2002年日本オペレーションズ・リサーチ学会
春季研究発表会

モジュールサイズの分布に基づいたソフトウェア欠陥密度の評価

本田紘介, 土肥正 (01307065), 岡村寛之 (01013754)
広島大学大学院工学研究科情報工学専攻

1. はじめに

高度情報化社会と呼ばれる現代において，ソフトウェアシステムの信頼性評価は最も重要な基盤技術として認識されている．ソフトウェアの信頼性はシステム内部に残存する欠陥 (fault) 数に大きく依存するため，高信頼化ソフトウェアを実現するためには残存欠陥数の統計的性質を正確に把握する必要がある．特に，コンポーネントの概念に基づいたソフトウェア工学 (CBSE) の有効性 [1] が叫ばれるようになった昨今において，ソフトウェアモジュールの大きさと各モジュールの欠陥密度 (fault density) の関係を解明することは重要であると考えられる．

Malaiya and Denton [2] はモジュールサイズの確率分布とソフトウェア欠陥密度を関係付けるための確率モデルを提案し，システム全体の欠陥密度を推定する方法について考察を行なっている．特に文献 [2] のような方法は，ソフトウェア統合テストの初期段階のように観測データが極めて少ない状況においては有効であるが，反面，いくつかの理論的な問題点が含まれており，ソフトウェア欠陥密度を正確に推定することが困難である．

そこで本稿では，文献 [2] で考察されたモジュールサイズの確率分布表現を修正した改良モデルを提案し，システムのモジュール構造から欠陥密度を推定するための方法について議論する．最終的に数値例において，実データを用いてモデルの比較を行い，本稿で提案したモデルの有効性を検証する．

2. ソフトウェアモジュールの欠陥密度

ソフトウェアは複数のモジュールによって構成され，各モジュールにはプログラムの実行命令が含まれている．ソフトウェアの欠陥は，モジュールに関連した欠陥とプログラム中の実行命令に関連した欠陥に分類出来るものとし，各モジュールのサイズ（行数, 命令数など）を独立で同一に分布する非負の確率変数 $S\ (>0)$ によって表す．モジュールに関連した欠陥はモジュール内に一様に分布していると考えられるため，その欠陥密度は $D_m(S)=a/S$ によって与えられるものと仮定する．ここで，$a\ (\geq 0)$ は任意のパラメータである．一方，プログラム中の実行命令に関連した欠陥はモジュールサイズが大きくなるにつれ増加すると考えられるため，その欠陥密度を $D_i(S)=b+cS$ によって表現する．ここで，$b\ (\geq 0)$ と $c\ (\geq 0)$ は任意のパラメータである．これより，サイズ S を持つモジュールの欠陥密度は，

$$D(S)=D_m(S)+D_i(S)=\frac{a}{S}+b+cS \tag{1}$$

のように記述される．

Malaiya and Denton [2] は各ソフトウェアモジュールのサイズは近似的に連続値をとり，指数分布に従うものと仮定している．すなわち，モジュールサイズの確率分布を $F_1(s)$ とすると，$F_1(s)=\Pr\{S\leq s\}=1-\exp(-gs)$ となる．ここで $1/g\ (>0)$ は平均モジュールサイズを表すパラメータである．いま，サイズ S のモジュールの欠陥数を $N(S)$ とし，$N(S)=D(S)\cdot S\cdot 10^{-3}$ のようなスケール変換を行なう．Malaiya and Denton [2] はシステム全体の総期待欠陥数を

$$\mathrm{E}[N(S)]\approx M\int_1^{s_{max}}N(s)dF_1(s) \tag{2}$$

のように求めている．ここで，$M\ (>1)$ はシステム内のモジュール総数，$s_{max}\ (>0)$ は最大モジュールサイズを表し，いずれも既知定数である．最終的に，システム全体の欠陥密度 D_1 は，平均システムサイズ $k_1=M/g$ を用いて，

$$D_1=\frac{\mathrm{E}[N(S)]}{k_1}\times 10^3 \tag{3}$$

のように表現される．

3. 改良モデル

3.1 修正指数分布モデル

Malaiya and Denton モデル [2] の問題点はモジュールサイズの確率分布を大まかにしか表現していないことである．換言すれば，モジュールサイズ分布 $F_1(s)$ の裾の影響を無視しているため，システム全体の総期待欠陥数 $\mathrm{E}[N(S)]$ は厳密な意味で期待値ではない．この問題点を克服するために，次のような修正指数分布

$$F_2(s)=\frac{\int_1^s g\exp(-gx)dx}{\int_1^{s_{max}} g\exp(-gx)dx} \tag{4}$$

を仮定する．このとき，システム全体の総期待欠陥数は，

$$\mathrm{E}[N(S)]=M\int_1^{s_{max}}N(s)\ dF_2(s) \tag{5}$$

となり，システム全体の欠陥密度は，

$$D_2=\frac{\mathrm{E}[N(S)]}{k_2}\times 10^3 \tag{6}$$

となる．ここで

$$k_2=M\int_1^{s_{max}}sdF_1(s)\Bigg/\int_1^{s_{max}}dF_1(s) \tag{7}$$

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

は平均システムサイズである．

3.2 幾何分布モデル

モジュールサイズがコード行数などの離散値として測られるならば，モジュールサイズは $S=1,2,3,\cdots,s_{max}$ のように整数値をとる離散形確率変数によって表現されるべきである．いま，各モジュールサイズ S は独立で同一に分布する幾何分布 $\Pr\{S=s\}=p_1(s)$ に従うものと仮定すれば，

$$p_1(s)=p(1-p)^{s-1},\quad s=1,2,\cdots \tag{8}$$

となる．ここで $p\in[0,1]$ は未知パラメータである．モジュールサイズが $S\in[1,s_{max}]$ であることを考慮すれば，S に関する修正確率関数は

$$p_2(s)=\frac{p(1-p)^{s-1}}{1-(1-p)^{s_{max}}},\quad s=1,2,\cdots,s_{max} \tag{9}$$

で表される．このモジュールサイズの確率分布を用いてシステム全体の総期待欠陥数を求めると

$$\mathrm{E}[N(S)]=M\sum_{s=1}^{s_{max}}N(s)p_2(s) \tag{10}$$

となり，システム全体の欠陥密度は，

$$D_3=\frac{\mathrm{E}[N(S)]}{k_3}\times 10^3 \tag{11}$$

となる．ここで，

$$k_3=\frac{M\sum_{s=1}^{s_{max}}sp(1-p)^{s-1}}{1-(1-p)^{s_{max}}} \tag{12}$$

は平均システムサイズである．

3.3 欠陥密度関数 $D(S)$ の改良

ソフトウェアシステム全体の欠陥密度を評価するためには，個々のモジュールサイズの欠陥密度関数 $D(S)$ の推定が必要である．式 (1) で与えられた欠陥密度関数は直線と双曲線の和として表現される単純な関数であり，任意パラメータ a,b,c は実際のモジュールサイズと個々のモジュールに対する欠陥密度から構成される n 個のデータ $(s_1,d_1),(s_2,d_2)\cdots(s_n,d_n)$ より最小二乗法によって推定される．式 (1) で与えられた欠陥密度関数の代わりに，n 組のデータ点を $n-1$ 次の Lagrange 多項式によって補間することにより，欠陥密度関数 $D(S)$ を推定することを考える．すなわち，

$$D(S)=\sum_{i=1}^{n}\frac{P_i(S)}{P_i(s_i)}d_i. \tag{13}$$

ここで，

$$P_i(S)\;=\;(S-x_1)\cdots(S-s_{i-1})(S-s_{i+1})\cdots(S-s_n) \tag{14}$$

である．

4. 数値例

表 1: ソフトウェア欠陥データ [3].

Module Size (Max)	No.	Defect Density
50	258	16.0
100	70	12.6
150	26	12.4
200	13	7.6
350	3	6.4

実際のソフトウェアシステムにおけるモジュールサイズ，個数，事後的に計測された平均欠陥密度に関するデータ [3] を用いて未知パラメータを推定し，式 (1) における $\hat{a}=220.90$, $\hat{b}=7.83$, $\hat{c}=0.0000$, $\hat{g}=\hat{p}=0.0054$ を得た．モジュール数 $M=370$，最大モジュールサイズ $s_{max}=350$ であるので，式 (3)，(6)，(11) から各モデルに対する欠陥密度 $D_1=5.418$, $D_2=9.616$，$D_3=9.622$ の値を得ることが出来た．また，事後的に計測された欠陥密度の算術平均は 14.731 となり，モジュールサイズの確率分布を考慮したモデルの方が若干良好な結果を示した．

一方，式 (1) の代わりに Lagrange の多項式を用いて欠陥密度を推定した場合，S が指数分布に従う場合におけるシステム全体の欠陥密度は $D_1=8.190$ となった．また，S の確率分布に修正指数分布と修正幾何分布を仮定した場合，システム全体の欠陥密度はそれぞれ $D_2=14.537$，$D_3=14.635$ のようになった．事後的に計測されたシステム全体の欠陥密度の値が 14.731 であることを考えるとモジュールサイズの確率分布と欠陥密度関数を改良することにより，実際のシステム全体の欠陥密度に近い値を算出することが可能となった．

同様にして文献 [4] におけるデータでは，モジュール数 $M=362$，最大モジュールサイズ $s_{max}=5160$ であり，式 (1) におけるパラメータの推定値は $\hat{a}=121.19$, $\hat{b}=1.76$, $\hat{c}=0.0063$, $\hat{g}=\hat{p}=0.0041$ となった．このとき，各モデルに対する欠陥密度は $D_1=5.330$，$D_2=5.331$，$D_3=5.327$ となり，事後的に計測された欠陥密度の算術平均が 4.817 であったことを考慮すれば，顕著の差は観測されなかった．一方，Lagrange の多項式を用いて欠陥密度を推定した場合，$D_1=1327.55$，$D_2=1327.56$，$D_3=1313.79$ となり，逆に推定結果は極端に悪化した．Lagrange の多項式に基づいた方法がうまく機能しなかった理由として，Lagrange 関数がデータの過適合するために極端に大きい欠陥密度の値を推定したことが挙げられる．

参考文献

[1] 青山，中所，向山 (編)，コンポーネントウェア，共立出版，1998.

[2] Y. K. Malaiya and J. Denton, Module size distribution and defect density, *Proc. Int'l Symp. on Software Reliab. Eng.*, pp. 62–71, IEEE CS Press, 2000.

[3] V. R. Basili and B. R. Perricone, Software errors and complexity, *Comm. ACM*, vol. 27, pp. 42–52, 1984.

[4] C. Withrow, Error density and size in Ada software, *IEEE Software*, vol. 7, pp. 26–30, 1990.

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．