

ソフトウェアメトリクスに基づいた信頼性評価モデル

木南智規[†], 林坂弘一郎 (01800035)[‡], 藤原隆次 (01507374)^{††}, 土肥正 (01307065)[‡][†] 広島大学工学部第二類 (電気系)[‡] 広島大学大学院工学研究科情報工学専攻^{††} 富士通周辺機株式会社

1. はじめに

一般に, ソフトウェア信頼性評価モデルは Black Box (BB) モデルと White Box (WB) モデルに分類される. 前者がテスト工程におけるフォールト検出件数に関連したデータから当該ソフトウェア製品に対する障害発生メカニズムを予測するのに対し, 後者はテスト工程を特徴付けるメトリクスデータ (種類, 規模, 言語等) に基づいて信頼性評価を行う. BB モデルの代表例が非同次ポアソン過程 (NHPP) 等の確率過程に基づいたモデルであり, WB モデルの代表例がソフトウェアサイエンス理論である [1].

本稿では, ソフトウェアメトリクスデータを共変量として捉えることにより, BB モデルと WB モデルの融合を図る. すなわち, 西尾ら [2] による比例ハザードに基づいたモデルを動的モデルに拡張した比例強度モデル (例えば Lawless [3]) を適用し, フォールト検出件数データとテスト労力データを有機的に利用した新しい信頼性評価モデルを提案する. 具体的には, Yamada ら [4] によるテスト労力依存型 NHPP モデルとの適合性比較を行い, 共変量データの信頼性評価に与える影響について定量的に評価する.

2. テスト労力依存型 NHPP モデル

ソフトウェア開発のテスト工程では大量のテスト資源が投入されており, 工数, テスト時間 (CPU 時間), 投入テストケース数といったテスト労力がソフトウェアフォールトの検出頻度に大きく依存していると考えられている. Yamada ら [4] は, テスト時刻 $t (\geq 0)$ におけるテスト労力投入量を決定論的な関数 (テスト労力関数) $w(t)$ によって表現し, 次のような平均値関数をもつ NHPP モデルを提案している.

$$H(t) = a(1 - \exp[-r \cdot W(t)]). \quad (1)$$

ここで, $a (> 0)$ は初期内蔵フォールト数の平均であり, $r (> 0)$ はテスト時刻 t において投入される単位テスト労力当たりのフォールト発見率,

$$W(t) = \int_0^t w(x) dx \quad (2)$$

はテスト時間区間 $(0, t]$ で投入される累積テスト労力関数である.

テスト労力関数 $w(t)$ には指数関数やレイリー関数を仮定することが多い. 文献 [1, 4] ではテスト労力としてテスト実行時間だけを適用し, $w(t)$ に含まれるパラメータを最小 2 乗

法により推定した後に, a と r を実際に検出されたフォールト発見個数データから最尤法を用いて推定している.

上述のモデルにおいて, 2 種類以上のテスト労力データを利用するためには若干の工夫が必要となる. すなわち, $l (\geq 1)$ 種類のテスト労力データに対して累積テスト労力関数 $W_i(t)$ ($i = 1, \dots, l$) と未知定数 r_i を定義し, 以下のように平均値関数を定義する.

$$H(t) = a \left(1 - \exp \left[- \sum_{i=1}^l r_i \cdot W_i(t) \right] \right). \quad (3)$$

例えばテスト労力関数にレイリー関数を仮定すると,

$$w_i(t) = 2\alpha_i \delta_i t \cdot \exp[-\delta_i t^2], \quad (4)$$

$$W_i(t) = \alpha_i (1 - \exp[-\delta_i t^2]) \quad (5)$$

となる. ただし, α_i はテストに必要な $i (= 1, \dots, l)$ 番目のテスト労力の総量, δ_i は労力 i の尺度パラメータである.

しかしながら, テスト労力依存型 NHPP モデルの欠点として, 最小二乗法と最尤法を併用することにより, モデルパラメータの最適性に関する統計的な意味が不明瞭であることが挙げられる. 換言すると, 上述のようなモデルとパラメータ推定法を適用するならば, せっかくテスト工程においてメトリクスデータを記録したとしてもこれらの情報を信頼性評価モデルに矛盾なく反映することは困難であり, メトリクスデータを採集すること自体が意味をなさない可能性がある. そこで次節では, 比例強度モデル [3] に基づいたソフトウェア信頼性モデルを提案する.

3. 比例強度モデル

テスト時刻 $t = t_k (= 0, 1, 2, \dots)$ において, ソフトウェアメトリクスデータ $\mathbf{x} = (x_{k1}, \dots, x_{kl})$ が観測可能であるとし, メトリクスデータを考慮した場合の強度関数を

$$\lambda_x(t | \mathbf{x}) = \lambda(t)g(\mathbf{x}; \beta) \quad (6)$$

のように仮定する. ただし, $\lambda(t)$ はベースライン強度関数であり, $g(\mathbf{x}; \beta)$ は共変量関数, $\beta = (\beta_1, \dots, \beta_l)^T$ は係数ベクトルである. ここで

$$g(\mathbf{x}; \beta) = \exp(\mathbf{x}\beta) \quad (7)$$

と仮定する. このとき平均値関数は

$$\begin{aligned} H_x(t) &= \int_0^t \lambda_x(u) du \\ &= H(t) \exp(\mathbf{x}\beta) \end{aligned} \quad (8)$$

表 1: 適合性評価 (PIM)

Exp	LLF	MSE	AIC	BIC
PIM-3	-25.006	2.093	56.012	57.929
PIM-2ei	-25.006	2.093	56.012	57.929
PIM-2ec	-26.712	2.181	61.424	63.981
PIM-2ic	-25.006	2.093	56.012	57.929
PIM-1e	-26.969	2.363	59.938	61.856
PIM-1i	-25.006	2.093	56.012	57.929
PIM-1c	-26.719	2.147	59.439	61.356
PIM-0	-29.378	3.217	62.756	64.034

と表され、 $H_x(t)$ は時間区間 $(0, t]$ において発見される期待累積フォールト数を表す。モデルパラメータの推定には最尤法を用いる。

4. 数値例

ここでは、実際の開発プロジェクトにおけるデータ [5] を用いて、提案モデルの有効性を検証する。観測された検出フォールト数は 38 個であり、 $k = 14$ 日間テストを実行した。ここでは、メトリクスデータとして、テスト実行時間 (e: CPU hr), フォールト特定労力 (i: person hr), 計算機稼働時間 (c: CPU hr) を採用する。表 1, 2 は、各モデルに対して最尤法に基づいてパラメータを推定し、対数尤度 (LLF), 実検出数との平均二乗誤差 (MSE), 情報量基準 (AIC, BIC) をそれぞれ比較したものである。ここで、PIM-3 は、3 種類のメトリクスを共変量にもつ比例強度モデル、PIM-2 および PIM-1 は 2 種類と 1 種類のメトリクスを共変量にもつ比例強度モデル (例えば、PIM-2ei は e と i の共変量をもつ), PIM-0 はメトリクスデータを全く考慮しない (通常の NHPP) モデルを指す。ここで、PIM のベースライン強度関数は指数形を仮定しているため、PIM-0 は指数形 NHPP モデルと等価である。

表 2 において、TEDM はテスト労力依存型 NHPP モデルであり、TEDM-3, TEDM-2, TEDM-1 はそれぞれ 3 つのメトリクス, 2 つのメトリクス, 1 つのメトリクスを考慮したテスト労力依存型 NHPP モデルである。ここで、Exp はテスト労力関数に指数形を、Ray はレイリー形、Logistic はロジスティック形をそれぞれ仮定している。なお、テスト労力依存型 NHPP モデルのベースライン強度関数は指数形を仮定している。

表 1 と 2 から、比例強度モデルとテスト労力依存型 NHPP モデルの比較において、LLF, MSE, AIC, BIC 全ての適合性評価規範に対して比例強度モデルの適合性が高くなることからわかる。テスト労力依存型 NHPP モデルにおいては、テスト労力関数の推定時に大きな誤差が発生しており、これがモデル全体の精度に影響を及ぼしているものと考えられる。

表 1 から、比例強度モデルにおいて、メトリクスデータをモデルに考慮することにより LLF, MSE, AIC, BIC 全ての適合性評価規範に対して適合性が高くなることからわかる。また、3 つのメトリクスを考慮したモデルが、結局フォールト特定労力のみを考慮したモデルに帰着されることから、3 つ

表 2: 適合性評価 (TEDM)

Exp	LLF	MSE	AIC	BIC
TEDM-3	-29.378	3.217	62.756	64.034
TEDM-2ei	-29.378	3.217	62.756	64.034
TEDM-2ec	-29.378	3.217	62.756	64.034
TEDM-2ic	-29.378	3.217	62.756	64.034
TEDM-1e	-29.378	3.217	62.756	64.034
TEDM-1i	-29.378	3.217	62.756	64.034
TEDM-1c	-29.378	3.217	62.756	64.034
Ray	LLF	MSE	AIC	BIC
TEDM-3	-36.047	18.281	78.588	79.866
TEDM-2ei	-36.047	18.281	78.588	79.866
TEDM-2ec	-36.450	19.345	76.899	78.177
TEDM-2ic	-36.047	18.281	78.588	79.866
TEDM-1e	-37.294	21.639	78.588	79.866
TEDM-1i	-36.047	18.281	78.588	79.866
TEDM-1c	-36.450	19.345	76.899	78.177
Logistic	LLF	MSE	AIC	BIC
TEDM-3	-33.571	13.960	71.142	72.420
TEDM-2ei	-33.571	13.960	71.142	72.420
TEDM-2ec	-33.944	14.742	71.888	73.166
TEDM-2ic	-33.571	13.960	71.142	72.420
TEDM-1e	-43.368	38.750	90.735	92.013
TEDM-1i	-33.571	13.960	71.142	72.420
TEDM-1c	-33.944	14.742	71.888	73.166

のメトリクスデータの中でフォールト特定労力が最も大きい影響を与えている因子であることがわかる。よって、ソフトウェア信頼性モデルにおいて、モデルに有効な影響を与えるメトリクスデータを慎重に選択しなければならないことがわかる。

以上のことから、テスト工程を特徴付ける共変量データを信頼性評価モデルの統計情報として組み込むことにより、モデルの適合性が高くなることが示される。このことは、一見して当り前の結論のように見受けられるが、従来からの BB モデルの限界を打破するためには重要な知見であり、ソフトウェアメトリクスに基づいた信頼性評価モデルの有効性を示すには十分な結果である。

参考文献

- [1] 山田 茂, 「ソフトウェア信頼性モデル」, 日科技連, 1994.
- [2] 西尾 泰彦, 上肥 正, 尾崎 俊治, 比例ハザードモデルに基づいたソフトウェア製品の信頼性評価, 信学論 (A), J85-A (1), 84-94, 2002.
- [3] J. F. L. Lawless, Regression methods for Poisson process data, *JASA*, **82** (399), 808-815, 1987.
- [4] S. Yamada, H. Ohtera and H. Narihisa, Software reliability growth models with testing-effort, *IEEE Trans. Reliab.*, **R-35** (1), 19-23, 1986.
- [5] J. D. Musa, A. Iannino and K. Okumoto, *Software Reliability Measurement, Prediction, Application*, 408-434, McGraw-Hill, 1987.