日本オペレーションズ・リサーチ学会
2004年春季研究発表会

フォールト修正を行う環境下における Single-Use 信頼度のベイズ推定

岡村寛之 (01013754)†, 古村仁志‡, 土肥正 (01307065)†
† 広島大学大学院工学研究科情報工学専攻
‡ 広島大学工学部第二類（電気系）

1. はじめに

ソフトウェアの品質は，バグが無いこと，つまり期待される動作を正確に行えることがその指標となる．作成段階で作り込まれた様々なバグや予期しない動作はテストを行うことにより可能な限り取り除いて行く方法が一般的である．

ソフトウェアの信頼性を評価する方法は一般的に動的な方法と静的な方法に大別される．動的な方法では，ソフトウェアの実行過程におけるフォールト発見数の計測から信頼性評価を行う．一方，静的な方法はソフトウェアの入出力関係から，正常に稼動する確率やソフトウェア内に潜在するフォールト数を予測することで，ソフトウェアに対する信頼性評価を行う．

文献 [1] では静的信頼性評価のうち，入力空間に基づいたソフトウェア信頼性評価として，実行したテストケースの数 n の中で障害が発生したテストケースの数 n_f から，ソフトウェアの信頼度 R およびその推定値 $\hat{R}$ を次のように定義した．

$$\hat{R} = \frac{n - n_f}{n} = 1 - \frac{n_f}{n}, \tag{1}$$

$$R = \lim_{n\to\infty} \hat{R} = 1 - \lim_{n\to\infty} \frac{n_f}{n}. \tag{2}$$

ここで定義される信頼性尺度は，実行され得る全ての入力データのうち正しい出力が得られる入力データの割合である．つまりソフトウェアが障害を発生することなく正常に終了する確率と等価であり，Single-Use 信頼度と呼ばれる．

Miller [2] は，Single-Use 信頼度に対してベイズ統計に基づいた推定手法を提案した．ソフトウェアに含まれる全てのテストケースの数を N，障害を含むテストケースの数を N_f $(< N)$ （N, N_f は未知）とすると，Miller [2] によるベイズ推定では予め得ている事前情報から $D_f = N_f/N$ の事前分布を決定する．次に，テストを行った後の事後分布をベイズの定理から導出することで，D_f に対する推定を行う．このモデルの特徴は事前分布を用いることによってテストで障害が発見されなかった場合でも，信頼性を評価することが可能な点にある．しかしながら，先に述べた定義において Single-Use 信頼度は初期時点におけるフォールトの割合に基づいた信頼性尺度である．すなわち，ソフトウェアテストにおいては当然起こりえる「障害が発見されたら修正をする」という事象を考慮していない．同様に Miller [2] による推定方法もフォールトの修正を行うことは考慮されていない．つまり，実際にはフォールトの修正が行われるのにもかかわらず，修正されたフォールトも含めて信頼度を計測しているため，実際の信頼度よりも過小評価されている．そこで本稿では，フォールトが修正される状況を考慮した信頼性評価モデルの提案を行う．

2. Miller による方法

Miller [2] による信頼度の推定は，ソフトウェア全体あるいは単一モジュールに対して行われる．この方法ではソフトウェアの障害発見確率 $(D_f = N_f/N)$ がパラメータ (p, q) のベータ分布に従うと仮定している．テストを行った後，障害発見確率の事後分布がベイズの定理から得られる．いま，障害発見確率 D_f の事前分布に対する確率密度 f_D は次式で与えられる．

$$f_D(x) = \frac{x^{p-1}(1-x)^{q-1}}{B(p,q)}. \tag{3}$$

ここで $B(p,q)$ はベータ関数である．一回のテストにおいて障害が発見される事象を $(false)$，また，障害発見確率の事後分布 $f_D(x|false)$ とする．一方，障害が発見されなかった事象を $(success)$，障害発見確率の事後分布 $f_D(x|success)$ とする．このとき，ベイズの定理からそれぞれの事後分布は

$$f_D(x|false) = \frac{x^{p}(1-x)^{q-1}}{B(p+1,q)}, \tag{4}$$

$$f_D(x|success) = \frac{x^{p-1}(1-x)^{q}}{B(p,q+1)} \tag{5}$$

となり，パラメータ $(p+1, q)$，$(p, q+1)$ のベータ分布となる．

これより，t 回のテスト終了時点において障害が発見された回数を f 障害が発見されなかった回数を s をすると，障害発見確率 D_f はパラメータ $(p+f, q+s)$ のベータ分布で表される．ここで $t = f + s$ である．最終的に Single-Use 信頼度の推定量は $R = 1 - D_f$ となるため，推定値および推定量の分散（推定精度）は以下のように与えられる．

$$E[R] = E[1 - D_f] = 1 - \frac{p+f}{p+q+t}, \tag{6}$$

$$Var[R] = Var[1 - D_f] = \frac{(p+f)(q+s)}{(p+q+t)^2(p+q+t+1)}. \tag{7}$$

3. フォールト修正を考慮した推定手法

ここではフォールト修正が行われる環境における Single-Use 信頼度に関する再考を行う．前節における推定手法は，ソフトウェアテストにおいて同じテストケースを 2 回以上行わないという環境においては，テスト開始時点における障害発生確率を推定していることになる．そこで本稿では，有限個のテストケースに対してフォールトが発見された場合に修正が行われることを考慮した推定手法の提案を行う．いま，N をテストケースの総数（既知の定数）として，N_f に対する

事前分布がパラメータ (N,p,q) のベータ二項分布に従うものと仮定する．このとき N_f の事前分布に対する確率密度 f_{N_f} は次式で与えられる．

$$f_{N_f}(n) = \binom{N}{n} \frac{B(p+n, q+N-n)}{B(p,q)}. \tag{8}$$

一回のテストにより障害が発見された場合の N_f に対する事後分布は，ベイズの定理から次のようになる．

$$f_{N_f}(n|false) = \binom{N-1}{n-1} \frac{B(p+n, q+N-n)}{B(p+1,q)}. \tag{9}$$

従って，事後分布はパラメータ $(N-1, p+1, q)$ のベータ二項分布となる．同様に，障害が発見されなかった時の N_f に対する事後分布 $f_{N_f}(n|success)$ はパラメータ $(N-1, p, q+1)$ のベータ二項分布となる．

最終的に，$t = f + s$ 回のテストにより f 回障害が見つかり，s 回障害が見つからなかった時の N_f の事後分布は，パラメータ $(N-t, p+f, q+s)$ のベータ二項分布となる．いま，Single-Use 信頼度が $R = 1 - N_f/N$ で与えられるため，推定値および推定量の分散（推定精度）は次のようになる．

$$\begin{aligned} E[R] &= E\left[1 - \frac{N_f}{N}\right] \\ &= 1 - \frac{(N-t)(p+f)}{N(p+q+t)}, \end{aligned} \tag{10}$$

$$\begin{aligned} Var[R] &= Var\left[1 - \frac{N_f}{N}\right] \\ &= \frac{(N-t)(p+f)(q+s)(p+q+N)}{N^2(p+q+t)^2(p+q+t+1)}. \end{aligned} \tag{11}$$

4. 数値例

ここでは，ソフトウェアテストをシミュレーションすることで，先に示した二つの推定方法の違いを検証する．シミュレーション環境は，ソフトウェア全体のテストケースの個数を $N = 100$，初期時点における障害を含むテストケースの個数を $N_f = 50$ とした．特に以下の二つの環境を考えそれぞれの推定手法の違いを検討する．

Case 1（復元抽出）：ソフトウェアテストにおける障害発生に対する標本を復元抽出により行う．つまり n 回のテストで n_f 個の障害が発生した場合，$n+1$ 回目のテストにおいて障害が発生する確率は $(N_f - n_f)/N$ となる．

Case 2（非復元抽出）：ソフトウェアテストにおける障害発生に対する標本を非復元抽出により行う．つまり n 回のテストで n_f 個の障害が発生した場合，$n+1$ 回目のテストにおいて障害が発生する確率は $(N_f - n_f)/(N-n)$ となる．

上記はそれぞれ，テストケースをランダムに選択してテストをする場合と，テストケースを順番に選んでテストする場合に対応する．これらの環境のもとで Miller [2] による推定手法（Miller）と本稿で提案する推定手法（Modified）による Single-Use 信頼度の推定結果を図 1 および図 2 に示す．それぞれの推定では事前情報を表すパラメータを $p = q = 1$ とし，テスト総数に関するパラメータを Case 1 においては $N = 450$，Case 2 においては $N = 100$ とした．ここでは，Case 1 に関して実際の総テストケースは 100 であるが，すべてのフォールトを取り除くためのテスト回数の期待値が 450 であることに着目している．また，実際の Single-Use 信頼度（SUR）は $R = 1 - (N_f - n_f)/N$ によって与えている．

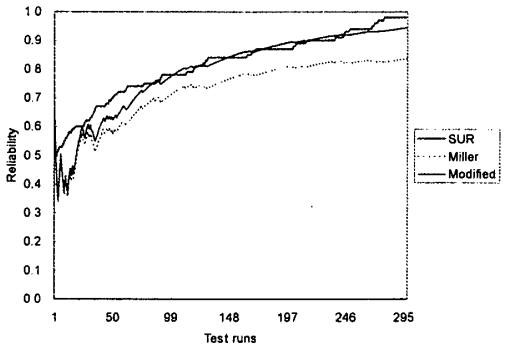

図 1: Single-Use 信頼度の推定結果（Case 1）．

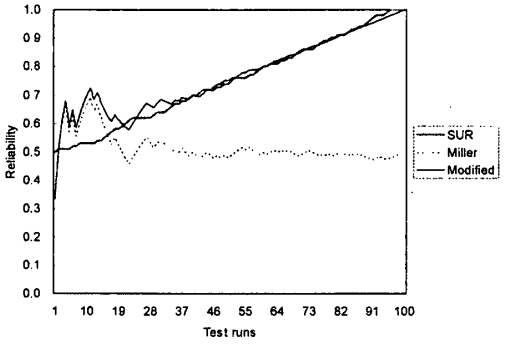

図 2: Single-Use 信頼度の推定結果（Case 2）．

これらの図から，Case 1 および Case 2 において Miller による方法は実際の Single-Use 信頼度よりも低い推定値を算出しており，実際に過小評価となっていることがわかる．特に Case 2 では，初期時点での Single-Use 信頼度である 0.5 を推定値として算出しており，フォールト修正される環境においては Miller による方法を直接適用することの危険性を示している．

参考文献

[1] T.A. Thayer, M. Lipow, and E.C. Nelson, Software Reliability, TRW Series of Software Technology, North-Holland, Amsterdam, 1978.

[2] K.W. Miller, L.J. Morell, R.E. Noonan, S.K. Park, D.M. Nicol, B.W. Murrill, and J.M. Voas, "Estimating the Probability of Failure When Testing Reveals No Failures," IEEE Transactions on Software Engineering, vol. 18, no. 1, pp. 33–43, 1992.