

科学研究費助成事業 研究成果報告書

平成 27 年 6 月 10 日現在

機関番号：34509

研究種目：若手研究(B)

研究期間：2013～2014

課題番号：25750132

研究課題名(和文) 高品質ソフトウェア開発のための動的テスト環境を考慮した信頼性評価モデルの構築

研究課題名(英文) Software reliability models in consideration of the dynamic testing environment for high quality software development

研究代表者

林坂 弘一郎 (Rinsaka, Koichiro)

神戸学院大学・経営学部・准教授

研究者番号：20351794

交付決定額(研究期間全体)：(直接経費) 1,400,000円

研究成果の概要(和文)：ソフトウェア開発におけるソフトウェアテスト段階で、ソフトウェアの信頼性を把握するためにソフトウェア信頼性モデルが利用される。本研究課題では、動的に変化するテスト環境をメトリクスデータとして有機的に利用したモデルを考え、カーネル回帰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した。実際のソフトウェアフォールト検出データを用いて提案モデルの有効性を検証し、提案モデルが従来のモデルと比較してより高精度にデータに適合することが示された。また、モデルのパラメータ推定を行うためのソフトウェアツールを開発した。さらに、ソフトウェア最適若化スケジューリングや保証契約モデルの考察も行った。

研究成果の概要(英文)：Software reliability models, such as the Non-Homogeneous Poisson Process model, evaluate the quality characteristics of software from fault count data. In this study, we propose non-parametric estimation methods based on a kernel regression for a proportional intensity model, and utilize software metrics data as well as fault count data. Throughout numerical examples, we have shown that the proposed methods have a higher goodness-of-fit and predictive performance. We have developed a software reliability assessment tool, by using several testing metrics data as well as software fault data observed in the testing phase. Moreover, we have developed an adaptive approach to estimate the optimal preventive rejuvenation schedule which maximizes the steady-state system availability, and discussed an extended warranty service contract covering two types of failures.

研究分野：ソフトウェア信頼性工学

キーワード：ソフトウェア 信頼性 カーネル回帰 NHPP

1. 研究開始当初の背景

今日のような高度情報化社会において、鉄道システム、発電所などの社会インフラから、銀行や企業、国、地方自治体の情報システム、自動車、携帯電話、スマートフォンに至るまで多くのシステム、機器の動作を支えるためにソフトウェアが非常に重要な役割を果たしている。このようなソフトウェアの開発はその多くが人的作業に依存しているため、開発段階で人的ミスによるソフトウェアフォールト（バグ）が埋め込まれてしまうことは不可避である。万が一ソフトウェア内に埋め込まれたフォールトが原因となってシステムが停止または誤作動すれば、社会的に大きなダメージを与えることとなるため、高品質なソフトウェアを開発することが社会的に重要な課題となっている。

ソフトウェアの開発工程にはウォーターフォールモデル、プロトタイプモデル、スパイラルモデルの他に、エクストリームプログラミングに代表されるアジャイル開発手法などが存在する。今日においても最も利用されているソフトウェア開発工程はウォーターフォールモデルである。これは要求定義、外部設計、内部設計、プログラミング、テスト、運用・保守の行程を上流工程から下流工程へと滝が流れるように次々と進めていく開発モデルである。

ウォーターフォールモデルによる開発の各工程の中で最も期間・人員・コストを必要とする行程がテスト工程である。ソフトウェアテストの目的はシステムが要求される性能や機能を満たしているかどうかを検証することであり、正しく動作する事を確認するよりも、より多くのエラーを検出することにある。テストの進捗状況を把握したり、テストが終了した時点におけるソフトウェアの信頼性を把握したりするためにソフトウェア信頼性モデルが利用されている。代表的なソフトウェア信頼性モデルは非同次ポアソン過程（NHPP）モデルである。NHPPモデルは一定のテスト環境下におけるフォールト検出過程を記述するモデルである。しかしながら、現実のテスト工程では、そのテスト期間中にテストチーム人員数に増減があったり、テストの負荷が変更されるなど、必ずしもテスト環境が一定であるとは限らない。例えば、ソフトウェアの納期が近づきつつあるにもかかわらず、テスト項目の消化が不十分であれば、テストの人員を増員して集中的にテストを実施することも考えられる。しかし、このような場合に将来のフォールト検出の振る舞いを予測するためのモデルとして NHPP モデルは不十分である。

2. 研究の目的

これまでの先行研究において動的に変化するテスト環境をメトリクスデータとして有機的に利用したソフトウェア信頼性モデルを提案した。これらのモデルにおいて考えられているメトリクスデータは多次元の時間依存共

変量である。これらのモデルでは、テスト環境の動的な変化を考慮することで、従来の NHPP モデルと比較してソフトウェアフォールトデータに対する適合度が高くなることが示された。しかしながら、これらの先行研究ではパラメータ推定に関するいくつかの問題点をもっている。つまり、推定時にパラメータの非負制約違反が多発したり、大域的な最適化が難しいという問題点である。したがって、動的に変化する環境をメトリクスデータとして有機的に利用したソフトウェア信頼性モデルに対して、カーネル回帰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し、従来と比較して最尤法における推定パラメータ数を減少させるとともに、最尤法と各種回帰分析手法を組み合わせることで、最尤推定で発生する多くの問題点を解決することを目的とする。

3. 研究の方法

(1) 先行研究で提案した比例強度モデルに対して、ベースライン強度関数と比例強度関数をそれぞれ、最尤法、カーネル回帰によって推定する手法を提案する。このようにベースライン強度関数のパラメータと比例強度関数を個別に推定することで、最尤法における推定パラメータ数を減少させることが可能となる。しかし、カーネル回帰による比例強度関数のノンパラメトリック推定においては、カーネル回帰のカーネルパラメータ（バンド幅）や正則化パラメータの決定に長時間のシミュレーション実験が必要となる。具体的には公差確認法によって、実際の共変量データから得られた比例強度とカーネル法によって計算された比例強度関数の平均誤差を求め、この誤差が最小となるカーネルパラメータと正則化パラメータを決定する。

(2) また、意志決定支援ツールの開発についても行う。すなわち、提案モデルに対してソフトウェアの信頼性評価ツールを開発する。Java 言語を使用してソフトウェアを開発する。このソフトウェアは直感的なユーザーインタフェースを備え、ソフトウェアフォールト発見数データとテスト期間に投入された各種労力量データを入力できる。データ入力後に、ベースライン強度関数の分布、推定アルゴリズムなどを選択すれば、当該ソフトウェアの信頼性を評価し、結果をグラフなどで表示する。

4. 研究成果

(1) 比例強度モデルに対するカーネル回帰および意思決定支援ツールの開発 (①, ⑤)

先行研究で提案された比例強度モデルに対してカーネル回帰を実施するために、ベースライン強度関数と実際のフォールト検出数の比率の対数に対してカーネル回帰を適用した。具体的には以下のような手順で行った。

ソフトウェアのテスト時間 $t_k (= 0, 1, 2, \dots)$

に対して、メトリクスデータ $\mathbf{x} = (x_{k1}, \dots, x_{kd})^T$ ($k = 1, 2, \dots$) が観測可能であるとする。メトリクスデータを考慮した強度関数を

$$\lambda_{\mathbf{x}}(t_k; \boldsymbol{\theta}, \boldsymbol{\beta}) = \lambda_0(t_k; \boldsymbol{\theta}) g(\mathbf{x}_k; \boldsymbol{\beta})$$

と定義する。ここで $\lambda_0(t_k; \boldsymbol{\theta})$ はベースライン強度関数、 $\boldsymbol{\beta}$ は係数ベクトルである。共変量関数 $g(\mathbf{x}_k; \boldsymbol{\beta})$ について

$$g(\mathbf{x}_k; \boldsymbol{\beta}) = g(x_k)$$

のように関数形を規定しなければ、比例強度モデルの平均値関数は次式ようになる。

$$H_p(t_k; \boldsymbol{\theta}) = \sum_{i=1}^k g(x_i) \int_{t_{i-1}}^{t_i} \lambda_0(u; \boldsymbol{\theta}) du$$

すなわち、ある時間区間において検出されるソフトウェアフォールト数は同じ時間区間に投入されたメトリクス（テスト労力）に比例することを意味している。今回の研究成果では、ベースライン強度関数のパラメータ $\boldsymbol{\theta}$ を最尤法で、共変量関数 $g(\mathbf{x}_k)$ をカーネル回帰によってノンパラメトリックに推定することを考えた。

具体的には、以下のような推定アルゴリズムを提案した。

- Step 1: フォールト検出数データからベースライン強度関数のパラメータ $\boldsymbol{\theta}$ を最尤法によって推定する。
- Step 2: ベースライン強度関数と実際の検出フォールト数との比率を求める。
- Step 3: 共変量データおよび比率の自然対数から共変量関数の自然対数をカーネル回帰によりノンパラメトリックに推定する。
- Step 4: 推定された共変量関数を用いて平均値関数を求める。

実際のソフトウェアフォールト検出データを用いて、提案モデルの有効性を検証した。表 1 に NHPP モデルと提案モデルの平均誤差を示す。また、図 1 には平均値関数と強度関数の振る舞いを示す。これらより、提案モデルが従来の NHPP モデルと比較して、より高精度に実際のフォールト数データに適合していることが読み取れる。さらに、表 2 には 16 週の時点でソフトウェア信頼度を推定し、その後 5 週間における各モデルの予測値と実際のフォールト数データとの予測誤差平均を示す。表 2 より、予測評価においても提案モデルが NHPP モデルよりも高い精度で予測可能であることが示された。

表 1 Mean square errors

Model	NHPP	Kernel
Exponential	534.51	193.46
S-Shape	84.82	5.33

表 2 Prediction square errors

Model	NHPP	Kernel
Exponential	125.07	113.89
S-Shape	258.32	281.42

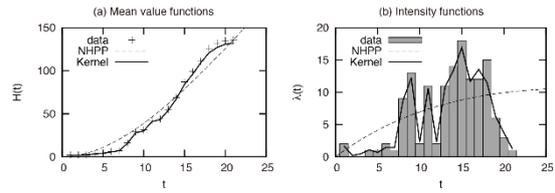


図 1 Behavior of mean value functions and intensity functions

(2) ソフトウェア最適若化スケジューリングに対するノンパラメトリック予測推論 (④)

次に、ソフトウェアの最適若化スケジュールを適応的に決定することを目的として、ノンパラメトリック予測推論に基づいた推定アルゴリズムを提案した。

ソフトウェア障害は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。ソフトウェアの経年劣化現象はソフトウェアエージングと呼ばれるが、一過性の障害であることが多い。すなわち、ハードウェアのリポートなどソフトウェア若化を実施することで、あたかも障害が発生していなかったような操作可能状態に復帰することが多い。ここでは、 n 個の障害発生時間データを得た後に、 $n+1$ 番目、および $n+2$ 番目の障害に対して最適な予防若化スケジュールをノンパラメトリック予測推論に基づき適応的に導出した。

まず、図 2 のような状態を持つマルコフ再生過程を考えた。

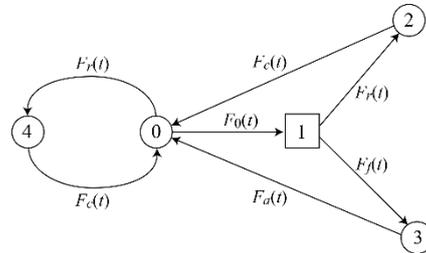


図 2 Transition diagram of Markov regenerative process

ここで、各状態は次のように定義される。

- State 0: 正常稼働状態
- State 1: 劣化状態
- State 2: 劣化状態からの予防保全状態
- State 3: 障害発生状態
- State 4: 正常稼働状態からの予防保全状態

一方で、 n 個の障害時間データ ($x_{(1)} < x_{(2)} < \dots < x_{(n)}$) が観測されたとき、 $n+1$ 番目の障害発生時間 x_{n+1} に対する予測確率はノンパラメトリック予測推論に基づけば、

$$\Pr\{X_{n+1} \in (x_{(j)}, x_{(j+1)})\} = \frac{1}{n+1}, \quad j = 0, \dots, n$$

となり、予測信頼度は次式となる。

$$S_{x_{n+1}}(x_{(j)}) = \frac{n-j+1}{n+1}, \quad j = 0, \dots, n+1$$

さらに、 $n+1$ 番目の障害に対して、障害が発

生するまでに予防的に若化を実施した場合は、障害時間データはひとつの打切時間を含むこととなる。実際に打ち切られた時間を $x_{(k)}$ と書くと、打切を含んだ X_{n+2} に対する予測信頼度関数は

$$S_{x_{n+2}}(x_{(j)}) = \frac{n+2-j}{n+2}, \quad j = 0, \dots, k$$

$$S_{x_{n+2}}(x_{(j)}) = \frac{(n+2-k)(n+1-j)}{(n+2)(n+1-k)},$$

$$j = k+1, \dots, n+1$$

となる。上式から得られる予測信頼度関数の上限および下限を、図 2 のモデルから定式化された定常アベイラビリティ関数に代入した。これらの詳細な解析の結果、次の定理が得られた。

Theorem 11

- (a) The maximum value of lower bound of predictive system availability, $LAV_{X_{n+2}}(t_0)$, for X_{n+2} can be attained by any of $x_{(j)}$, $j = 1, \dots, n$.
- (b) The maximum value of upper bound of predictive system availability, $UAV_{X_{n+2}}(t_0)$, for X_{n+2} can be attained by any of $x_{(j)}$, $j = 1, \dots, n$.

表 3 にはシミュレーション実験によって得られた誤差平均を示す。この結果、システムアベイラビリティの下限が最大になるような予防若化スケジュールを採用すればよいことが分かった。また、提案した適応的な若化スケジュールリングによってより誤差を小さくできることが示された。

表 3 Absolute error averages of pessimistic and optimistic system availabilities

	$\Delta_{low, n+1}$	$\Delta_{low, n+2}$	$\Delta_{up, n+1}$	$\Delta_{up, n+2}$
n = 10	3.880E-05	3.714E-05	5.268E-05	4.993E-05
n = 100	1.339E-05	1.330E-05	1.504E-05	1.495E-05
n = 200	9.620E-06	9.586E-06	1.041E-05	1.037E-05

(3) 延長保証契約モデル (②, ③)

最後にスマートフォンやタブレット端末などの携帯端末を念頭に、故障時にメーカーが修理再生品に無料で交換するような保証サービスを考え、この契約料金の最適な設定方法について提案した。

システムの故障は次の 2 種類に分類できると仮定する。Type-I 故障はボタンやバッテリーの劣化などの摩耗故障で、メーカーによって保証される故障である。一方で Type-II 故障は、落下によるディスプレイの損傷など、顧客の過失による外装部品の故障であり、通常はメーカー保証の適用外である。

システム導入後の保証期間を W_1 とする。保証期間中 $(0, W_1]$ にシステムが Type-I 故障すれば、メーカーまたはプロバイダ (以降、プロバイダと呼ぶ) が無料で修理を行う。プロバイ

ダは顧客に対して以下の 3 種類のオプションを提供していると仮定する。

- ・ **オプション A_1** プロバイダは契約料金 P_a で保証期間を延長して $(0, W_2]$ の間、Type-I 故障に対して保証を行い、この期間中のすべての Type-I 故障に対して無償で修理を提供する。また、保証適用範囲も拡大し、 $(0, W_2]$ 期間中の Type-II 故障に対しても無償修理を提供する。また、保証期間終了後 $(W_2, L]$ の Type-I, Type-II 故障についてはどちらも 1 回当たり C_s の料金を修理を提供する。
- ・ **オプション A_2** プロバイダはメーカー保証期間中 $(0, W_1]$ の Type-I 故障に対しては無償で修理を提供する。また、 $(W_1, L]$ の Type-I 故障および、 $(0, L]$ の Type-II 故障については、1 回当たり C_s の料金を修理を提供する。
- ・ **オプション A_0** システムを購入しない。

このようなオプションに対して、顧客の期待効用を導出し、プロバイダについては期待利益を導出した。この結果、顧客の最適戦略 $A^*(P_a, C_s)$ は

$$A^*(P_a, C_s) = \begin{cases} A_1 & \text{if } (P_a, C_s) \in \Omega_1 \\ A_2 & \text{if } (P_a, C_s) \in \Omega_2 \\ A_0 & \text{if } (P_a, C_s) \in \Omega_0 \end{cases}$$

となることが示された。なお、図 3 に顧客の最適戦略の構造を示す。

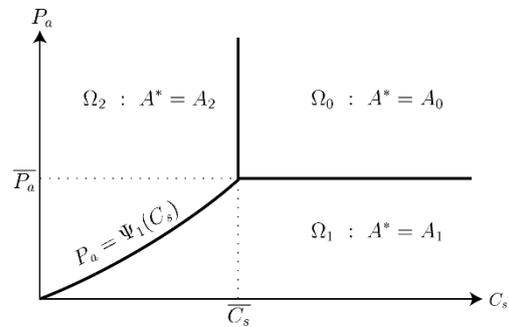


図 3 Characterization of customer's optimal actions

一方で、プロバイダの最適戦略は、自身が決定した (P_a, C_s) に対する顧客の反応を考慮した上で、自身の期待利益が最大となるような (P_a, C_s) を決定することになる。

$(P_a, C_s) \in \Omega_1$ のとき、顧客の最適な反応はオプション A_1 を選択することとなる。このとき、プロバイダの期待利益は P_a に関して単調増加となる。したがって、プロバイダの期待利益を最大にするのは、 $P_a^* \rightarrow \bar{P}_a - 0$ かつ $C_s^* > \bar{C}_s$ のときである。

$(P_a, C_s) \in \Omega_2$ のとき、顧客の最適な反応はオプション A_2 を選択することとなる。このとき、プロバイダの期待利益は C_s に関して単調増加となる。したがって、プロバイダの期待利益

を最大にするのは、 $P_a^* > \bar{P}_a$ かつ $C_s^* \rightarrow \bar{C}_s - 0$ のときである。

$(P_a, C_s) \in \Omega_0$ のとき、顧客の最適な反応はオプション A_0 を選択することとなる。このとき、プロバイダは自身の期待利益を制御することができない。

したがって、顧客がオプション A_1 または A_2 を選択した場合のプロバイダの期待利益が少なくとも一方について正となる場合、自身の期待利益が大きくなるオプションを顧客に選択させることがプロバイダの最適戦略となる。

さらに、スマートフォン等を念頭に置いた数値例を通して、危険中立的な顧客にとってはオプション A_1 と A_2 が無差別になることを示した。また危険回避的な顧客に対しては有償修理料金 C_s を高く設定することで、顧客は延長保証サービス契約を選択し、この結果プロバイダは自身の期待利益を大きくできることを示した。

5. 主な発表論文等

[雑誌論文] (計 5件)

- ① K. Rinsaka, Kernel Regression for Proportional Intensity-Based Software Reliability Model, The Proceedings of the 9th Korea-Japan Workshop on Sustainable Management Systems in Service Industry, The 3rd International Workshop on Advanced Service Management, 査読あり, Jeju, Korea, pp.40-43, August 2013.
- ② K. Rinsaka, A study on extended warranty service contract covering two types of failures, The Proceedings of the 10th Korea-Japan Workshop on Sustainable Management Systems in Service Industry, Horuto Hall Oita, pp.85-89, August 2014.
- ③ 林坂弘一郎, 2種類の故障を対象とした延長保証サービス契約に関する一考察, 神戸学院大学経営学論集, Vol.11, No.1, pp.29-51, September 2014.
- ④ K. Rinsaka and T. Dohi, Toward high assurance software systems with adaptive fault management, Software Quality Journal, 査読あり, 21pages, December 2014, <http://link.springer.com/journal/11219>
- ⑤ K. Shibata, K. Rinsaka and T. Dohi, M-SRAT: Metrics-based software reliability assessment tool, International Journal of Performability Engineering, 査読あり, Vol.11, No.4, pp.369-379, July 2015.

[学会発表] (計 2件)

- ① K. Shibata, K. Rinsaka and T. Dohi, M-SRAT: Metrics-based software reliability assessment tool, The

Proceedings of International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE 2014), Dalian, Liaoning, China, July 22-25, 2014.

- ② 林坂弘一郎, “不慮の事故による故障に対して修理を提供する延長保証サービス契約モデル”, 日本 OR 学会秋季研究発表会 アブストラクト集, pp.100-101, 2014. (2014年8月28,29日, 北海道科学大学)

6. 研究組織

(1) 研究代表者

林坂 弘一郎 (RINSAKA, Koichiro)

神戸学院大学・経営学部・准教授

研究者番号：20351794