異なるプロジェクト間における Fault-Prone モジュール判別の精度評価

Empirical Evaluation of Faulty Module Detection Using Datasets from Different Projects

木浦 幹雄[*] **松本 真佑**[†] **亀井 靖高**[‡] **門田 暁人**[§] **松本 健一**[¶]

あらまし 信頼性の高いソフトウェアを開発するために，ソフトウェアテストおよび保守工程で fault-prone モジュール（バグを含む確率の高いモジュール）を特定することが重要である．そのために，従来，多数のモジュールから計測されたプロダクトメトリクスを説明変数とし，モジュールに含まれるバグの有無を目的変数とする判別モデルが多数提案されている．一般に，判別モデルの構築には，旧バージョンのモジュールから計測されたメトリクス値とバグの実績データが用いられる．しかし，この方法は，旧バージョンの存在しない新規開発プロジェクトでは使えない．本稿では，異なるプロジェクトのモジュールから計測された実績データを元にモデル構築を行い，fault-prone モジュール判別を試みた結果について報告する．

1 はじめに

ソフトウェアの開発において，ソフトウェアテスト及び保守を効率よく行うために，fault-prone モジュール（バグを含みやすい傾向があるモジュール）を特定する fault-prone モジュール判別モデルが提案されている [6] [7]．提案されている判別モデルでは，多数のモジュールから計測されたプロダクトメトリクス（プログラム行数やサイクロマティック数，変更行数など）を説明変数とし，モジュールのバグの有無を目的変数としており，線形判別分析 [1]，ロジスティック回帰分析 [2]，ニューラルネット [10] など様々な種類のモデルが存在する．一般にこれらの判別モデルは，旧バージョンのモジュールから計測されたメトリクス値とバグの実績データ記したデータセット（以降，フィットデータ）を用いて構築し，現行バージョンの fault-prone モジュールの予測を行う [4]．

しかし，新規開発プロジェクトでは旧バージョンが存在しないため，判別モデルを構築することができず，fault-prone モジュールを予測することができない．情報処理推進機構ソフトウェア・エンジニアリング・センターが 19 社から収集した 1419 件のプロジェクトの実績データでは，全体の 59.6%が新規開発プロジェクトであり [3]，フィットデータを用意できないプロジェクトが過半数を占める．

一方，現行のプロジェクトとは異なる（過去の）プロジェクトで計測されたデータセットをフィットデータとして用いる場合，プロジェクトの特性（アーキテクチャや開発言語，開発規模など）の違いがメトリクス値の傾向の違いとして表れるため，精度の高いモデルを構築できるとは限らない．また，この場合の予測精度については従来十分に研究されていない．

そこで，本稿では異なるプロジェクトのモジュールから計測された実績データを元に判別モデルの構築を行い，fault-prone モジュールの予測を試みた結果について報告する．本稿では実験の妥当性を向上させるために，筆者らの予備実験 [5] を拡

[*]Mikio KIURA, 奈良先端科学技術大学院大学　情報科学研究科

[†]Shinsuke MATSUMOTO, 奈良先端科学技術大学院大学　情報科学研究科

[‡]Yasutaka KAMEI, 奈良先端科学技術大学院大学　情報科学研究科

[§]Akito MONDEN, 奈良先端科学技術大学院大学　情報科学研究科

[¶]Ken-ichi MATSUMOTO, 奈良先端科学技術大学院大学　情報科学研究科

張し，プロジェクト数を増やすとともに，性質の異なるプロジェクトを除去し実験を行った．実験では，NASA IV&V Facility Metrics Data Program [8] が公開しているデータセットの内，C 言語で開発された 9 個のデータセットを用い，自身を除くそれぞれの組み合わせ計 72 通りと，他の全プロジェクトのデータセットをフィットデータとした場合の 9 通り，各プロジェクトによる予測値の平均値を用いる場合の 9 通りについて予測を行った．

以降，2 章で fault-prone モジュール判別について説明し 3 章で実験の方法とその手順について説明する．4 章で実験結果について述べ，最後に 5 章で本稿の結論と今後の課題を述べる．

2 Fault-Prone モジュール判別

2.1 諸定義

モジュールとは，ソフトウェアの最小の品質評価単位であり，通常，ソフトウェアは複数のモジュールから構成される．fault とは，モジュール中に存在する正しくない処理のことであり，ソフトウェアが所定の機能を遂行できないことの直接の原因となったプログラムの記述誤りである．fault-prone モジュールとは，fault を含む確率が高いモジュールのことを指す．

2.2 Fault-Prone モジュール判別モデル

Fault-prone モジュール判別モデルは，過去に開発されたモジュールに関するプロダクトメトリクスを説明変数とし，モジュールに含まれるバグの有無を目的変数として構築される．構築した判別モデルに対して新規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．モデル構築方法としては，線形判別分析 [1] やニューラルネット [10] など様々な手法が存在するが，本稿では fault-prone モジュール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 [2] を用いた．

2.3 ロジスティック回帰分析

ロジスティック回帰分析は，説明変数の変動によって目的変数が 2 値のどちらを取る確率が高いかを予測する手法である．ロジスティック回帰分析では，説明変数と目的変数の関係が以下のロジスティック回帰式により定義される．

$$P(y|x) = \frac{1}{1 + e^{-(\beta^T x + \alpha)}} \tag{1}$$

ここで，y は 2 値の目的変数，x は説明変数となる入力ベクトル，β は係数ベクトル，α は係数値であり，P は入力ベクトル x に対して y が 1 となる確率である．

3 適用実験

3.1 概要

本実験では，予測対象プロジェクトとは異なったプロジェクトで計測されたデータセットをフィットデータとして，fault-prone モジュール判別モデルを構築した際の予測精度を評価する．本稿では

1. 単数のデータセットで判別モデルを構築し，予測値を用いて評価する．
2. 単数のデータセットで判別モデルを構築し，予測値の平均を用いて評価する．
3. 複数のデータセットで判別モデルを構築し，予測値を用いて評価する．

の 3 通りを行った．

3.2 データセット

本稿では，NASA IV&V facility Metrics Data Program (MDP) [8] が公開しているデータセットの中から C 言語で開発された 9 個のプロジェクトを実験対象とし

た．各プロジェクトの概略を表 1 に示す．表中のバグ含有率とはモジュール総数に対してのバグを含んだモジュールの数の割合を指す．

表 1　実験に用いたデータセットの概略

プロジェクト名	モジュール数	バグ含有率（%）	SLOC
CM1	505	9.5	20K
JM1	10,878	19.3	18K
KC1	2,107	15.4	25K
MW1	403	7.7	8K
PC1	1,107	6.9	40K
PC2	5,589	0.4	26K
PC3	1,563	10.2	40K
PC4	1,458	12.2	36K
PC5	17,186	3.0	164K

3.3　評価指標

Fault-prone モジュール判別モデルの予測精度の評価指標としては，Alberg Diagram [9] の AUC (Area Under the Curve) を用いた．Alberg Diagram とはバグを含んでいる可能性の高い順（rank-order）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 [9]．Alberg Diagram の例を図 1 に示す．グラフの横軸は rank-order に抽出されたモジュールの割合を表し，縦軸は判別モデルによって抽出された，実際にバグを含んでいたモジュールの割合を表す．このグラフは予測精度が高いほど左上に凸になり，逆に判別モデルがランダムに予測を行った場合，右肩上がりの直線を描く．また，AUC とは曲線下面積のことであり，ある曲線の積分した値を指す．

これらのことから，Alberg Diagram の AUC とは構築されたモデルの予測精度の高さを値で示したもので，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．図 1 の場合，AUC の値は 0.86 であり，非常に高い精度で予測を行ったといえる．

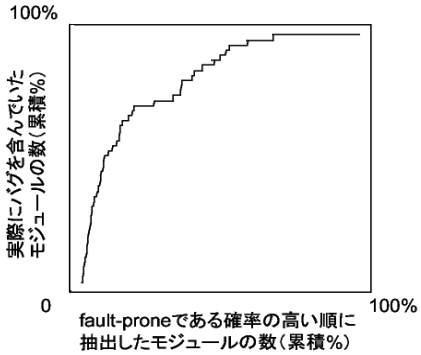

図 1　Alberg Diagram の例

3.4　実験手順

3.4.1　単数のデータセットで判別モデルを構築し，予測値を用いて評価

Step 1　判別モデルの構築

ロジスティック回帰分析を用いて，9 個のデータセットから 9 つの判別モデルを構築する．

Step 2　各モジュールのバグを含む確率の予測

構築した判別モデルを用いて，判別モデル構築に用いたデータセットを除く 8 個

のプロジェクトそれぞれに対して，各モジュールのバグを含む確率を予測する．

Step 3 判別モデルの精度の評価

予測した結果を基に Alberg Diagram を計算し，AUC を算出する．

3.4.2 単数のデータセットで判別モデルを構築し，予測値の平均を用いて評価

Step 1 判別モデルの構築

ロジスティック回帰分析を用いて，9 個のデータセットから 9 つの判別モデルを構築する．

Step 2 各モジュールのバグを含む確率の予測

構築した判別モデルを用いて，判別モデル構築に用いたデータセットを除く 8 個のプロジェクトそれぞれに対して，各モジュールのバグを含む確率を予測し，構築した各判別モデルの，各モジュールに対するバグを含む確率の平均を予測値とする．

Step 3 判別モデルの精度の評価

予測した結果を基に Alberg Diagram を計算し，AUC を算出する．

3.4.3 複数のデータセットで判別モデルを構築し，予測値を用いて評価

Step 1 判別モデルの構築

予測対象データセットを除く 8 個のデータセットを統合し，1 個の判別モデルを構築する．

Step 2 各モジュールのバグを含む確率の予測

構築した fault-prone モジュール判別モデルを用いて，予測対象データセットの各モジュールのバグを含む確率を予測する．

Step 3 判別モデルの精度の評価

予測した結果を基に Alberg Diagram を計算し，AUC を算出する．

4 実験結果

4.1 単数のデータセットで判別モデルを構築し，予測値を用いて評価した場合の実験結果

各プロジェクトで収集されたデータセットを用いて判別モデルを構築し，異なるプロジェクトで収集されたデータセットそれぞれに判別モデルを適用した場合の AUC を表 2 に示す．太字は AUC が 0.5 以上のものを示す．

表 2 単数のデータセットで判別モデルを構築し，予測値を用いて評価した場合の実験結果

	テストデータ								
フィットデータ	CM1	JM1	KC1	MW1	PC1	PC2	PC3	PC4	PC5
CM1	-	**0.56**	**0.76**	**0.72**	**0.65**	**0.67**	**0.65**	**0.57**	**0.79**
JM1	**0.75**	-	**0.78**	**0.78**	**0.73**	**0.87**	**0.77**	**0.75**	**0.90**
KC1	**0.63**	**0.61**	-	**0.69**	0.47	0.19	0.48	0.29	**0.66**
MW1	**0.80**	**0.62**	**0.67**	-	**0.70**	**0.67**	**0.70**	0.46	**0.73**
PC1	**0.73**	0.49	0.43	**0.69**	-	**0.67**	**0.78**	**0.66**	0.49
PC2	0.35	0.39	0.41	0.33	0.35	-	0.41	0.42	0.29
PC3	**0.76**	**0.54**	**0.66**	**0.77**	**0.78**	**0.75**	-	**0.68**	**0.65**
PC4	0.44	0.45	**0.57**	0.31	**0.56**	**0.87**	**0.54**	-	**0.62**
PC5	**0.76**	**0.63**	**0.78**	**0.70**	**0.70**	**0.88**	**0.74**	**0.71**	-

表 2 に示す実験結果によると，AUC の平均値は 0.62 であり，74%のケース（72 件中の 53 件）は AUC が 0.5 以上（ランダムに予測した場合の AUC は 0.5）であった．この結果より，ランダムに予測する場合と比較して，異なるプロジェクトの実績データに基づく fault-prone モジュール判別が有効であることがわかった．

4.1.1 フィットデータとして利用した場合の結果

表 2 より，判別するデータセットの傾向に寄らず性能の高いモデルの構築に寄与するデータセット，つまりフィットデータとして適しているデータセットが存在することがわかった．その一方で，フィットデータとして適していないデータセットが存在することもわかった．例えば，JM1 をフィットデータに用いた場合，AUC の平均値が 0.79 であり（ランダムに予測した場合の AUC は 0.5），全てのテストデータに対して，AUC が 0.75 以上であった．一方 PC2 をフィットデータに用いた場合，AUC の最高値は 0.41，平均値は 0.37 であり，ランダムに予測した場合と比べて低い値であることから，fault-prone モジュール判別に適していないといえる．

4.1.2 テストデータとして利用した場合の結果

表 2 より，どのようなモデルに対しても判別しやすいデータセット，つまりテストデータとして適しているデータセットが存在することがわかった．一方で，テストデータとして適していないデータセットが存在することもわかった．

例えば，PC2 に対する判別結果は，AUC の平均値が 0.70 であり，8 件中 7 件の AUC が 0.65 以上であった．一方 JM1 に対する判別結果は，AUC の最高値が 0.63，平均値は 0.54 であり，ランダムに予測した場合とほぼ同じ値であることから，fault-prone モジュール判別に適していないといえる．

4.2 単数のデータセットで判別モデルを構築し，予測値の平均を用いて評価した場合の実験結果

各プロジェクトで収集されたデータセットを用いて判別モデルを構築し，異なるプロジェクトで収集されたデータセットそれぞれに判別モデルを適用した場合に，それぞれのモジュールに対してバグを含むと予測した確率の平均値を予測値とした場合の AUC の平均値を表 3 に示す．

表 3　単数のデータセットで判別モデルを構築し，予測値の平均を用いた場合の結果

テストデータ	CM1	JM1	KC1	MW1	PC1	PC2	PC3	PC4	PC5
-	**0.75**	**0.67**	**0.77**	**0.75**	**0.70**	**0.88**	**0.73**	**0.72**	**0.89**

表 3 に示す実験結果によると，AUC の平均値は 0.77 であり，9 件中 9 件で AUC が 0.65 以上（ランダムに予測した場合の AUC は 0.5）であった．ランダムに予測する場合と比較して，他プロジェクトの実績データに基づく fault-prone モジュール判別が有効であることがわかった．

単数のデータセットで予測を行った場合と同様，予測しやすいデータセット，予測しにくいデータセットが存在するが，全てのデータセットに対して，単数のデータセットで予測した場合の AUC の平均値よりも高い精度で予測できており，単数のプロジェクトの予測値の平均値を用いて予測することが，他プロジェクトの実績データに基づく fault-prone モジュール判別が有効であることがわかった．

4.3 複数のデータセットで判別モデルを構築し，予測値を用いて評価した場合の実験結果

異なるプロジェクトで収集したデータセットすべてを 1 つのフィットデータとし，判別モデルを構築した場合の実験結果を表 4 に示す．

表 4　複数のデータセットで判別モデルを構築し，予測値を用いて評価した場合の実験結果

テストデータ	CM1	JM1	KC1	MW1	PC1	PC2	PC3	PC4	PC5
-	**0.76**	**0.70**	**0.78**	**0.78**	**0.72**	**0.88**	**0.75**	**0.63**	**0.89**

表 4 に示す実験結果によると，AUC の平均値は 0.77 であり，9 件中 8 件で AUC が 0.7 以上（ランダムに予測した場合の AUC は 0.5）であった．ランダムに予測す

る場合と比較して，他プロジェクトの実績データに基づく fault-prone モジュール判別が有効であることがわかった．

単数のデータセットを用いた場合や，予測値の平均を用いた場合同様，複数のデータセットを用いた場合にも予測しづらい（テストデータに適していない）データセットが存在した．

5 おわりに

本稿では，異なるプロジェクトのモジュールから計測された実績データを元にモデル構築を行い，fault-prone モジュールの判別を試みた．実験では，NASA IV & V Facility Metrics Data Program が公開しているデータセットの内，9 つのデータセットを用い，自身を除くそれぞれの組み合わせ計 72 通りと，異なる全プロジェクトのデータセットをフィットデータとした場合の 9 通りについて予測モデルを構築しその性能を実験的に評価した．

実験により得られた主な結果および知見は，以下の通りである．

- ランダムに予測する場合と比較して，他プロジェクトの実績データに基づく fault-prone モジュール判別の方が有効である．
- どのようなモデルに対しても予測しやすいデータセット，つまりテストデータとして適しているデータセットが存在する．
- 予測するデータセットの傾向に寄らず性能の高いモデルの構築に寄与するデータセット，つまりフィットデータとして適しているデータセットが存在する．
- フィットデータに適したデータセットと，テストデータに適したデータセットの間には相関が少ない．
- 複数プロジェクトで収集したデータセットを用いて判別モデルを構築することは有効である．

fault-prone モジュール判別モデルにとって，どのようなデータセットがフィットデータやテストデータに適しているのか，その原因について検討していくことが今後の課題である．

謝辞 本研究の一部は、文部科学省「e-Society 基盤ソフトウェアの総合開発」の委託に基づいて行われた．

参考文献

[1] R. A. Fisher, “The Use of Multiple Measurements in Taxonomic Problems”, Annals Eugenics, Vol.7, Part II, pp.179-188, 1936.

[2] 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.

[3] （独）情報処理推進機構ソフトウェア・エンジニアリング・センター，ソフトウェア開発データ白書 2006—IT 企業 1400 プロジェクトの定量データで示す開発の実態，日経 BP 社，2006.

[4] 亀井 靖高, 柗本 真佑, 柿元 健, 門田 暁人, 松本 健一, “Fault-Prone モジュール判別におけるサンプリング法適用の効果,” 情報処理学会論文誌, Vol.48, No.8, pp.2651–2662, 2007.

[5] 木浦 幹雄, 柗本 真佑, 亀井 靖高, 門田 暁人, 松本 健一, “他プロジェクトの実績データに基づく Fault-Prone モジュール判別の試み”, ソフトウェア信頼性研究会 第 4 回ワークショップ, pp.45–53, 2007.

[6] P. L. Li, J. Herbsleb, M. Shaw and B. Robinson, “Experiences and Results from Initiating Field Defect Prediction and Product Test Prioritization Efforts at ABB Inc,” Proc. 28th Int’l Conf. on Softw. Eng. (ICSE’06), pp.413–422, Shanghai, China, 2006.

[7] J. C. Munson and T. M. Khoshgoftaar, “The Detection of Fault-prone Programs”, IEEE Trans. Softw. Eng., Vol.18, No.5, pp.423–433, 1992.

[8] NASA IV&V Facility, Metrics Data Program, http://mdp.ivv.nasa.gov/.

[9] N. Ohlsson and H. Alberg, “Predicting Fault-Prone Software Modules in Telephone Switches”, IEEE Trans. Softw. Eng., Vol.22, No.12, pp.886–894, 1996.

[10] D. E. Rumelhart, G. E. Hinton and R. J. Williams, “Learning Representations by Back-propagating Errors”, Nature, Vol.323, pp.533-536, 1986.