\theta}^* - \hat{\theta}$ の分布で近似する手法である。すなわち， $\hat{\theta} - \theta$ の分布における α および $(1 - \alpha)$ 分位点 v_α および $v_{1-\alpha}$ を，それぞれ $\hat{\theta}_{[B\alpha]}^* - \hat{\theta}$ および $\hat{\theta}_{[B(1-\alpha)]}^* - \hat{\theta}$ で近似すると，

$$\begin{aligned} 1 - 2\alpha &= \Pr\{v_\alpha \leq \hat{\theta} - \theta \leq v_{1-\alpha}\} \\ &= \Pr\{\hat{\theta} - v_{1-\alpha} \leq \theta \leq \hat{\theta} - v_\alpha\} \\ &= \Pr\{2\hat{\theta} - \hat{\theta}_{[B(1-\alpha)]}^* \leq \theta \leq 2\hat{\theta} - \hat{\theta}_{[B\alpha]}^*\}, \end{aligned}$$

を得る。これより， $100(1 - 2\alpha)\%$ 信頼区間は，

$$[2\hat{\theta} - \hat{\theta}_{[B(1-\alpha)]}^*, 2\hat{\theta} - \hat{\theta}_{[B\alpha]}^*], \quad (9)$$

のように求められる。ここで， $\hat{\theta}_{[B\alpha]}^*$ は，大きさ順に並べた B 個の $\hat{\theta}_{(b)}^*$ の中で $B\alpha$ 番目の値を示す。次に，標準正規ブートストラップ信頼区間 [9] は， $\hat{\theta} - \theta$ の分布を $N(0, \text{SD}[\hat{\theta}]^2)$ で正規近似して得られる信頼区間であり，

$$1 - 2\alpha = \Pr\left\{z_\alpha \leq \frac{\hat{\theta}^* - \hat{\theta}}{\text{SD}[\hat{\theta}]} \leq z_{1-\alpha}\right\},$$

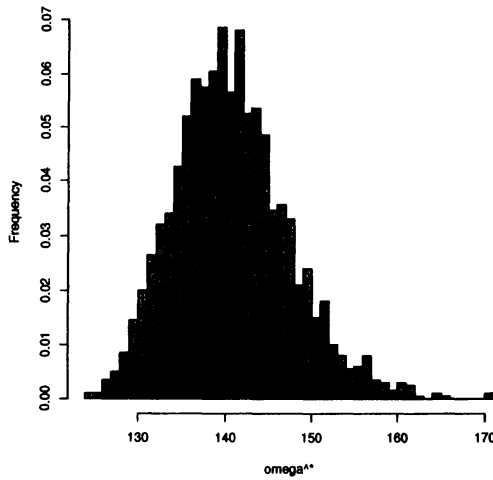
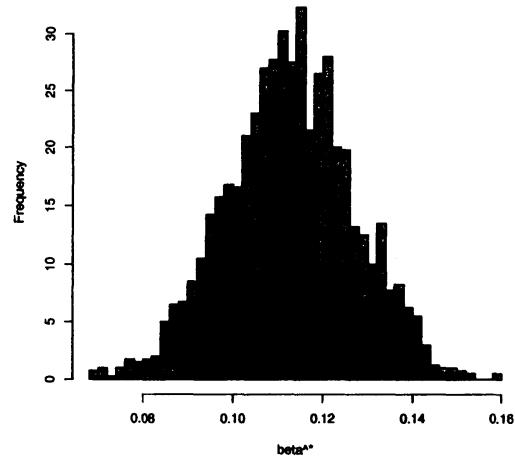
として表される。したがって， $100(1 - 2\alpha)\%$ 信頼区間は，

$$[\hat{\theta} - w_{1-\alpha} \text{SD}[\hat{\theta}], \hat{\theta} - w_\alpha \text{SD}[\hat{\theta}]], \quad (10)$$

として与えられる。ここで， $w_\alpha = \Phi^{-1}(\alpha)$ である。パーセンタイル信頼区間 [9] は， $\hat{\theta}^*$ の反復値 $\hat{\theta}_{(1)}^*, \hat{\theta}_{(2)}^*, \dots, \hat{\theta}_{(B)}^*$ に基づいて得られる信頼区間であり， $100(1 - 2\alpha)\%$ 信頼区間は，

$$[\hat{\theta}_{[B\alpha]}^*, \hat{\theta}_{[B(1-\alpha)]}^*], \quad (11)$$

として求められる。

図 1: $\hat{\omega}$ に対するブートストラップ分布.図 2: $\hat{\beta}$ に対するブートストラップ分布.

5 適用例

実際のテスト工程において観測されたフォールト発見数データを用いて、今回議論した離散化指数形 SRGM に基づいたソフトウェア信頼性評価のためのノンパラメトリックブートストラップ法の適用例を示す。本研究では、適用する実測データとして 25 組から成るフォールト発見数データ: $(n, y_n) (n = 1, 2, \dots, 25; y_{25} = 136)$ [5] を用いると共に、ブートストラップ反復値を $B = 2000$ とする。

まず、式 (4) の回帰式を用いた最小二乗法に基づいて $\hat{\alpha}_{0(0)} = 139.9564$ および $\hat{\alpha}_{1(0)} = 0.1133109$ をそれぞれ得る。これより、今回議論したブートストラップ推定手順に従って、2000 個のブートストラップ標本 $\{z_{(1)}^*, z_{(2)}^*, \dots, z_{(2000)}^*\}$ 、および、これらによって得られる ω および β に関するブートストラップ推定値も得られる。

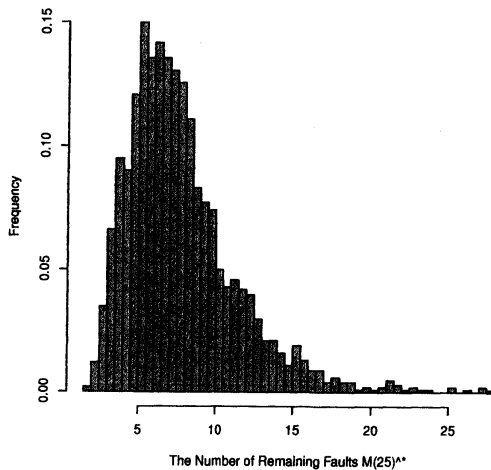
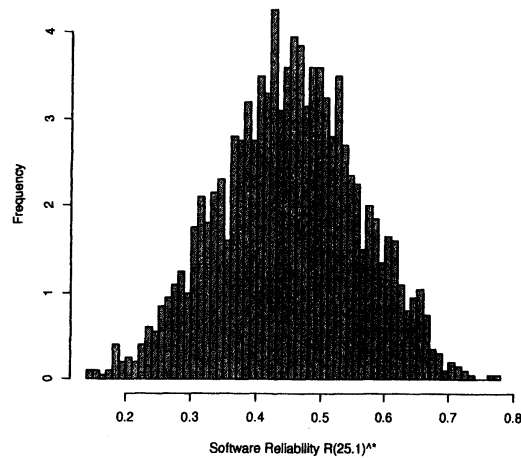
図 1 および図 2 に、それぞれ、パラメータ推定量 $\hat{\omega}$ および $\hat{\beta}$ に関する確率分布を捉えるためのブートストラップ分布を示す。また、図 3 および図 4 に、それぞれ、テスト開始後 $n = 25$ 期目における期待残存フォールト数 $\hat{M}_{25}$ とテスト開始後 $n = 25$ 期目にテストを終了して同じ環境の下で 1 期間ソフトウェアを実行させたときのソフトウェア信頼度 $\hat{R}(25, 1)$ に対するブートストラップ分布もそれぞれ示す。なお、図 3 および図 4 のブートストラップ分布を構成する $M_{25(b)}^*$ および $R_{(b)}^*(25, 1)$ は、それぞれ、式 (7) および式 (8) から、

$$\hat{M}_{25(b)}^* = \frac{\hat{\alpha}_{0(b)}^*}{\hat{\alpha}_{1(b)}^*} (1 + \hat{\alpha}_{1(b)}^*)^{25}, \quad (12)$$

$$\hat{R}_{(b)}^*(25, 1) = \exp \left[\frac{\hat{\alpha}_{0(b)}^*}{\hat{\alpha}_{1(b)}^*} \left\{ 1 - (1 + \hat{\alpha}_{1(b)}^*)^1 \right\} (1 + \hat{\alpha}_{1(b)}^*)^n \right], \quad (13)$$

によって求められる。

次に、求められた各ブートストラップ分布における平均値と標準偏差をそれぞれ表 1 に示す。さらに、表 2 に、図 1 から図 4 に示したパラメータ推定量およびソフトウェア信頼性評価尺度に対するブートストラップ分布から、今回議論した 3 つのブートストラップ信頼区間に基づいた区間推定結果 (95% 信頼区間, $\alpha = 0.025$) を示す。表 2 から、 $\hat{M}_{25}$ に対する基本および標準正規ブートストラップ信頼区間に基づいた区間推定結果において、非現実的な推定結果を得ていることがわかる。これは、基本および標準正規ブートストラップ信頼区間が対称性を有した推定量の確率分布を前提においた信頼区間導出手法であるためと考えられる。一方、パーセンタイル信頼区間は、ブートストラップ分布自体の分位点をそのまま利用しているた

図 3: $\widehat{M}_{25}$ に対するブートストラップ分布.図 4: $\widehat{R}(25,1)$ に対するブートストラップ分布.

め, $\widehat{M}_{25}$ が取り得る値として現実的な区間推定結果を得ていることがわかる.

6 おわりに

本研究では, 現実のフォールト発見数データの収集方法と整合性をもつ離散化 NHPP モデルの中でも比較的シンプルなモデル構造をもつ離散化指数形 SRGM を取り上げ, 計算統計手法の 1 つであるブートストラップ法に基づいたモデルパラメータやソフトウェア信頼性評価尺度の区間推定手法について議論した. 特に, 離散化指数形モデルのパラメータは, モデルを特徴付ける差分方程式から得られる回帰式に基づいた最小二乗法によって比較的容易に推定できるため, 独立かつ同一の分布に従うものと仮定した回帰残差から構成される経験分布関数 $\widehat{F}$ に基づいたノンパラメトリックブートストラップ法を適用した推定手順を示した. 適用例においては, モデルパラメータとソフトウェア信頼性評価尺度に対するブートストラップ分布からそれらの区間推定を行うため, 3 つのブートストラップ信頼区間に基づいた区間推定を行った. その結果, 非対称性を有するブートストラップ分布に従う推定量については, 現実的でない区間推定結果を得ることを確認した. 今後は, ブートストラップ- t 法や BCa 法など, 推定量の分布の分散に関する情報や, 推定量の分布の非対称性やバイアスと歪みの大きさなどを考慮したブートストラップ信頼区間を適用しながら, 信頼区間の精度改善に向けた議論を行う必要がある.

表 1: ブートストラップ分布の平均と標準偏差.

	Mean	Standard Deviation
$\widehat{\alpha}_0$	15.79443	1.500837
$\widehat{\alpha}_1$	-0.1127263	0.01422163
$\widehat{\omega}$	140.7685	6.609685
$\widehat{\beta}$	0.1127263	0.01422163
$\widehat{M}_{25}$	7.761994	3.59502
$\widehat{R}(25,1)$	0.4514124	0.1072053

表 2: ブートストラップ信頼区間に基づいたパラメータと信頼性評価尺度の区間推定結果.

		Lower	Upper
$\hat{\omega}$	Basic	124.0544	150.3849
	Standard Normal	127.0015	152.9114
	Percentile	129.5280	155.8585
$\hat{\beta}$	Basic	0.08618762	0.141511
	Standard Normal	0.08543655	0.1411854
	Percentile	0.08511091	0.1404343
$\widehat{M}_{25}$	Basic	-2.835469	10.84297
	Standard Normal	-0.1235227	13.96896
	Percentile	3.002461	16.68090
$\widehat{R}(25, 1)$	Basic	0.2567593	0.6711605
	Standard Normal	0.2431593	0.6665069
	Percentile	0.2416085	0.6560098

謝辞

本研究の一部は、文部科学省科学研究費補助金 基盤研究 (C) (課題番号 22510150) の援助を受けたことを付記する。

参考文献

- [1] J.D. Musa, D. Iannio, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*. McGraw-Hill, New York, 1987.
- [2] H. Pham, *Software Reliability*. Springer-Verlag, Singapore, 2000.
- [3] 山田茂, ソフトウェア信頼性の基礎 —モデリングアプローチ, 共立出版, 東京, 2011.
- [4] T. Kaneishi and T. Dohi, "Parametric bootstrapping for assessing software reliability measures," *Proc. 17th IEEE Pacific Rim Intern. Symp. Dependable Computing*, pp. 1–9, 2010.
- [5] S. Inoue and S. Yamada, "Discrete software reliability assessment with discretized NHPP models," *Computers & Mathematics with Applications: An International Journal*, 51(2), pp. 161–170, 2006.
- [6] B. Efron, "Bootstrap methods: another look at the jackknife," *The Annals of Statistics*, 7(1), pp. 1–26, 1979.
- [7] M. Kimura, "A study on bootstrap confidence intervals of software reliability measures based on an incomplete gamma function model," in *Advanced Reliability Modeling II*, T. Dohi and W.Y. Yun Eds., World Scientific, Singapore, pp. 419–426, 2006.
- [8] A. 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.
- [9] M.L. Rizzo, *Statistical Computing with R*, Chapman and Hall/CRC, Boca Raton, 2008.