2001年度日本オペレーションズ・リサーチ学会
秋季研究発表会

隠れマルコフモデルによる不完全デバッグ環境でのソフトウェア信頼性評価法に関する考察

01108985　鳥取大学　*木村光宏　KIMURA Mitsuhiro
01702425　鳥取大学　山田茂　YAMADA Shigeru

1　はじめに

ソフトウェアの定量的な信頼性評価手法は開発されたソフトウェア成果物の信頼性や可用性を計測し予測するために必要であり，ソフトウェア工学における重要な問題の1つとなっている．そのようなソフトウェア信頼性評価問題に取り組むために，確率・統計論に基づいて構築された数多くのソフトウェア信頼性評価モデルがこの30年間に多くの研究者らによって開発されており．これらのモデルの中には，実用に供されているモデルもある[1]．しかしながら一般に，確率モデルはそのモデルの成立のためにいくつかの仮定を必要とするが，その仮定のいくつかは実際の現象を記述するには厳しすぎるという批判がある．

ソフトウェア信頼性評価モデルの研究分野において，完全デバッグの仮定はモデルを簡単化するのに非常に役立つことがよく知られているが，多くのソフトウェア品質／信頼性研究者らと専門家らは，ソフトウェアテスト環境に非現実的な完全デバッグ環境を仮定することに批判的である．そのような背景の下，この仮定を緩めたいくつかの不完全デバッグソフトウェア信頼性評価モデルが開発された．しかしながら，これらのモデルを実際のテスト工程などで適用する際に必要となる，不完全デバッグ率を与える未知パラメータを推定する実際的な方法は提案されていないのが現状である．

本研究では，不完全デバッグ環境を考慮した簡単なソフトウェア信頼性評価モデルをとりあげ，このモデルに含まれる未知パラメータの実用的な推定方法に焦点を当てる．具体的には，Okumoto and Goel[2]により提案された不完全デバッグモデルとテスト工程において実測された発見フォールト数データから，彼らの研究では実用的に推定することができなかった不完全デバッグ率の未知パラメータを推定する方法について考察する．

2　モデルの記述

2.1　モデルの仮定

本研究が対象とするソフトウェアのテスト及びデバッグ環境に対して次のように仮定する．

- ソフトウェアは要求仕様に基づきあらかじめ規定されたテストケースを使ってテストされる．
- ひとつのソフトウェア故障はひとつのソフトウェアフォールトによって引き起こされる．
- デバッグ過程において，ソフトウェアフォールトは正しく修正されるか，あるいは正しく修正されるとともに新しいフォールトがプログラムの別の部分(当該テストケースの実行によってテストされるプログラムパス以外の部分)に作り込まれる．

一般に，ソフトウェアのテスト工程では，テストケースの実行によりソフトウェア内にフォールトが潜在していることが判明すると，そのソフトウェアは詳しく調べられ，最終的に当該テストケースが首尾よく処理されるまでデバッグされる．その結果として，あるテストケースがソフトウェアシステムに対して正しく処理されたとき，そのテストケースが処理される際に通過するプログラムパスは少なくともそのテストケースに関してはフォールトを含んではいない．言い換えれば，ソフトウェアデバッグ作業者があるテストケースで発見されたフォールトの修正に失敗するということは，デバッグ作業が当該テストケースが正しく処理されるまで行われることを考えると，新しいフォールトがプログラムのある部分に作りこまれることになり，この作り込まれたフォールトは，以後に適用されるテストケースによって発見されるか，あるいはテスト工程が終了するまでもはや発見されないことになる．本研究ではこのようなデバッグ環境を仮定する．

2.2　従来のモデル

前節の仮定に基づき，Okumoto and Goel[2]は不完全デバッグ環境を考慮したソフトウェア信頼性評価モデルを提案した．このモデルはマルコフ過程としてソフトウェア内の潜在フォールト数の振舞いを表したものであり，その状態遷移図を図1に表す．この図におけるパラメータpは完全デバッグ率を，qは不完全デバッグ率をそれぞれ表す$(p+q=1)$．またNはテスト開始時の潜在フォールト数を表す．

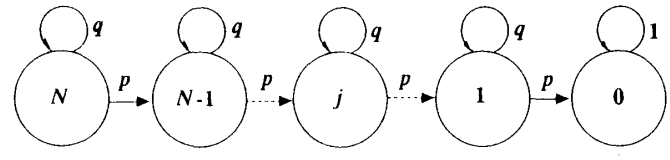

図1　状態遷移図

このモデルの状態jにおける滞在時間分布関数，すなわち，ソフトウェア内に潜在する真のフォールト数がj個であるとき，次のソフトウェア故障が発生するまでの時間分布を，

$$F_j(t) = 1-\exp[-\lambda_j t] \quad (j=0,1,\ldots,N), \qquad (1)$$

により表す．ここで，λ_jはソフトウェア故障発生に対するハザードレート(hazard rate)を表す．特に$\lambda_j=\lambda\cdot j$とすれば，この分布関数は最初にJelinski and Moranda[3]によって提案されたものを表す．

Okumoto and Goel[2]は最尤法とベイズ型推定によって統計的に推定する方法を議論したが，この方法にはデータとしてそれぞれのデバッグが完全であったか否かに関するデータが得ら

れることを前提としている．しかしながらそのようなデータを得ることは実際には非常に難しい．したがって，テスト工程におけるソフトウェアフォールト発見時間間隔の様な，観測することが可能なデータからモデルに含まれる未知パラメータを推定することが必要になる．

2.3 本研究のモデル

前節に挙げた従来のモデルの未知パラメータの推定に関する問題点を解決するため，本研究では隠れマルコフモデル（hidden-Markov model）を用いた不完全デバッグモデルを考える．隠れマルコフモデルはマルコフモデルの一種であるが，その状態空間は隠れた状態，すなわち外部から観測できない状態を含んでいるところに特徴がある．このモデル化技法の有利な点は隠れた状態に含まれる未知パラメータを推定することができることにある．このような隠れマルコフモデルは主に音声認識の研究分野で用いられてきた[4,5]．

以下に，隠れマルコフモデルとしてモデル化する際に必要となる緒量を定義する．

N：ソフトウェア開発のテスト段階の開始時におけるソフトウェアシステム内の潜在フォールト数，すなわち初期状態を表す．

p：完全デバッグ率$(0 < p \leq 1)$

q：不完全デバッグ率$(q = 1 - p)$

λ_j：状態jにおけるソフトウェア故障発生に対するハザードレート(hazard rate)

j：ソフトウェア内の潜在フォールト数を表す状態変数

$F_j(t)$：状態jにおける滞在時間分布

π_j：状態jの初期確率分布$(\sum_j \pi_j = 1)$

a_{ij}：状態iから状態jへの推移確率

3 未知パラメータの推定法

本節では，前節で定義された隠れマルコフモデルに含まれる未知パラメータλ_j，qおよびNを推定する方法について述べる．まず，分析するために得たフォールト発見時間間隔データを$(i,\ x_i)$の形式に加工する．ここで，iはi番目のフォールト発見を表し，x_iは$(i-1)$番目からi番目のフォールト発見時間間隔を表す．

このデータに対して隠れマルコフモデルにおけるBaum-Welch再推定手続き[4,5]を用いることにより，不完全デバッグ率$q(=1-p)$をはじめとする未知パラメータλ_j，およびNを推定することができる．

3.1 Baum-Welch再推定公式

以下にBaum-Welch再推定公式を示す．

$$\bar{q} = \sum_{t=1}^{T-1} \varepsilon_{00}(t) / \sum_{t=1}^{T-1} \gamma_0(t), \tag{2}$$

$$\bar{\lambda_j} = \sum_{t=1}^{T} \alpha_j[t]\beta_j[t] / \sum_{t=1}^{T} \alpha_j[t]\beta_j[t]x_t, \tag{3}$$

$$\bar{\pi_j} = \gamma_j(1). \tag{4}$$

ここで，Tは観測されたデータの個数を表す．また，

$$a_{ij} = \begin{cases} 1 & (i=0, j=0) \\ q & (i=j) \\ p & (i-j=1) \\ 0 & (\text{otherwise}) \end{cases}, \tag{5}$$

$$b_j(k) = \frac{dF_j(k)}{dk} = \lambda_j \exp[-\lambda_j k] \quad (j = N^+, N^+-1, \ldots, 1, 0), \tag{6}$$

$$\alpha_j(t) = \begin{cases} \pi_j b_j(x_1) (t=1; j = N^+, N^+-1, \ldots, 0) \\ (\sum_{i=0}^{N^+} \alpha_i(t-1)a_{ij}) b_j(x_t) (t = 2,3,\ldots,T; \\ \qquad j = N^+, N^+-1, \ldots, 1, 0) \end{cases}, \tag{7}$$

$$\beta_i(t) = \begin{cases} 1 (t = T; i = N^+, N^+-1, \ldots, 1, 0) \\ \sum_{j=0}^{N^+} a_{ij} b_j(x_{t+1}) \beta_j(t+1) \\ (t = T-1, T-2, \ldots, 1; \\ \qquad i = N^+, N^+-1, \ldots, 1, 0) \end{cases}, \tag{8}$$

$$\gamma_i(t) = \alpha_i(t)\beta_i(t) / \sum_{i=0}^{N^+} \alpha_i(T), \tag{9}$$

$$\varepsilon_{ij}(t) = \alpha_i(t) a_{ij} b_j(x_{t+1}) \beta_j(t+1) / \sum_{i=0}^{N^+} \alpha_i(T), \tag{10}$$

となる．ここに，N^+は繰返し推定のために必要となる潜在フォールト数の初期値を表す．また，計算に先立って未知パラメータq，λ_j，および$\pi = \{\pi_0, \pi_1, \ldots, \pi_{N^+}\}$に適当な初期値を与える．これらの再評価手順を繰返した後，収束した値として最終的な推定値を得ることができる．また，式(9)に与えられた$\gamma_i(t)$を用いて，最尤な状態推移の順序を推定することができ，i番目のデバッグ作業が行われた状態の推定値$\hat{s}_i$は

$$\hat{s}_i = \underset{(0 \leq j \leq N^+)}{\mathbf{argmax}} [\gamma_j(i)] \quad (1 \leq i \leq T), \tag{11}$$

により得られる．さらに，$\hat{N}$により表されるパラメータNの推定値は式(11)から$\hat{N} = \hat{s}_1$によって与えられる．

謝辞

本研究の一部は文部科学省科学研究費補助金奨励研究(A)（課題番号13780364）の下で行われた．

参考文献

[1] J. D. Musa, A. Iannino, and K. Okumoto: *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York, 1987.

[2] K. Okumoto and A. L. Goel: "Classical and Bayesian inference for the software imperfect debugging model," Technical Report No. 78-2, Department of IE & OR, Syracuse University, 1978.

[3] 山田茂: ソフトウェア信頼性モデル—基礎と応用—, 日科技連, 東京, 1994.

[4] 今井聖: 音声認識, 共立出版, 東京, 1995.

[5] L. R. Rabiner and B. H. Juang: "An introduction to hidden Markov Models," *IEEE ASSP Magazine*, Vol. 3, No. 1, pp. 4-16, 1986.