

バグモジュール率を従属変数とした Fault-Prone モジュール判別モデルの提案

水野 勇貴† 内田 眞司† 門田 暁人‡

†奈良工業高等専門学校 ‡奈良先端科学技術大学院大学

1 はじめに

ソフトウェア開発におけるテスト工程において、欠陥(fault)を含んでいる確率の高いモジュール(以下、**fault-prone** モジュール)を特定することは、テストの効率化、および信頼性を確保する上で重要である。そのために、**fault-prone** モジュール判別モデルが提案されており、線形判別分析、ロジスティック回帰分析[1]など、様々な種類のモデルが存在する。

従来、**fault-prone** モジュール判別モデルの構築は、従属変数に、あるモジュールに対するバグの有無を用いて行われてきた。それに対し、本研究では、あるメトリクスの値域に含まれるモジュールにバグが存在する確率(以下、**バグモジュール率**)を従属変数とする、**fault-prone** モジュール判別モデルを提案する。

実験として、提案手法とロジスティック回帰分析の其々で **fault-prone** モジュール判別モデルを構築し、**fault-prone** モジュールを予測した結果、提案手法の予測精度が高いことが分かった。

2 Fault-Prone モジュール判別モデル

fault-prone モジュール判別モデル(以下、判別モデル)とは、モジュールのメトリクス値(ソースコード行数や分岐数など)を独立変数とし、従属変数であるモジュールに含まれるバグの有無を予測するためのモデルである。

判別モデルの構築は、過去に開発されたプロジェクトで計測されたデータセット(多数のモジュールのメトリクス値とバグの有無を記録したもの)を用いて行われる。構築した判別モデルに対して、新たに開発したモジュールのメトリクス値を入力することで、そのモジュールのバグが含まれる確率やバグの有無を予測する。

判別モデルの構築法には、様々な手法が存在するが、本稿では判別モデルとしてよく用いられる、ロジスティック回帰分析(以下、LRA)を用いた。

3 提案モデル

3.1 概要

提案モデルは、あるメトリクス $i(i=1,2,\dots,n)$ の値域に対するバグモジュール率 y_i を従属変数

として、判別モデルを構築する。提案モデルでは、従属変数と独立変数の関係を式(1)で定義する。

$$y = \frac{\sum_{i=1}^n w_i f(x_i)}{\sum_{i=1}^n w_i} \quad \dots (1)$$

ここで、 x_i は独立変数となるメトリクス値、 w_i は重み付け係数値、 $f(x_i)$ は y_i を従属変数とする x_i の近似式、 y は y_i の加重平均である。

3.2 提案モデル作成法

提案手法による判別モデルの基本的な作成手順を、以下に示す。

- 手順1. あるメトリクス i の値域を、値域に含まれるモジュール数が、ユーザが決めた基準値以上になるような、最低限の幅を持つ値域に分ける。(本稿では、基準値に全モジュール数の二乗根を用いた)
- 手順2. 各値域に含まれるモジュール数と、バグを含むモジュール数を計測し、各値域のバグモジュール率を求める。
- 手順3. 各値域の中央値を独立変数、バグモジュール率を従属変数として、近似式 $f(x_i)$ と重み付け係数値 w_i を決定する。(本稿では、 $f(x_i)$ に最二乗法による対数近似を、 w_i に寄与率を用いた)
- 手順4. 手順 1~3 を、全てのメトリクスに対して行う。

4 評価実験

4.1 概要

本稿では、提案手法の有効性を検討するために、提案手法と LRA の其々を用いて判別モデルを構築し、予測精度を測定した。その際、結果の妥当性を確保するために、実験を 10 回繰り返した。

4.2 データセット

実験には、NASA IV&V Facility Metrics Data Program(MDP) [2] が公開しているデータセットのうち、9 個のプロジェクトのデータセットを用い、LOC_TOTAL などの 15 種類のメトリクスを使用した。表1に、各プロジェクトの概略を示す。

表1 各プロジェクトの概略

プロジェクト名	開発言語	モジュール数	バグ含有率(%)
CM1	C	505	9.5
KC1	C++	2,107	15.4
KC3	java	458	9.4
KC4	Perl	125	48.8
MC1	C&C++	9,466	0.7
MW1	C	403	7.7
PC1	C	1,107	6.9
PC2	C	5,589	0.4
PC3	C	1,563	10.2

4.3 評価方法

判別モデルの予測精度の評価方法としては、Alberg Diagram[3] の Area Under the Curve (以下、AUC) を用いた。

Alberg Diagram とはバグを含んでいる可能性の高い順にモジュールを抽出した際に、実際にバグを含んでいたモジュールがどれだけ抽出できたかの割合をグラフ化したものである。Alberg Diagram の例を、図1に示す。

AUC とは曲線下面積のことであり、ある曲線を積分した値を指し、その値域は[0, 1] を取る。予測精度の高いモデルでは、AUC の値が大きく、ランダムに予測するモデルの場合、AUC の値は0.5程度となる。

4.4 実験手順

9 個のプロジェクトのデータセット全てに対して、以下に示す実験の手順を行った。

- 手順1. データセットをランダムに二等分し、一方をフィットデータセット fit、もう一方をテストデータセット test とする。
- 手順2. fit に対して、提案手法と LRA の其々で判別モデルを構築する。
- 手順3. 構築した各判別モデルを用いて、test の fault-prone モジュールを予測する。
- 手順4. 予測した結果を基に、Alberg Diagram を計算し、AUC を求める。
- 手順5. 手順 1~4 を 10 回繰り返す、AUC の平均値を求める。

5 実験結果

実験により得られた各プロジェクトに対する AUC の平均値を、表2に示す。得られた知見は以下の通りである

- 9 個のデータセットのうち、6 個のデータセットで、提案手法が LRA を上回っていた。
- モジュール数が多いデータセットほど、AUC の値が高い。

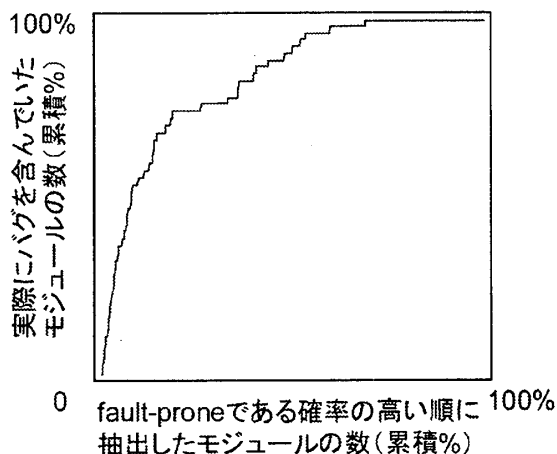


図1 Alberg Diagram の例

表2 各プロジェクトの AUC の平均値

プロジェクト名	提案手法	LRA
CM1	0.75	0.71
KC1	0.80	0.79
KC3	0.82	0.77
KC4	0.75	0.75
MC1	0.90	0.85
MW1	0.79	0.69
PC1	0.74	0.76
PC2	0.89	0.74
PC3	0.78	0.79
平均	0.80	0.76

6 まとめと今後の予定

本稿では、NASA IV&V Facility Metrics Data Program(MDP) が公開している、9 個のデータセットを用い、提案手法と LRA の其々で fault-prone モジュールの判別を行った。その結果、LRA に比べ、提案手法の予測精度が高いことが分かった。

今後は、近似式 $f(x_i)$ に対数近似以外を用いた場合の予測精度の測定や、LRA に比べ提案手法の予測精度が高い理由の考察、LRA 以外の従来法との比較・検討を行う予定である。

参考文献

- [1] A. R. Gray and S. G. MacDonell, "Software Metrics Data Analysis - Exploring the Relative Performance of Some Commonly Used Modeling Techniques", Empirical Softw. Eng., Vol.4, No.4, pp.297-316, 1999.
- [2] NASA IV&V Facility, Metrics Data Program, <http://mdp.ivv.nasa.gov/>.
- [3] N. Ohlsson and H. Alberg, "Predicting Fault-Prone Software Modules in Telephone Switches", IEEE Trans. on Software Engineering, Vol.22, No.12, Dec, 1996