数理解析研究所講究録
1504巻 2006年 42-51

少数サンプルに対する最適ソフトウェア若化スケジュールの推定

林坂 弘一郎†, 土肥 正†
Koichiro Rinsaka† and Tadashi Dohi†

† 広島大学大学院工学研究科情報工学専攻
E-mail: {rinsaka, dohi}@rel.hiroshima-u.ac.jp

要旨— 本論文では，定常状態におけるアベイラビリティを最大にするソフトウェアシステムの若化スケジュールを決定するためのモデルに対して，少数の障害発生時間データしか得ることができない状況下における推定精度を向上させる統計アルゴリズムを提案する．具体的には，得られた障害発生時間データから核型密度推定により障害発生時間の密度関数及び総試験時間変換の推定量をノンパラメトリックに推定する．これにより少数のデータから直接最適ソフトウェア若化スケジュールを推定する枠組みを提案する．シミュレーション実験により，提案アルゴリズムは障害発生時間データ数が少ない場合においても，従来のアルゴリズムと比較して真の最適解に収束する速度が向上することを示す．

キーワード— ソフトウェア若化，セミマルコフ過程，総試験時間変換，核型密度推定，定常アベイラビリティ

1. はじめに

近年のコンピュータシステムの爆発的な普及により，一旦システムに障害が発生すると社会的に大きな影響を及ぼすことは周知の通りである．特に，コンピュータのシステムダウンによる障害はハードウェア障害よりもむしろソフトウェア障害に起因することが多く，ソフトウェアシステムの信頼性を向上させることは最重要課題となっている．ソフトウェア障害は，(i) ソフトウェアプログラムに含まれる固有フォールトによる障害と (ii) ソフトウェアシステムの経年劣化による障害とに大別される．特に後者は，ソフトウェアの運用期間が経過するにつれてソフトウェアシステムの内部構造が変化することにより生じる障害を意味する．このような現象はソフトウェアエージング (software aging) と呼ばれ，オペレーティングシステム，ミッドウェアシステム，通信アプリケーションの動作時において頻繁に観測されている [1–3].

ソフトウェアエージングによる障害は一過性の障害であることが多い [3]．すなわち，障害が発生した後，若干異なる内容（データ，環境）でシステムをリトライすることにより，あたかも障害が発生していなかったかのような操作可能状況に復帰する可能性がある．反面，このような一過性の障害は，ソフトウェアシステムのソースコード上で原因を特定することが極めて困難であることから，その対処法について数多くの研究がなされてきた．特に，ソフトウェア若化 (software rejuvenation) と呼ばれる方策は，ソフトウェアエージングによる一過性の障害を予防するための有効な方法として認識されており，ソフトウェアシステムの稼働を一時的に停止し，その内部構造を浄化した後にシステムを再稼働する一連の予防保全手続きを意味する [4–10]．ここで，ソフトウェアシステムの内部構造の浄化とは，ガーベジコレクションやオペレーティングシステムにおけるカーネルテーブルの洗浄，データ構造の初期化などを示す．アプリケーションシステムの運用上，極端ではあるが最も頻繁に行われているソフトウェア若化の一例として，ハードウェアリブートが挙げられる．

ここで問題となるのは，どのようなタイミングでシステムを予防的に若化するかにある．Huang ら [4] はソフトウェアシステムの時間的挙動を，正常稼働状態，障害発生可能な状態，障害の発生状態，ソフトウェア若化状態の 4 状態をもつ連続時間マルコフ連鎖で記述し，ランダムな若化スケジュールのもとでシステムのアンアベイラビリティや定常状態における運用費用を評価している．Dohi ら [6,7] は Huang ら [4] のモデルをセミマルコフモデルに拡張し，更に障害発生時間データから直接最適な若化スケジュールを推定する統計的手法を開発している．また，Dohi ら [8] や土肥ら [9] では，総期待割引費用とコスト有効性と呼ばれる評価規範を導入し，文献 [6,7] とは異なる視点からソフトウェア若化スケジュールを決定している．ここで，文献 [6–9] において，具体的には，観測された障害発生時間データから，経験分布関数に基づいた標準総試験時間統計量を定義することによりソフトウェア若化スケジュールをノンパラメトリックに推定するアルゴリズムを提案してい

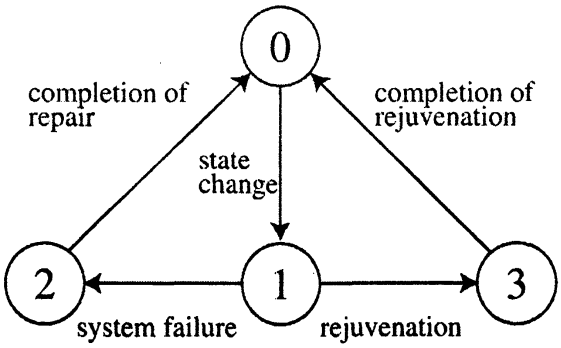


図 1: モデル 1 の状態推移図

る．この推定アルゴリズムにより，20 個程度のデータが得られれば，推定値は真の最適解に収束することが示された．しかしながら，少数の障害発生時間データしか得られない状況下において定常アベイラビリティを最大化する最適ソフトウェア若化スケジュールをより正確に推定することは重要であると考えられる．

本論文では文献 [6,7] と同様に，定常アベイラビリティを最大にする最適ソフトウェア若化スケジュールを推定することを考え，少数の障害発生時間データしか得ることができない状況下において更に推定精度を向上させることを目的とする．具体的には，得られたデータから核型密度推定量により直接最適ソフトウェア若化スケジュールを推定するためのノンパラメトリック推定アルゴリズムを提案する．また，シミュレーション実験において，提案アルゴリズムの漸近収束性を調べ，文献 [6,7] のアルゴリズムと比較することにより，少数サンプルでの推定精度が向上されることを示す．

2. セミマルコフモデル

2.1. モデル 1

ここでは，以下の四つの状態をもつ 2 種類のセミマルコフモデルについて考える．

状態 0： 正常稼働状態

状態 1： 劣化状態

状態 2： 障害発生状態

状態 3： 予防保全状態

ここで，状態 1 はメモリ漏れ量があるしきい値を超えたり，システムが正常稼働状態と比べて不安定な状態に陥ることを意味している．モデル 1 において，ソフトウェアシステムは確率的に正常稼働状態から劣化状態へと移行し，状態 0 から状態 1 へと推移する時間 Z の確率分布関数を $\Pr\{Z \leq t\} = F_0(t)$，平均を μ_0 (> 0) とする．システムが障害発生可能な状態に推移すると，ソフトウェア若化を行うか否かの決定をするものとする．システムが障害発生可能な状態から具体的に障害に至るまでの時間は，非負で連続な確率変数 X によって記述され，その確率分布関数を $\Pr\{X \leq t\} = F_f(t)$，平均を λ_f (> 0) とする．一旦障害が発生すると，事後保全が直ちに開始される．ここで事後保全とは，障害が発生した後に事後的にシステムを若化することを意味し，本論文では修理という言葉によって代用する．修理に要する時間 Y もまた非負の連続形確率変数であり，その分布関数を $\Pr\{Y \leq t\} = F_a(t)$，平均を μ_a (> 0) とする．修理が完了すると，システムの障害発生率は正常稼働状態における初期状態まで復旧される．

一方，ソフトウェアの予防的な若化は，システムが障害発生可能な状態に推移した後の T 時間経過後になされるものとする．ここで，T は非負で連続な確率変数であり，その確率分布関数を $F_r(t)$，平均を t_0 (> 0) とする．システム障害が発生する前に時刻 T が経過した場合は，直ちに予防的に若化を開始し，そのシステムオーバーヘッド V に対する確率分布関数を $\Pr\{V \leq t\} = F_c(t)$，平均を μ_c (> 0) とする．修理の場合と同様に，予防的若化が完了すると，システムの障害発生率は正常稼働状態における初期状態まで復旧される．このモデルは状態 0 から再度状態 0 に戻るまでの時間間隔を周期にもつセミマルコフ過程であり，すべての状態が

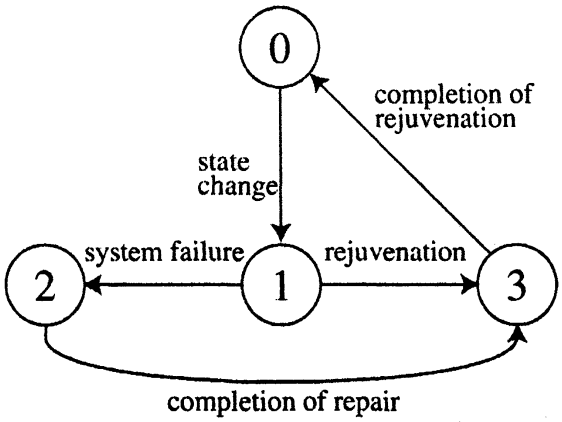


図 2: モデル 2 の状態推移図

再生点 (regeneration points) となることから，推移確率を求めるために通常のマルコフ解析の手法が適用可能である [11]．システムの状態推移図を図 1 に示す．特に障害発生可能な状態から予防的に若化を実施するまでの時間 T が一定である (periodic rejuvenation) とすれば，確率分布関数 $F_r(t)$ を次のようなユニット関数

$$F_r(t) = U(t-t_0) = \begin{cases} 1 & \text{if } t \geq t_0 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

に置き換えればよい．

Dohi ら [6,7] の結果を直接用いることによって，モデル 1 における定常アベイラビリティは

$$A_1(t_0) = \frac{\mu_0 + \int_0^{t_0} \overline{F}_f(t)dt}{\mu_0 + \mu_a F_f(t_0) + \mu_c \overline{F}_f(t_0) + \int_0^{t_0} \overline{F}_f(t)dt} \tag{2}$$

のように求めることができる．ここで，$\overline{F}_f(\cdot) = 1 - F_f(\cdot)$ である．

2.2. モデル 2

モデル 1 では，障害が発生した後に修理を開始し，修理完了後にはシステムは元の状態に復帰するものと考えていた．しかしながら，修理の内容が予防的に実施される若化と異なる場合，すなわち，ファイル系システムにおいて修理が障害により失われたデータを復旧することを含むような状況を考えるのであれば，データを回復した後に再度システム全体の若化を実施する必要がある．このような場合においては，モデル 1 は若干修正される必要が生じ，具体的には図 2 に示すような状態遷移図をもつセミマルコフ過程に帰着される．よって，モデル 2 では時間 $\min\{Z+t_0,\ Z+X+Y\}$ 経過後にソフトウェア若化を実施することになる．このとき，定常アベイラビリティは次式で与えられる．

$$A_2(t_0) = \frac{\mu_0 + \int_0^{t_0} \overline{F}_f(t)dt}{\mu_0 + \mu_c + \mu_a F_f(t_0) + \int_0^{t_0} \overline{F}_f(t)dt} \tag{3}$$

3. 標準総試験時間変換によるソフトウェア最適若化スケジュールの導出

ここでは，前述した二つのモデルに対して，各々の定常アベイラビリティを最大にする最適ソフトウェア若化スケジュールをグラフ上で導出する．今，障害発生時間分布 $F_f(t)$ に対する標準総試験時間変換 (scaled total time on test transform) を次のように定義する [12]．

$$\phi(p) = \frac{1}{\lambda_f} \int_0^{F_f^{-1}(p)} \overline{F}_f(t)dt \tag{4}$$

ここで，$F_f(t)$ は絶対連続かつ非減少関数であるので，その逆関数

$$F_f^{-1}(p) = \inf\{t_0; F_f(t_0) \geq p\}, \quad 0 \leq p \leq 1 \tag{5}$$

が必ず存在する．よく知られた結果として，確率分布関数 $F_f(t)$ が IFR (DFR) であるための必要かつ十分条件は，関数 $\phi(p)$ が $p \in [0,1]$ に関して凹（凸）関数であることである．式 (2) と式 (3) で与えられる定常アベイラビリティを $F_f(t)$ の標準総試験時間変換を用いて書き換えることにより，以下の結果が得られる [6,7].

定理 1 モデル i $(i = 1,2)$ において，定常アベイラビリティ $A_i(t_0)$ を最大にする最適ソフトウェア若化スケジュールを求める問題は，以下の問題の解 p^* $(0 \leq p^* \leq 1)$ を求めることと等価である．

$$\max_{0 \leq p \leq 1} \frac{\phi(p) + \alpha}{p + \eta_i} \tag{6}$$

ただし，

$$\alpha = \mu_0/\lambda_f \tag{7}$$

$$\eta_1 = \mu_c/(\mu_a - \mu_c) \tag{8}$$

$$\eta_2 = \mu_c/\mu_a \tag{9}$$

定理 1 から，障害発生時間分布 $F_f(t)$ が既知であるならば，最適ソフトウェア若化スケジュールは $t_0^* = F_f^{-1}(p^*)$ によって求めることができる．ここで，$p^*(0 \leq p^* \leq 1)$ は 2 次元平面上で点 $(-\eta_i, -\alpha) \in (-\infty, 0) \times (-\infty, 0)$ から曲線 $(p, \phi(p)) \in [0,1] \times [0,1]$ に引いた線分のうち，最も大きい傾斜を示す曲線上の点に対する x 座標値 p^* によって与えられる．

4. 経験分布に基づいたノンパラメトリック推定アルゴリズム

前章で得られた最適ソフトウェア若化スケジュールに関する結果は，パラメータ μ_0, μ_c, μ_a 及び障害発生時間分布 $F_f(t)$ が既知である状況下では有効である．しかしながら，実際のソフトウェアシステムの運用環境において，上記のパラメータに関する統計情報が正確に把握されていることはまれである．特に，障害発生時間分布 $F_f(t)$ を特定するためには，かなり多くの障害発生時間に対するデータが必要とされる．また，仮に過去におけるシステムの運用実績からデータが取得されていたとしても，$F_f(t)$ に適当な理論分布を仮定し，複数のパラメータ推定法から最良のものを選択し，更に理論分布と実データの適合度を検証するという一連の手続きは非常に煩雑であるばかりか，データ解析に関する経験と要領を必要とすることが知られている．ここでは，実際に障害発生時間データが与えられているという状況において，Dohi ら [6,7] により提案されたアベイラビリティを最大にする最適ソフトウェア若化スケジュールをノンパラメトリックに推定するためのアルゴリズムについて概観する．

今，障害発生時間分布 $F_f(t)$ は未知であるが，対応する障害発生時間データの順序統計量 $x_{(0)} = 0, x_{(1)}, x_{(2)}, \cdots, x_{(n)}$ が観測されており，これらは打切のない完全データであると仮定する．障害発生時間分布 $F_f(t)$ の推定量を $x_{(j)}$ $(j = 0,1,2,\cdots,n)$ に基づいた経験分布関数

$$F_n(x) = \begin{cases} j/n & \text{for} \quad x_{(j)} \leq x \leq x_{(j+1)} \\ 1 & \text{for} \quad x_{(n)} \leq x \end{cases} \tag{10}$$

によって定義する．更に，経験分布に基づいた標準総試験時間変換の推定量として，次のような標準総試験時間統計量 (scaled total time on test statistics) を定義する．

$$\phi_{nj} = \psi_j/\psi_n \tag{11}$$

ここで，関数

$$\psi_j = \sum_{k=1}^{j}(n-k+1)\left(x_{(k)} - x_{(k-1)}\right) \qquad j = 1,2,\cdots,n; \quad \psi_0 = 0 \tag{12}$$

は総試験時間統計量と呼ばれる [12]．最終的に，点列 $(j/n, \phi_{nj})$ $(j = 0,1,2,\cdots,n)$ を 2 次元平面上にプロットし，それらを線分でつなぐことにより，標準総試験時間プロット (scaled total time on test plot) を得る．推定量 $(j/n, \phi_{nj})$ $(j = 0,1,2\cdots,n)$ は $(p, \phi(p))$ $(p \in [0,1])$ のノンパラメトリック推定量であるので，定理 1 の結果を直接適用することにより，最適ソフトウェア若化スケジュールに関する以下の定理を得る [6,7].

表 1: 核型関数の例 [16]

Kernel	$K(t)$
Rectangular	$\frac{1}{2}$ for $\|t\| < 1$, 0 otherwise
Gaussian	$\frac{1}{\sqrt{2\pi}} e^{-(1/2)t^2}$
Triangular	$1 - \|t\|$ for $\|t\| < 1$, 0 otherwise
Biweight	$\frac{15}{16}\left(1-t^2\right)^2$ for $\|t\| < 1$, 0 otherwise
Epanechnikov	$\frac{3}{4}\left(1-\frac{1}{5}t^2\right)/\sqrt{5}$ for $\|t\| < \sqrt{5}$, 0 otherwise

定理 2 モデル i $(i = 1, 2)$ において，障害発生時間に対する n (> 0) 個の完全データの順序統計量 $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$ が観測されているものとする．定常アベイラビリティを最大にする最適ソフトウェア若化スケジュールのノンパラメトリック推定量 t_0^* は，次式を満たす x_{j^*} によって与えられる．

$$j^* = \left\{ j \middle| \max_{0 \leq j \leq n} \frac{\phi_{nj} + \alpha}{j/n + \eta_i} \right\} \tag{13}$$

ここで，式 (7) の λ_f はサンプル平均 $\sum_{k=1}^{n} x_{(k)}/n$ によって置き換えられる．

5. 核型密度推定に基づいたノンパラメトリック推定アルゴリズム

前章で示した経験分布に基づいた推定アルゴリズムによると，20 個程度の障害発生時間データが得られれば推定値は真の最適解に収束することが示されている．しかし，実用上の問題点として，ソフトウェアシステムの運用期間中に多くの障害発生時間データを取得するのはきわめて困難である．したがって，少数の障害発生時間データしか得ることができない状況下において最適ソフトウェア若化スケジュールを高精度に推定することは常用であると考えられる．以下で核型密度推定 (kernel density estimation) に基づいたノンパラメトリック推定アルゴリズムを提案する．

観測された障害発生時間データ $x_1, x_2, \cdots, x_n$ が確率密度関数 f_f からの標本であると仮定する．今，核型密度推定量 (kernel density estimator) を次式によって定義する [13–16].

$$\hat{f}_f(y) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{y - x_i}{h}\right) \tag{14}$$

ここで，h (> 0) はウインドウサイズと呼ばれる．関数 $K(\cdot)$ は核型関数 (kernel function) と呼ばれ，次式を満足するように決定される．

$$\int K(t)dt = 1, \quad \int tK(t)dt = 0, \quad \int t^2 K(t)dt = r^2 \neq 0 \tag{15}$$

表 1 には代表的な核型関数を示す．通常，$K(\cdot)$ には対称な密度関数が選択される．

今，障害発生時間データから最適ソフトウェア若化スケジュールを推定するために，障害発生時間分布の推定量 $\hat{F}_f(t) = \int_0^t \hat{f}_f(s)ds$ の標準総試験時間変換を次式のように定義する．

$$\phi_{\mathrm{KDE}}(p) = \frac{1}{\hat{\lambda}_{fn}} \int_0^{\hat{F}_f^{-1}(p)} \overline{\hat{F}}_f(t)dt \tag{16}$$

ただし，

$$\hat{\lambda}_{fn} = \sum_{j=1}^{n} x_j/n \tag{17}$$

である．推定量 $(p, \phi_{\mathrm{KDE}}(p))$ は $(p, \phi(p))$ のノンパラメトリック推定量であるので，定理 1 の結果を直接適用することにより，最適ソフトウェア若化スケジュールに関する以下の定理が得られる．

定理 3 モデル i $(i = 1, 2)$ において，障害発生時間に対する n (> 0) 個の完全データ $x_1, x_2, \cdots, x_n$ が観測されているものとする．定常アベイラビリティ $A_i(t_0)$ を最大にする最適ソフトウェア若化スケジュールのノンパラメトリック推定量 $\hat{t}_0^*$ を求める問題は以下の問題の解 $p^*(0 \leq p^* \leq 1)$ を求めることと等価である．

$$\max_{0 \leq p \leq 1} \frac{\phi_{\mathrm{KDE}}(p) + \hat{\alpha}_n}{p + \eta_i} \tag{18}$$

ただし，

$$\hat{\alpha}_n = \mu_0 / \hat{\lambda}_{fn} \tag{19}$$

である．

定理 3 から，障害発生時間分布 $F_f(t)$ が未知の場合においても障害発生時間データが得られるならば，最適ソフトウェア若化スケジュールは $\hat{t}_0^* = \hat{F}_f^{-1}(p^*)$ によって求めることができる．なお，定理 3 の幾何学的解釈は定理 1 と同様である．

なお，式 (14) で提案した核型密度推定アルゴリズムを適用する場合に重要となるのはウインドウサイズ h の決定である．本論文では Duin [17] や Habbema ら [18] によって提案された対数尤度最大化基準に基づいたウインドウサイズ決定法を採用する．すなわち，観測されたデータに対して次式の問題の解 h^* を最適なウインドウサイズとする．

$$\max_{h \geq 0} L(h) = \frac{1}{n} \sum_{k=1}^{n} \log \hat{f}_{nk}(x_k) \tag{20}$$

ただし，

$$\hat{f}_{nk}(x_k) = \frac{1}{(n-1)h} \sum_{\substack{j=1 \\ j \neq k}}^{n} K\left(\frac{x_k - x_j}{h}\right) \tag{21}$$

である．

6. シミュレーション実験

ここでは本論文で提案した核型密度推定に基づいたノンパラメトリック推定アルゴリズムの漸近的性質とその収束速度について考察を行う．文献 [6,7]，及び本論文で示した定理 3 の結果は，いずれも統計的に十分な数の障害発生時間データが与えられれば，最適ソフトウェア若化スケジュールの推定量は（未知ではあるが）真の最適解に収束する．しかしながら，実用上の問題点として，ソフトウェアシステムの運用期間中に十分な数の障害発生時間データを取得することは極めて困難である．よって，それぞれのアルゴリズムがどれくらいの障害発生時間データ数で機能しするかを考察する．更に，本論文で提案したアルゴリズムで得られる推定値の推定精度が少数サンプルに対して高く，高速に収束することについても示す．

以下では障害発生時間がワイブル分布

$$F_f(t) = 1 - e^{-\left(\frac{t}{\theta}\right)^{\gamma}} \tag{22}$$

に従うものと仮定する．ここで，形状パラメータ $\gamma = 4.0$，尺度パラメータ $\theta = 0.9$ とする．また，その他のパラメータとして $\mu_0 = 2.0$, $\mu_a = 0.04$, $\mu_c = 0.03$ と設定した．以上のようなパラメータ設定のもとで **3.** で示した標準総試験時間変換により，モデル 1 対して $t_0^* = 0.5870$，$A_1(t_0^*) = 0.9878$，モデル 2 に対しては $t_0^* = 0.3777$，$A_2(t_0^*) = 0.9870$ が得られた．

まず，障害発生時間分布は未知であるが，障害発生時間データがあらかじめ観測されているときに定常アベイラビリティを最大にする最適若化スケジュールを推定する．ここでは 20 個の観測データが式 (22) で与えられる疑似乱数から構成されるものとする．なお，以下のシミュレーション実験においては核型関数 $K(\cdot)$ に対して Epanechnikov 核型関数 [19]

$$K(t) = \begin{cases} \frac{3}{4}\left(1 - \frac{1}{5}t^2\right)/\sqrt{5} & \text{for } |t| < \sqrt{5}, \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

を適用する．障害発生時間データに対する 20 個の疑似乱数から，式 (20) を用いて最適なウインドウサイズは $h^* = 0.130622$ となった．図 3 (a) にはモデル 1 に対して定常アベイラビリティを最大にする最適若化スケジュールを推定した場合の結果を示す．20 個の障害発生時間データから核型密度推定によって得られた $(p, \phi_{\mathrm{KDE}}(p))$ に対して，$(-\eta_1, -\alpha) = (-3.0, -2.3797)$ から引いた線分のうち傾きを最大にする点は $(0.1477, 0.6555)$ である．したがって，最適ソフトウェア若化スケジュールは $\hat{t}_0^* = \hat{F}_f^{-1}(0.1477) = 0.56493$ と推定され，そのときの定常アベイラビリティは $A_1(\hat{t}_0^*) = 0.98781$ と推定された．同様に，モデル 2 に対しても，図 3 (b) に示すとおり $(-\eta_2, -\alpha) = (-0.75, -2.3797)$ から $(p, \phi_{\mathrm{KDE}}(p))$ に対して引いた線分のうち傾きを最大にする点は $(0.0172, 0.4512)$ となるので，最適ソフトウェア若化スケジュールは $\hat{t}_0^* = 0.37993$，定常アベイラビリティは $A_2(\hat{t}_0^*) = 0.98727$ と推定された．

次に，2 種類の推定アルゴリズムに対して，最適ソフトウェア若化スケジュールの推定値とその定常アベイラビリティの推定値に関する漸近的性質を図 4，5 に示す．なお，図中の Kernel density, Empirical distribution はそれぞれ，本論文で提案した核型密度推定による推定値，文献 [6,7] の経験分布による推定値を意味している．また，real optimal は真の最適解を示している．これらの図より，2 種類の推定アルゴリズムに対して，障害発生時間データ数が 20 付近で真の最適解に収束している様子を読み取ることができる．

更に，障害発生時間分布のパラメータに関する感度分析及び推定値の収束速度について調査する．ワイブル分布のパラメータについての 4 種類の組み合わせについて上述のシミュレーション実験をそれぞれ 100 回繰り返すことにより得られた最適ソフトウェア若化スケジュールの推定値と真の最適解との相対絶対誤差平均 (RAEA) を図 6，7 に示す．これらの図より，本論文で提案した核型密度推定アルゴリズムの収束速度は経験分布に基づいた推定アルゴリズムと比較して非常に高速であることが読み取れる．特に，得られる障害発生時間データが 15 個以下の場合においては本論文で提案した統計アルゴリズムが極めて有効であることが確認できる．

7. まとめと今後の課題

本論文では，文献 [6,7] において扱われたソフトウェアシステムの若化スケジュールを決定するためのモデルに対して，少数の障害発生時間データしか得ることができない状況下における推定精度を向上させる統計アルゴリズムを提案した．具体的には，得られたデータから核型密度推定により障害発生時間の密度関数及び総試験時間変換の推定量をノンパラメトリックに推定した．これにより少数のデータから直接最適ソフトウェア若化スケジュールを推定する枠組みを提案した．シミュレーション実験により検証した結果，障害発生時間データ数が少ない場合でも従来のアルゴリズムと比較して早い段階で真の最適解に収束するため，提案アルゴリズムは少数サンプルに対して極めて有効であることが示された．

本論文では，定常状態におけるアベイラビリティを評価規範とするソフトウェアシステムの若化スケジュールを取り扱った．今後の課題として，文献 [8,9] で考えられた総期待割引費用やコスト有効性を評価規範とするような若化スケジュールに対しても本論文で提案した推定アルゴリズムを適用し，有効性を検証する．また，障害発生時間データをオンラインで取得しながら適応的に若化スケジュールを決定する管理ソフトウェアの開発を行う予定である．

参考文献

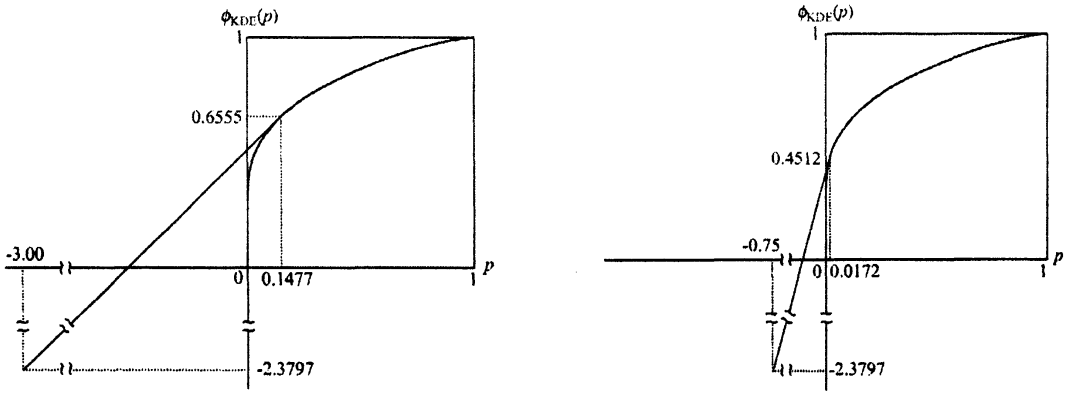
[1] E. Adams, "Optimizing preventive service of the software products," IBM J. Research and Development, vol.28, pp.2–14, 1984.

[2] V. Castelli, R.E. Harper, P. Heidelberger, S.W. Hunter, K.S. Trivedi, V. Vaidyanathan, and W.P. Zeggert, "Proactive management of software aging," IBM J. Research & Development, vol.45, pp.311–332, 2001.

[3] J. Gray and D.P. Siewiorek, "High-availability computer systems," IEEE Comput., vol.9, pp.39–48, 1991.

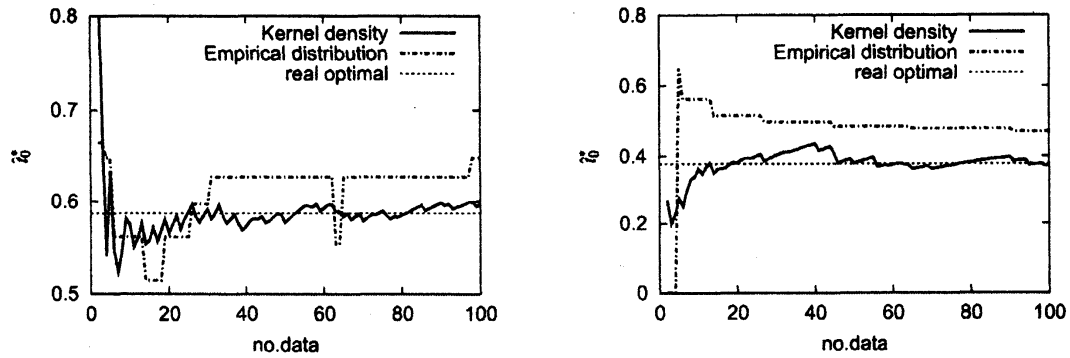
[4] Y. Huang, C. Kintala, N. Kolettis, and N.D. Funton, "Software rejuvenation: Analysis, module and applications," Proc. 25th IEEE Int'l Symp. Fault Tolerant Computing, pp.381–390, IEEE Computer Society Press, Los Alamitos, CA, 1995.

[5] K.S. Trivedi, K. Vaidyanathan, and K. Goševa-Popstojanova, " Modeling and analysis of software aging and rejuvenation," Proc. 33rd Annual Simulation Symp. pp.270–279, IEEE Computer Society Press, Los Alamitos, CA, 2000.



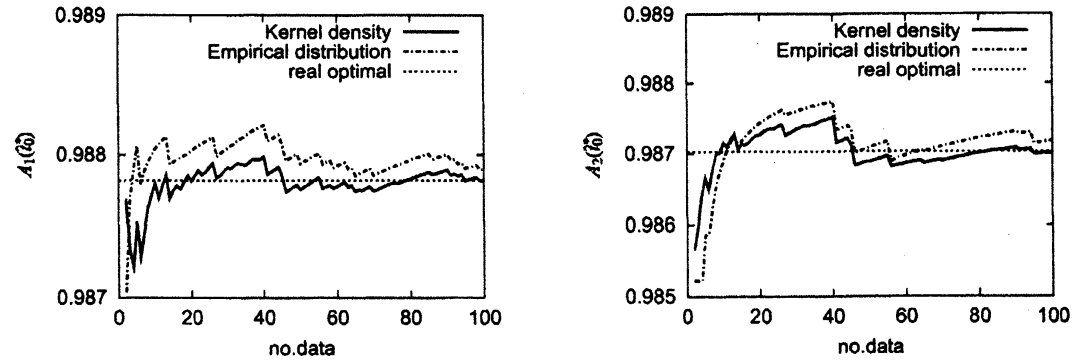
(a) モデル 1　　(b) モデル 2

図 3: 核型密度推定に基づいた最適ソフトウェア若化スケジュールの推定



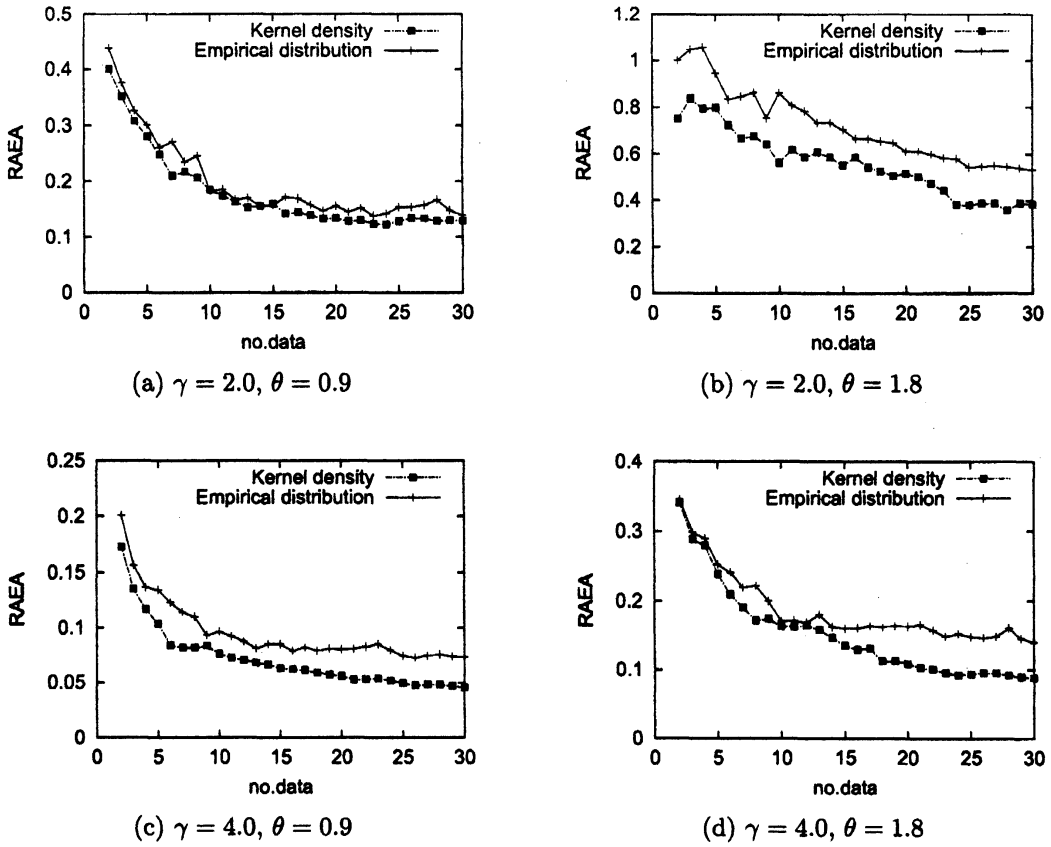
(a) モデル 1　　(b) モデル 2

図 4: 最適ソフトウェア若化スケジュールの推定値に関する漸近的性質



(a) モデル 1　　(b) モデル 2

図 5: 定常アベイラビリティの推定値に関する漸近的性質



(a) $\gamma = 2.0, \theta = 0.9$ (b) $\gamma = 2.0, \theta = 1.8$

(c) $\gamma = 4.0, \theta = 0.9$ (d) $\gamma = 4.0, \theta = 1.8$

図 6: 最適ソフトウェア若化スケジュールの推定値に関する相対絶対誤差平均（モデル 1）

[6] T. Dohi, K. Goševa-Popstojanova, and K.S. Trivedi, "Analysis of software cost models with rejuvenation," Proc. 5th IEEE Int'l Symp. High Assurance Systems Engineering, pp.25–34, IEEE Computer Society Press, Los Alamitos, CA, 2000.

[7] T. Dohi, K. Goševa-Popstojanova, and K.S. Trivedi, "Statistical non-parametric algorithms t estimate the optimal software rejuvenation schedule," Proc. 2000 Pacific Rim Int'l Symp. on Dependable Computing, pp.77-84, IEEE Computer Society Press, Los Alamitos, CA, 2000.

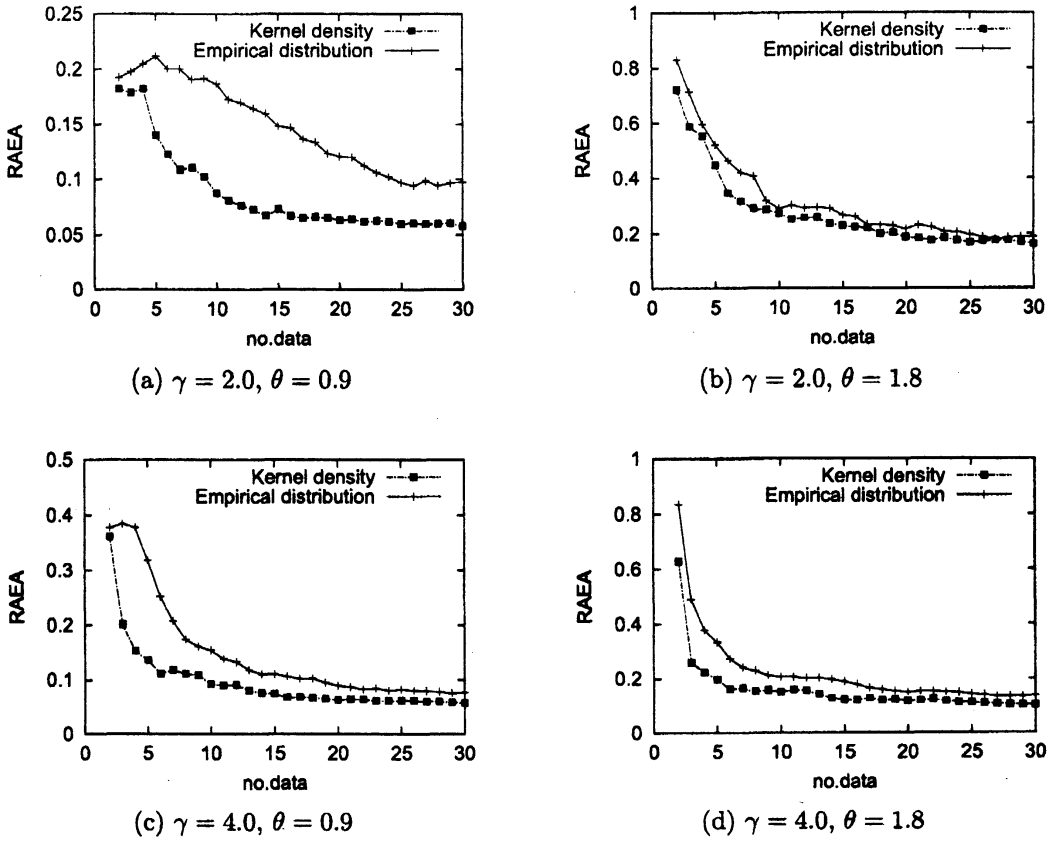
[8] T. Dohi, T. Danjou, and H. Okamura, "Optimal software rejuvenation policy with discounting," Proc. 2001 Pacific Rim Int'l Sympo. on Dependable Computing, pp.87–94, IEEE Computer Society Press, Los Alamitos, CA, 2001.

[9] 土肥 正, 海生直人, 尾崎俊治, "コスト有効性に基づいた通信ソフトウェアシステムに対する予防保全スケジュールの決定," 信学論 (A), vol.J85-A, no.2, pp.197–206, Feb. 2002.

[10] K. Rinsaka and T. Dohi, "Behavioral analysis of a fault-tolerant software system with rejuvenation," IEICE Transactions on Information and Systems, vol.E88-D, no.12, pp.2681–2690, 2005.

[11] S. Osaki, Applied Stochastic System Modeling, Springer-Verlag, Berlin, 1992.

[12] R.E. Barlow and R. Campo, "Total time on test processes and applications to failure data", Reliability and Fault Tree Analysis, ed. R.E. Barlow, J. Fussell and N.D. Singpurwalla, pp.451–481, SIAM, Philadelphia, 1975.



(a) $\gamma = 2.0, \theta = 0.9$ (b) $\gamma = 2.0, \theta = 1.8$

(c) $\gamma = 4.0, \theta = 0.9$ (d) $\gamma = 4.0, \theta = 1.8$

図 7: 最適ソフトウェア若化スケジュールの推定値に関する相対絶対誤差平均（モデル 2）

[13] E. Parzen, "On the estimation of a probability density function and the mode," Annals of Mathematical Statistics, vol.33, pp.1065–1076, 1962.

[14] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," Annals of Mathematical Statistics vol.27, pp.832–837, 1956.

[15] T. Cacoullos, "Estimation of a multivariate density," Annals of the Institute of Statistical Mathematics, vol.18, pp.178–189, 1966.

[16] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman and Hall, London, 1986.

[17] R.P.W. Duin, "On the choice of smoothing parameters for Parzen estimators of probability density functions," IEEE Trnas. Comput., vol.C-25, pp.1175–1179, Nov. 1976.

[18] J.D.F. Habbema, J. Hermans, and K. van der Broek, "A stepwise discrimination program using density estimation," Compstat, ed. G. Bruckman, pp.100–110, Physica Verlag, Vienna, 1974.

[19] V.A. Epanechnikov, "Nonparametric estimation of a multidimensional probability density," Theory of Probability and Its Applications, vol.14, pp.153–158, 1969.