

ソフトウェア信頼度成長モデル —2段階エディット法を用いて—

2005MM054 岡田健司

指導教員：尾崎俊治

1 はじめに

システムの信頼性を向上させるには、ハードウェア、ソフトウェアの両面からのアプローチが必要である。ハードウェアに関しては十分研究され、信頼性の向上が見られるが、ハードウェアの信頼性が高くなるにつれて、ソフトウェアの信頼性の低さが問題となっている。

ソフトウェアの信頼性の推定のためには、プログラム中のバグ数を推定する方法がある。テスト・デバック工程を始めた時点でのバグ数は分からないので、デバック工程を継続しても、いつの時点でプログラム中のバグを全て取り除いたかを確実に判断することができない。しかし、ソフトウェアのバグ数を推定できれば、発見して除去したバグ数から残りのバグ数を推定することができるので、ソフトウェアの信頼性も推定することができる[2]。

そこで本研究は、ソフトウェアテスト中におけるバグ発見過程の時間的ふるまいに関係なく、最終的に記録されたバグ数のみで推定を行う静的な手法である2段階エディット法と、それに対して、バグが発見される過程、つまりバグが発見される時間的ふるまいを考慮した動的な手法である動的2段階エディット法について考察していく。また、本研究は文献[1]を参考にした。

2 ソフトウェア信頼度成長モデル

ソフトウェア信頼性モデルは「解析モデル」と「経験モデル」に大別している。解析モデルは、テスト工程や運用段階における信頼性要因に対する仮定の下でモデルを構築して、潜在バグ数、ソフトウェア故障率、ソフトウェア信頼度、平均ソフトウェア故障時間間隔などの信頼性尺度を導出するものである。ソフトウェアをコンピュータ上で実行してその履歴から信頼性を計測・評価する動的モデルと、ソフトウェアの実行過程は無視してその入出力関係から信頼性を計測・評価する静的モデルに分類される。

特に動的モデルは、テスト工程や運用段階におけるソフトウェアの実行過程に対し、そのバグ発見事象やソフトウェア故障発生現象をソフトウェアの信頼度成長過程として記述することから、ソフトウェア信頼度成長モデルと呼ばれる[3]。

3 2段階エディット法によるバグ数の推定

2段階エディット法とは、2人のプログラマによって実施されるソフトウェア内に潜在するバグ数を推定する方法である(図1)。この方法は、埋め込みバグによる捕獲・再捕獲法により、固有バグ数を推定する技術的な困難性を避けるために提案された[4][5]。

3.1 2段階エディット法の手順

以下に2段階エディット法の手順について述べる[4][5]。

【手順1】

第1のプログラマにテストを実施させ、テスト開始前にソフトウェア内に潜在する M 個の固有バグのうち M_1 個を発見し記録するものとする。

【手順2】

同一のプログラムを第1のプログラマとは独立に第2のプログラマにテストを実施させ、 M 個の固有バグのうち M_2 個を発見し記録するものとする。

【手順3】

各プログラマの発見したバグの記録リストを比較して、2人のプログラマにより共通して発見された同一のバグを k 個とし、テスト開始前に潜在した固有バグの総数 M を算出する。

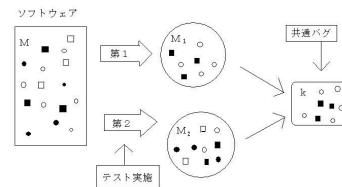


図1 2段階エディット法

3.2 固有バグの総数 M の算出

確率変数 X を2人のプログラマにより発見された同一の固有バグの個数を表すものとするれば、超幾何分布を用いて、 X の確率分布は

$$Pr\{X = k\} = \frac{\binom{M_1}{k} \binom{M - M_1}{M_2 - k}}{\binom{M}{M_2}}$$

となる。そして、最尤法を用いると

$$\frac{L(M)}{L(M-1)} = \frac{(M - M_1)(M - M_2)}{(M - M_1 - M_2 + k)M}$$

となるので

$$\frac{M_1 M_2}{k} - 1 \leq \hat{M} \leq \frac{M_1 M_2}{k}$$

このとき、 $L(M)$ は最大となるので、固有バグの総数 M は最尤推定値

$$\hat{M} = \left\lceil \frac{M_1 M_2}{k} \right\rceil$$

により算出できる。

4 動的2段階エディット法によるバグ数の推定

2段階エディット法は静的な手法であり、ソフトウェアテスト中におけるバグ発見過程の時間的ふるまいに関係なく、最終的に記録されたバグ数のみで推定が行われる。その点がソフトウェア信頼度成長モデルと異なってくる。

そこで、2段階エディット法によるソフトウェアのバグ数の推定法に対して、バグが発見される過程、バグの発見される時間的ふるまいを考慮した動的2段階エディット法について考察し、総バグ数の推定量を導出する。

4.1 数式モデル

時刻 t までに第1のプログラマと第2のプログラマがそれぞれ発見するバグ数を $M_1(t)$ と $M_2(t)$ によって表す。

また、時刻 t までに共通に発見されたバグ数を $k(t)$ とし、ソフトウェア内に潜在する全体のバグ数は M である。確率過程 $M_1(t)$ と $M_2(t)$ が互いに独立かつ独立増分な確率過程であると仮定すると、第1のプログラマが単一の共通バグを発見する確率は

$$P_1 = \frac{M_2(t) - k(t)}{M - M_1(t)}$$

第2のプログラマが単一の共通バグを発見する確率は

$$P_2 = \frac{M_1(t) - k(t)}{M - M_2(t)}$$

また、それぞれのプログラマによって、単一のバグが時刻 t までに発見される確率を $F_1(t)$ と $F_2(t)$ とすると、第1のプログラマが時刻 t において $M_1(t)$ 個のバグ数を発見する確率は、2項分布を用いて

$$Pr\{M_1(t)\} = \binom{M}{M_1(t)} F_1(t)^{M_1(t)} (1 - F_1(t))^{M - M_1(t)}$$

と定式化される。また、第2のプログラマの場合も同様である。

4.2 最尤推定法による総バグ数の推定

4.1の仮定のもとで最尤推定法を用いて総バグ数の推定を行う。そして、次のようなデータ列が与えられているものとする。

$$\begin{aligned} D_1 &= \{t_1^{(1)}, \dots, t_1^{(g_1)}\} \\ D_2 &= \{t_2^{(1)}, \dots, t_2^{(g_2)}\} \\ D_3 &= \{(M_1(t)^{(1)}, M_2(t)^{(1)}, k(t)^{(1)}), \\ &\dots, (M_1(t)^{(g_3)}, M_2(t)^{(g_3)}, k(t)^{(g_3)})\} \end{aligned}$$

ここで、 $t_1^{(i)}$ は第1のプログラマがバグを発見する時刻であり、 $t_2^{(i)}$ は第2のプログラマがバグを発見する時刻であり、 $(M_1(t)^{(i)}, M_2(t)^{(i)}, k(t)^{(i)})$ は第1のプログラマと第2のプログラマのどちらかがバグを発見したときのそれぞれのバグ数と、共通のバグ数を表している。よって、 $g_3 = g_1 + g_2$ である。

第1のプログラマと第2のプログラマは独立にテストを行っているので、尤度関数は

$$L_{1 \ 2 \ 3} = L_{3|1 \ 2} L_1 L_2 \quad (1)$$

となり、ここで L_1 と L_2 は第1のプログラマと第2のプログラマが発見するバグのデータ列 D_1 と D_2 のみに関連した尤度であり、 $f_1(t) = \frac{dF_1(t)}{dt}$ と $f_2(t) = \frac{dF_2(t)}{dt}$ とすると

$$L_1 = \prod_{i=1}^{g_1} f_1(t_1^{(i)}) (1 - F_1(T))^{M - M_1(t)} \quad (2)$$

$$L_2 = \prod_{i=1}^{g_2} f_2(t_2^{(i)}) (1 - F_2(T))^{M - M_2(t)} \quad (3)$$

となる。また、 $L_{3|1 \ 2}$ は共通バグの発見過程に関する尤度であるので、 D_1 と D_2 に関係なく

$$\begin{aligned} L_{3|1 \ 2} &= \prod_{i=1}^{g_3} \{(M_1(t)^{(i)} - M_1(t)^{(i-1)}) (k(t)^i - k(t)^{(i-1)}) P_1 \\ &+ (M_1(t)^{(i)} - M_1(t)^{(i-1)}) (1 - k(t)^i + k(t)^{(i-1)}) (1 - P_1) \\ &+ (M_2(t)^{(i)} - M_2(t)^{(i-1)}) (k(t)^i - k(t)^{(i-1)}) P_2 \\ &+ (M_2(t)^{(i)} - M_2(t)^{(i-1)}) (1 - k(t)^i + k(t)^{(i-1)}) (1 - P_2)\} \end{aligned} \quad (4)$$

となる。ここで、単一のバグ発見時間分布 $F_1(t)$ と $F_2(t)$ として指数分布を与える。

$$F_1(t) = 1 - e^{-\theta_1 t}$$

$$F_2(t) = 1 - e^{-\theta_2 t}$$

これより、式(2),(3),(4)の結果を用いて式(1)より尤度を算出し、それを最大にする M によって総バグ数を推定することができる。

5 おわりに

本研究では、ソフトウェアテスト中におけるバグ発見過程の時間的ふるまいに関係なく、最終的に記録されたバグ数のみで推定を行う従来の静的な2段階エディット法によるソフトウェアのバグ数の推定に対し、バグが発見される過程、つまりバグが発見される時間的ふるまいを考慮した動的な手法であり、ソフトウェア信頼度成長モデルに基づいた動的2段階エディット法によるソフトウェアのバグ数の推定法を提案した。

今後の課題としては、提案した動的2段階エディット法の推定量の統計的な特徴を検証し、従来の2段階エディット法と比較することである。

参考文献

- [1] 岡村寛之, 村上崇康, 土肥正: 『動的捕獲再捕獲法によるソフトウェア信頼度の推定』, 電子情報通信学会技術研究報告, vol. 103, No. 275, pp. 35-39, 2003.
- [2] 当麻喜弘: 『超幾何分布にもとづくソフトウェア残存フォールト数推定モデル』, 情報処理学会, vol. 31, No. 12, pp. 1641-1646, 1990.
- [3] 山田茂, 福島利彦: 『品質指向ソフトウェアマネジメント』. 森北出版, 東京, 2007.
- [4] 山田茂: 『ソフトウェア信頼性評価技術 ソフトウェア信頼度成長モデル入門』, HBJ 出版局, 東京, 1989.
- [5] 山田茂: 『ソフトウェア信頼性モデル - 基礎と応用』, 日科技連出版社, 東京, 1994.