

Comparison of Estimation Algorithms for NHPP-Based Software Reliability Models

岡村寛之, 渡部保博, 土肥正

Hiroyuki Okamura, Yasuhiro Watanabe and Tadashi Dohi

Department of Information Engineering, Graduate School of Engineering,
Hiroshima University, Japan

1 はじめに

現在, 高度情報化社会あるいは情報ネットワーク社会であると言われているように, コンピュータシステムはあらゆる分野で重要な役割を担っている. 特に金融や通信といった社会の基盤となる分野において, 業務支援・情報戦略などの主要業務におけるコンピュータシステムの必要性が加速度的に増大している. コンピュータシステムへの依存性が高まるにつれて, コンピュータシステムの構成要素であるソフトウェアの信頼性確保に対する十分な検討が必要とされている.

現在までにソフトウェアの信頼性を定量的に評価する方法が数多く提案されている. 一般的によく知られている手法は, テスト段階あるいは運用段階におけるフォールト発見事象を確率過程を用いて表現する手法であり, フォールト除去作業によってソフトウェアの信頼度が向上することから, これらの確率過程はソフトウェア信頼度成長モデルと呼ばれている. 代表的なソフトウェア信頼度成長モデルとしては, Jelinski-Moranda モデル [1] や指数形信頼度成長モデル [2] 等がある.

本論文では, 非同時ポアソン過程 (NHPP) に基づいたソフトウェア信頼性モデルにおけるパラメータ推定手法として, EM アルゴリズム [3,4,5,6] に基づいた推定手法を提案し, 他の推定アルゴリズムとの性能比較を行う.

2 NHPP に基づいたソフトウェア信頼性モデル

NHPP によるソフトウェア信頼度成長モデルでは, 任意の時刻までに発見されるフォールト数が次の確率関数に従う計数過程 $\{N(t), t \geq 0\}$ によって表現される.

$$\Pr\{N(t) = n\} = \frac{\{\Lambda(t)\}^n}{n!} \exp\{-\Lambda(t)\}, \quad n = 0, 1, 2, \dots \quad (1)$$

ここで $\Lambda(t)$ は NHPP の平均値関数と呼ばれ, 時間区間 $(0, t]$ において発見される総期待フォールト数 $E[N(t)]$ を表す. さらに, $\lambda(t) = d\Lambda(t)/dt$ は NHPP の強度関数と呼ばれ, テスト時刻 t における瞬間フォールト発見率 (ソフトウェア故障率) を表す. NHPP で記述されるソフトウェア信頼度成長モデルでは, 平均値関数 $\Lambda(t)$ の設定により様々なパターンを表現することが可能である. 特に有限のフォールトに対するデバッグ過程を記述するモデルは, 一般化順序統計量モデルによって統一的な取扱いが可能であることが知られている [7,8].

一般化順序統計量モデルは, 次の仮定の下で議論される.

(仮定 A) ソフトウェアのテスト中にソフトウェア故障が発生した場合, その原因となるフォールトは瞬間的に発見・除去される.

(仮定 B) プログラム中に含まれる初期フォールト数 N_0 は平均 $\omega (> 0)$ のポアソン分布に従う.

(仮定 C) ソフトウェア故障はおのおの独立かつ時間に関してランダムに発生し, 各ソフトウェアフォールトの発見時刻は確率分布 $F(t)$, 密度関数 $f(t)$, $t \geq 0$ に従う非負で連続の確率変数によって記述される.

いま, $\{X_i, i = 1, \dots, N_0\}$ をソフトウェア故障の発生時刻を表す確率変数列として定義する. このとき, 仮定 C は X_i が互いに独立で同一の確率分布 $F(t)$ に従う確率変数であることを意味する. 初期時刻 $t = 0$ においてソフトウェア内に n 個のフォールトが潜在している条件の下で, 時刻 t までに m 個のフォールトが発見される確率は

$$\Pr\{N(t) = m | N_0 = n\} = \binom{n}{m} \{F(t)\}^m \{1 - F(t)\}^{n-m} \quad (2)$$

となる。仮定 B より初期フォールト数はパラメータ ω のポアソン分布に従うため、時刻 t までに発見されるフォールト数の確率分布は

$$\Pr\{N(t) = m\} = \frac{\{\omega F(t)\}^m}{m!} \exp\{-\omega F(t)\} \quad (3)$$

と表現される。これは平均値関数 $\omega F(t)$ をもつ NHPP と等価であり、有限フォールトをもつ既存のソフトウェア信頼性モデルに対する一般的な表現となっている。

3 パラメータ推定

ソフトウェア信頼性モデルを実際のソフトウェア開発工程で用いるためには、ソフトウェアフォールトデータを用いてモデル内部のパラメータ (θ で表記する) を推定する必要がある。最も一般的な推定手法は最尤法であり、最尤法によるパラメータの推定値は時刻 t までに観測された $m (> 0)$ 個のフォールトデータ $\mathcal{D}_t = (x_1, \dots, x_m)$ に対する対数尤度関数

$$\log L_{\text{NHPP}}(\theta|\mathcal{D}_t) = \sum_{i=1}^m \log \lambda(x_i) - \Lambda(x_m) \quad (4)$$

を最大にするパラメータとして決定される。つまり、以下の非線形連立方程式で表現される同時尤度方程式を解くことによってパラメータの推定値を算出する。

$$\frac{\partial}{\partial \theta} \log L_{\text{NHPP}}(\theta|\mathcal{D}_t) = 0. \quad (5)$$

上記の非線形連立方程式を解く方法として、最も有名なものは Newton 法である。これは、対数尤度関数の 1 階微分と 2 階微分をそれぞれ

$$u(\theta) = \frac{\partial}{\partial \theta} \log L_{\text{NHPP}}(\theta|\mathcal{D}_t), \quad (6)$$

$$H(\theta) = \frac{\partial^2}{\partial \theta^2} \log L_{\text{NHPP}}(\theta|\mathcal{D}_t) \quad (7)$$

とおくと、現在の推定値 θ_0 から

$$\hat{\theta} = \theta_0 - H^{-1}(\theta_0)u(\theta_0) \quad (8)$$

の更新を繰り返すことで最尤推定値を算出する。Newton 法において $-H^{-1}(\cdot)$ を Fisher の情報行列の逆行列にしたものが Fisher's scoring 法と呼ばれる方法である。

Newton 法や Fisher's scoring 法の欠点は式 (5) の非線形連立方程式の解が必ずしもソフトウェア信頼性モデルにおいて有効な解となり得ない点にある。つまり、推定に用いる観測データの種類によっては、パラメータに対する (暗黙の) 制約を違反する可能性がある。

この問題を回避するため、文献 [9] では NHPP に基づいたソフトウェア信頼性モデルのパラメータ推定に EM アルゴリズムを用いた手法を提案している。以下ではその概要を述べる。

EM アルゴリズムとは、観測不可能な変数を含む推定問題において用いられる手法である。いま、一般的に確率変数 X の標本値が観測不可能であり、任意の可測関数 $u(\cdot)$ に対して確率変数 $X_{\text{obs}} = u(X)$ が観測可能であるとき、パラメータ集合 ξ を推定する問題を考える。確率変数 X が確率密度関数 $f(\cdot; \xi)$ をもつとき、EM アルゴリズムの $k+1$ 番目のステップにおける推定値は k 番目のステップで推定されたパラメータを適用したときの期待対数尤度を最大にするパラメータによって決定される。すなわち

$$\hat{\xi}^{(k+1)} = \underset{\xi}{\operatorname{argmax}} \left\{ E[\log f(X; \xi) | x_{\text{obs}}; \hat{\xi}^{(k)}] \right\}. \quad (9)$$

ここで、 x_{obs} と $\hat{\xi}^{(k)}$ はそれぞれ観測された標本値と k 番目のステップにおけるパラメータの推定値であり、 $\operatorname{argmax}\{\cdot\}$ は括弧中の値が最大になる引数を返す関数である。また、 $E[\cdot; \xi]$ はパラメータ ξ を用いたときの期待値演算子を表す。

一般化順序統計量モデルのパラメータを推定するために、総数が N_0 個の確率変数 $X_1, \dots, X_{N_0}$ (N_0 は観測不可能な確率変数) の順序統計量 $X_{[1]} < \dots < X_{[N_0]}$ を考える。いま、ソフトウェア内に潜在するフォー

ルトの総数が $N_0 = n$ という条件の下で、すべてのフォールトに対する発見時刻データ $D_\infty = (x_1, \dots, x_n)$ が与えられたならば、その尤度関数および対数尤度関数は

$$L(\omega, \theta | D_\infty) = \omega^n \exp\{-\omega\} \prod_{i=1}^n f(x_i; \theta), \quad (10)$$

$$\log L(\omega, \theta | D_\infty) = n \log \omega - \omega + \sum_{i=1}^n \log f(x_i; \theta). \quad (11)$$

となる。従って、ソフトウェア内に潜在するすべてのフォールトの発見時刻データが与えられた場合、パラメータ ω の推定値は通常の最尤法より観測されたフォールトの総数、つまり $\hat{\omega} = n$ となる。また、各ソフトウェアフォールトの発見時間分布に関するパラメータ θ の推定量は対数尤度関数 $\sum_{i=1}^n \log f(x_i; \theta)$ を最大にするものとして与えられる。

EM アルゴリズムは一部のフォールトの発見時刻に対する観測値が得られた下で構築される。すなわち、 $m < n$ に対して不完全なフォールトデータ $D_t = (x_1, \dots, x_m)$ が得られた時の EM アルゴリズムを考える。式 (9) と式 (11) から、一般化順序統計量モデルにおける期待対数尤度は

$$\log L_E(\omega, \theta | D_t) = E[N_0 | D_t; \omega, \theta] \log \omega - \omega + E \left[\sum_{i=1}^{N_0} \log f(X_i; \theta) \middle| D_t; \omega, \theta \right] \quad (12)$$

であり、これを最大にする ω, θ をパラメータの更新値とする。従って、EM アルゴリズムによる $k+1$ ステップの更新式は

$$\hat{\omega}^{(k+1)} = E[N_0 | D_t; \hat{\omega}^{(k)}, \hat{\theta}^{(k)}], \quad (13)$$

$$\hat{\theta}^{(k+1)} = \underset{\theta}{\operatorname{argmax}} \left\{ E \left[\sum_{i=1}^{N_0} \log f(X_i; \theta) \middle| D_t; \hat{\omega}^{(k)}, \hat{\theta}^{(k)} \right] \right\} \quad (14)$$

として与えられる。ここで、各フォールト発見時刻分布に具体的な確率分布を適用することでパラメータ推定のための更新式を得ることができる。

4 推定手法の評価

前節で紹介した各推定手法の有効性を、実際のソフトウェア開発工程で得られたフォールトデータを用いて数値的に検証する。実際のソフトウェアフォールトデータとして、文献 [10] に記載されている 6 種類のデータ DS1(データ数 86), DS2(データ数 136), DS3(データ数 207), DS4(データ数 394), DS5(データ数 123), DS6(データ数 97) を用いる。

はじめに、単一のフォールト発見時刻が指数分布に従う場合に関して、Newton 法および Fisher's scoring 法との解の安定性の比較を行う。図 1 および図 2 は DS1 において異なる 2 種類の初期値から推定を行った場合の、各推定手法による解の収束の振る舞いを示したものである。同様に、図 3 および図 4 は DS2 において異なる 2 種類の初期値から推定を行った場合の、各推定手法による解の収束の振る舞いを示したものである。これらの図から、Newton 法および Fisher's scoring 法による推定は初期値に依存しており、常に安定して解を得ることができていない。一方、EM アルゴリズムは全ての初期値に対して常に安定して解を得ていることがわかる。

次に各推定手法の性能を定量的に評価するために、100 個の初期値 ($\omega = 50, 100, \dots, 500$, $\lambda = 0.1\text{E-}03, 0.2\text{E-}03, \dots, 1.0\text{E-}03$) からそれぞれのアルゴリズムを 10 回試行したときの推定値を算出し、実際の最尤推定値との相対誤差を比較した。相対誤差は以下の式で定義される。

$$(\text{相対誤差}) = \frac{(\text{推定値}) - (\text{最尤推定値})}{(\text{最尤推定値})} \times 100(\%). \quad (15)$$

表 1-3 は、単一のフォールト発見時刻が指数分布、二次のガンマ分布、形状パラメータ 2 のワイブル分布に従う場合の結果を示している。特に推定値を得る過程でパラメータの制約違反が発生した場合、その時点で計算を終了し、相対誤差の評価には用いていない。また、表中の convergence は制約違反が発生しなかった割合を示している。これらの表から、前述の結果と同様に、EM アルゴリズムによる推定手法では安定して解が得られるが、全体的に収束スピードが遅い（相対誤差が大きい）という性質があることがわかる。他方 Newton 法と Fisher's scoring 法に関しては、最終的に推定値が算出できるかどうかは初期値に大きく依存していることがわかる。

5 まとめ

本論文では、非同時ポアソン過程 (NHPP) に基づいたソフトウェア信頼性モデルにおけるパラメータ推定手法として、Newton 法, Fisher's scoring 法および EM アルゴリズムに基づいた推定手法を紹介し、それらの性能を数値的に検証した。

6 参考文献

- [1] Jelinski, Z. and Moranda, P. B., Software reliability research, In *Statistical Computer Performance Evaluation* (Ed. by Freiberger, W.), pp. 465–484, Academic Press, New York (1972).
- [2] Goel, A. and Okumoto, K., Time-dependent error-detection rate model for software reliability and other performance measures, *IEEE Trans. Reliab.*, Vol. R-28, pp. 206–211 (1979).
- [3] Dempster, A. P., Laird, N. M. and Rubin, D. B., Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Statist. Soc. Ser. B*, Vol. 39, pp. 1–38 (1977).
- [4] Wu, C. F. J., On the convergence properties of the EM algorithm, *Ann. Statist.*, Vol. 11, pp. 95–103 (1983).
- [5] Redner, R. A. and Walker, H. F., Mixture densities, maximum likelihood and the EM algorithm, *SIAM Review*, Vol. 26, pp. 195–239 (1984).
- [6] 宮川雅巳, EM アルゴリズムとその周辺, 応用統計学, Vol. 16, pp. 1–19 (1987).
- [7] Langberg, N. and Singpurwalla, N. D., Unification of some software reliability models, *SIAM J. Sci. Comput.*, Vol. 6, pp. 781–790 (1985).
- [8] Joe, H., Statistical inference for general order statistics and nonhomogeneous Poisson process software reliability models, *IEEE Trans. Software Eng.*, Vol. SE-15, pp. 1485–1490 (1989).
- [9] 岡村寛之, 渡部保博, 土肥正, 尾崎俊治, EM アルゴリズムに基づいたソフトウェア信頼性モデルの推定, 電子情報通信学会論文誌 (A), Vol. J85-A, pp. 442–450 (2002).
- [10] Lyu, M. R. (ed.), *Handbook of Software Reliability Engineering*, McGraw-Hill, New York, NY (1996).

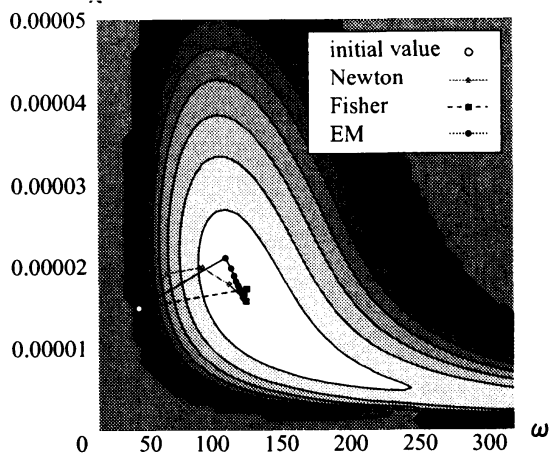


Figure 1: Behavior of estimats for DS1 ($\{\omega, \lambda\} = \{30, 0.000015\}$)

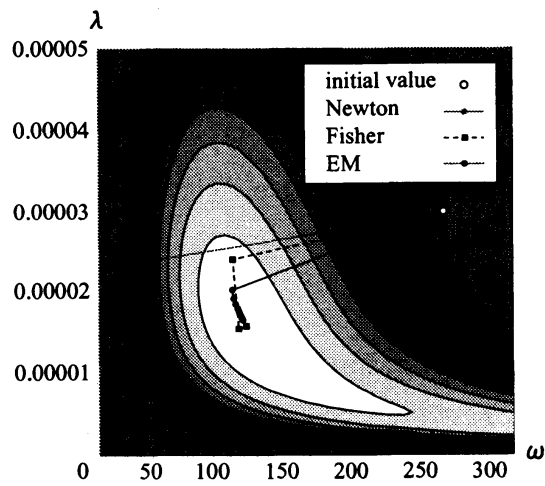


Figure 2: Behavior of estimats for DS1 ($\{\omega, \lambda\} = \{250, 0.00003\}$)

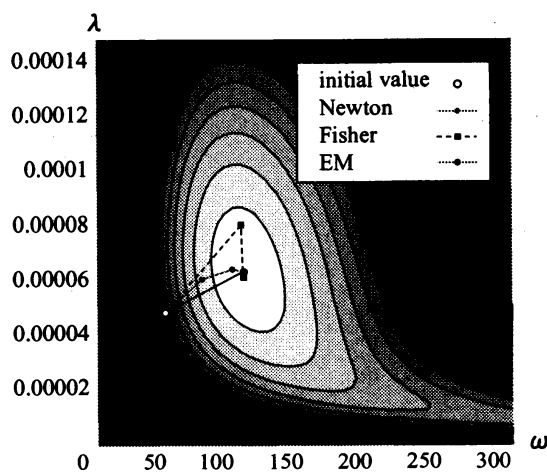


Figure 3: Behavior of estimats for DS2 ($\{\omega, \lambda\} = \{50, 0.00005\}$).

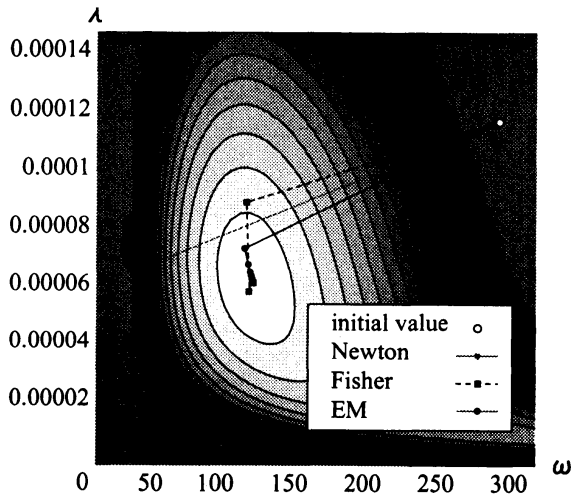


Figure 4: Behavior of estimats for DS2 ($\{\omega, \lambda\} = \{275, 0.00012\}$).

Table 1: Comparison with estimation methods (Exponential SRM).

data set (MLE: ω, λ)	method	convergence	relative errors (ω, λ)
DS1 (107.59, 1.56E-05)	EM	100%	(6.60%, 10.17%)
	Newton	19%	(8.54%, 6.81%)
	Fisher	66%	(0.24%, 0.23%)
DS2 (142.88, 3.42E-05)	EM	100%	(2.54%, 3.81%)
	Newton	44%	(0.04%, 0.65%)
	Fisher	82%	(0.05%, 2.85%)
DS3 (256.76, 9.86E-05)	EM	100%	(33.50%, 27.51%)
	Newton	97%	(0.71%, 1.06%)
	Fisher	72%	(0.07%, 1.07%)
DS4 (401.06, 4.22E-05)	EM	100%	(0.00%, 0.00%)
	Newton	84%	(2.11%, 1.58%)
	Fisher	64%	(1.03%, 1.40%)
DS5 (133.13, 3.90E-05)	EM	100%	(0.00%, 0.00%)
	Newton	45%	(1.67%, 1.93%)
	Fisher	90%	(0.03%, 9.61%)
DS6 (118.23, 1.37E-04)	EM	100%	(77.79%, 48.69%)
	Newton	81%	(0.33%, 0.24%)
	Fisher	78%	(0.23%, 1.43%)

Table 2: Comparison with estimation methods (Gamma SRM).

data set (MLE: ω, λ)	method	convergence	relative errors (ω, λ)
DS1 (90.06, 4.74E-05)	EM	100%	(17.61%, 24.90%)
	Newton	39%	(34.53%, 2.87%)
	Fisher	83%	(0.00%, 0.23%)
DS2 (136.99, 7.90E-05)	EM	100%	(0.63%, 0.04%)
	Newton	50%	(0.00%, 0.00%)
	Fisher	84%	(0.00%, 0.00%)
DS3 (216.36, 2.95E-04)	EM	100%	(5.20%, 8.78%)
	Newton	65%	(0.00%, 0.00%)
	Fisher	26%	(0.03%, 1.97%)
DS4 (397.28, 8.83E-05)	EM	100%	(0.06%, 0.00%)
	Newton	100%	(0.00%, 0.00%)
	Fisher	77%	(0.00%, 0.00%)
DS5 (129.51, 8.66E-05)	EM	100%	(0.35%, 0.01%)
	Newton	50%	(0.06%, 0.01%)
	Fisher	85%	(0.00%, 0.00%)
DS6 (106.58, 3.64E-04)	EM	100%	(10.69%, 16.36%)
	Newton	37%	(0.42%, 0.04%)
	Fisher	46%	(0.00%, 0.00%)

Table 3: Comparison with estimation methods (Weibull SRM).

data set (MLE: ω, λ)	method	convergence	relative errors (ω, λ)
DS1 (87.25, 4.03E-10)	EM	100%	(0.00%, 0.00%)
	Newton	32%	(0.04%, 0.10%)
	Fisher	0%	(—, —)
DS2 (136.17, 8.48E-10)	EM	100%	(0.00%, 0.00%)
	Newton	46%	(3.34%, 5.47%)
	Fisher	0%	(—, —)
DS3 (209.28, 1.63E-08)	EM	100%	(0.00%, 0.00%)
	Newton	45%	(1.09%, 2.49%)
	Fisher	0%	(—, —)
DS4 (397.02, 8.22E-10)	EM	100%	(0.00%, 0.00%)
	Newton	86%	(0.44%, 1.77%)
	Fisher	0%	(—, —)
DS5 (129.08, 9.24E-10)	EM	100%	(0.00%, 0.00%)
	Newton	48%	(2.72%, 5.08%)
	Fisher	0%	(—, —)
DS6 (105.13, 1.92E-08)	EM	100%	(0.00%, 0.00%)
	Newton	32%	(1.39%, 3.50%)
	Fisher	0%	(—, —)