

ソフトウェア信頼度成長モデルの適合性評価尺度に関する考察

1 J-9

木村光宏[†], 山田茂^{††}, 尾崎俊治[†]
[†]広島大学工学部, ^{††}鳥取大学工学部

1 はじめに

本研究では, ソフトウェア信頼度成長モデル [1] を実際のテスト工程で得られたデータに対して適用する際に重要な適合性評価尺度に関する考察を行う。従来, 適合性評価尺度としては, 偏差二乗和や K-S(Kolmogorov-Smirnov) 適合度検定統計量 [1] などが良く知られている。また, テスト工程においてソフトウェアエラー-故障発生時間データが得られた場合には, ソフトウェアの故障時間分布に着目した U-plot や Y-plot などの手法が提案されている [2]。U-plot および Y-plot はソフトウェア信頼度成長モデルのデータに対する適合性の良否のみを示すだけではなく, 適用したモデルによるソフトウェア故障時間分布の予測が楽観的であるかあるいは悲観的であるか, さらに予測の傾向がテスト工程の途中において変化していたかどうかを統計的に判断することが出来るという点で有用な尺度であると言える。しかし, 実際のテスト工程において, 個々のソフトウェア故障発生時刻をデータとして採取することは, 開発ソフトウェアが大規模なものになるほど困難になるであろう。このことから本研究では, テスト工程において比較的データの採取が容易なソフトウェアエラー-発見数データに対してソフトウェア信頼度成長モデルを適用することを想定し, モデルの適合性評価尺度として, エラー数の分布に基づく U-plot および Y-plot を提案する。また, 代表的なソフトウェア信頼度成長モデルとして指数形ソフトウェア信頼度成長モデルおよび対数型ポアソン実行時間モデルを用いて適用例を示す。

2 ソフトウェア信頼度成長モデル

2.1 仮定とモデルの記述

テスト時刻 t ($t \geq 0$) までに発生した累積ソフトウェアエラー-発見数を $N(t)$ で表すと, ソフトウェアエラー-発見過程のもつ確率的性質により, 以下の仮定の下で計数過程 $\{N(t), t \geq 0\}$ を非同次ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) として取り扱うことができる。

(A1) ソフトウェアエラーの修正時に新たなエラーは作り込まれない。

(A2) $N(0) = 0$ 。

(A3) 確率過程 $\{N(t), t \geq 0\}$ は独立増分をもつ。

(A4) $\Pr\{N(t + \Delta t) - N(t) = 1\} = h(t)\Delta t + o(\Delta t)$ 。

(A5) $\Pr\{N(t + \Delta t) - N(t) \geq 2\} = o(\Delta t)$ 。

上記の仮定のうち, 仮定 (A2)-(A5) は, 計数過程 $\{N(t), t \geq 0\}$ に対する NHPP の性質を表している。

これらの仮定の下で $\{N(t), t \geq 0\}$ は次のように記述される。

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

$$H(t) = \int_0^t h(x) dx. \quad (2)$$

ここで, 式 (1) の $H(t)$ は NHPP の平均値関数 (mean value function) と呼ばれ, テスト時刻 t までに発生した累積ソフトウェアエラー-発見数の期待値を表す。また式 (2) の $h(t)$ は, NHPP の強度関数 (intensity function) と呼ばれ, 時刻 t におけるソフトウェアエラー-発見率である。この NHPP に基づくソフトウェア信頼度成長モデルは, 式 (2) の平均値関数に物理的な意味をもつ適当な関数形を指定することにより特徴づけられる。現在までに, 多くの研究者によって様々な平均値関数が提案されている。本研究では, 最も基本的なモデルとして, 指数形ソフトウェア信頼度成長モデル (exponential software reliability growth model) および対数型ポアソン実行時間モデル (logarithmic Poisson execution model) を用いる [3]。指数形ソフトウェア信頼度成長モデルの平均値関数および強度関数は, それぞれ

$$H(t) = a(1 - e^{-bt}), \quad h(t) = abe^{-bt}, \quad (3)$$

で与えられる。ここで, 未知パラメータ a は総期待潜在エラー数, b は潜在エラー 1 個当りのエラー-発見率を表す。また, 対数型ポアソン実行時間モデルの平均値関数および強度関数は, それぞれ

$$H(t) = \frac{1}{\theta} \log[\lambda\theta t + 1], \quad h(t) = \frac{\lambda}{\lambda\theta t + 1}, \quad (4)$$

で与えられる。ここで, 未知パラメータ λ は初期ソフトウェア故障強度であり, θ はエラー 1 個当りの故障強度の減少率を示す。

3 エラー-発見数の確率分布に基づく U-plot および Y-plot

本研究では, テスト工程内のあるデータ採取時点において, ソフトウェアエラー-発見数データ [1,4] が得られている

A Note on Performance Assessment Measures for Software Reliability Growth Models

Mitsuhiro Kimura[†], Shigeru Yamada^{††} and Shunji Osaki[†]

[†]Department of Industrial and Systems Engineering
Hiroshima University, Higashi-Hiroshima-shi, 724, Japan

^{††}Department of Social Systems Engineering
Tottori University, Tottori-shi, 680, Japan

場合、ソフトウェア信頼度成長モデルを適用することにより、次のデータ採取時点までに発見されるエラー数に対する分布関数を記述し、それと実際に得られたデータとの関係から U-plot および Y-plot を構成することを試みる。

いま、テスト工程においてソフトウェアエラー発見数に関する n 組のデータが得られているものとする。このとき複数のソフトウェア信頼度成長モデルの中から適合性の良好なモデルを採用し、信頼性評価を行うことを考える。まず、U-plot の描画方法について述べる。

テスト開始から i 番目までのデータ ($i < n$) に前述のソフトウェア信頼度成長モデルを適用し、モデルに含まれる未知パラメータを推定することにより、時間区間 $(t_i, t_{i+1}]$ 番目までに発見されるエラー数に関して予測される累積分布関数は

$$\hat{F}_i(y) = \sum_{m=0}^y \frac{\left\{ \int_{t_i}^{t_{i+1}} \hat{h}_i(x) dx \right\}^m}{m!} \exp\left[-\int_{t_i}^{t_{i+1}} \hat{h}_i(x) dx\right], \quad (5)$$

となる。ここで、 $\hat{h}_i(\cdot)$ はテスト開始からテスト時刻 t_i までのデータを用いて推定されたソフトウェア信頼度成長モデルの強度関数を表す。

これを $(n-1)$ 番目までのデータについて逐次推定し、エラー数に関する累積分布関数 $\hat{F}_j(y)$ ($j = i, i+1, \dots, n-1$) を求める。これら $(n-i)$ 個の累積分布関数に対して実測されたエラー数の値をそれぞれ代入したときの累積分布関数値を U_j ($j = i, i+1, \dots, n-1$) とする。すなわち、

$$U_j = \hat{F}_j(y_{j+1} - y_j) \quad (j = i, i+1, \dots, n-1), \quad (6)$$

である。さらに、 U_j を値の小さい順に並べ替えたものを新たに \bar{U}_j とおき、座標軸上に $(\bar{U}_j, \frac{j-i+1}{n-i+1})$ ($j = i, i+1, \dots, n-1$) の点を描けばよい。以上の手順において、信頼性評価開始時点 i の選び方は任意であるが、テスト工程の 50%~60% の時点とするのが妥当であるとする報告があることを付記しておく(文献[4]参照)。

適合性評価の例として 7 個の打点からなる 3 種類の U-plot を図 1 に示す。この図に示す対角線は予測したエラー数の分布の適合性(換言すれば、その導出の基本となったソフトウェア信頼度成長モデルの適合性)を調べるための基準となる。すなわち、もし観測されたデータと適用したモデルが良く一致するものであるならば、U-plot の各点は対角線の付近に描かれることになる(図 1 の○印)。これに対して、エラー数の予測の平均よりも実測値が小さい傾向にあるとき、すなわち予測が実際に比べて悲観的であるときは U-plot の各点は対角線よりも上側に描かれる(同図△印)。さらに、予測が実際に比べて楽観的であるとき、U-plot の各点は対角線よりも下側に描かれることになる(同図□印)。

次に、Y-plot の描画方法について述べる。式(6)により与えられる U_j を次式により変換し、 T_j とする。

$$T_j = -\log(1 - U_j) \quad (j = i, i+1, \dots, n-1). \quad (7)$$

各 T_j を T_j の総和で規格化した値を新たに \bar{T}_j とおき、座標軸上に $(\bar{T}_j, \frac{j-i+1}{n-i+1})$ ($j = i, i+1, \dots, n-1$) の点を描けばよい。

U-plot の場合と同様にして、7 個の打点からなる 3 種類の Y-plot を図 2 に示す。Y-plot は、発見エラー数の確率分布による予測の傾向の変化を示すことができる。すなわち、図 2 において、Y-plot の各点が対角線の周辺に描かれた場合(図 2 の○印)、予測の傾向がテスト期間において変化していないこと、およびテストデータに適用したモデルの適合度が高いことを示す。これに対して図 2 の△印が示すような傾向を持つ場合、予測を始めた初期段階では発見エラー数に関して悲観的な予測を行っていたが、ある時点で楽観的な予測を行うように変化したことを示す。また、□印はその逆の傾向変化を示すことになる。

4 適用例

紙面の都合上、第 2 節で述べた 2 つのソフトウェア信頼度成長モデルを用いた適用例は割愛し、当日発表させて頂く。

参考文献

- [1] 山田茂・大寺浩志, 「ソフトウェアの信頼性~理論と実践の応用~, ソフト・リサーチ・センター, 東京(1990).
- [2] A. A. Abdel-Ghaly, P. Y. Chan and B. Littlewood, "Evaluation of competing software reliability predictions", *IEEE Trans. Softw. Eng.*, Vol. SE-12, No. 9, pp. 950-957, September (1986).
- [3] J. D. Musa, A. Iannino and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York (1987).
- [4] 山田茂・高橋宗雄, 「ソフトウェアマネジメントモデル入門-ソフトウェア品質の可視化と評価法-」, 共立出版, 東京(1993).

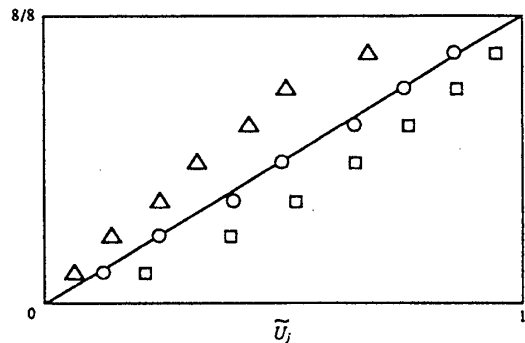


図 1 U-plot の描画例

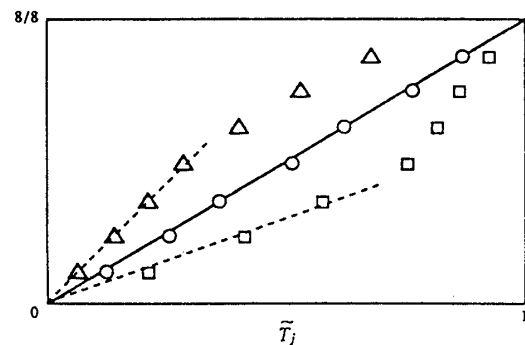


図 2 Y-plot の描画例