

Fault-Prone モジュール判別における外れ値除去法の比較

杉本 真佑^{†1} 亀井 靖高^{†1}
 門田 暁人^{†1} 松本 健一^{†1}

本論文では, fault-prone モジュール判別モデルの構築における課題の一つである, データセット中の外れ値 (特異なモジュール) による判別性能の低下を改善することを目的とする. そのために, 1つの母集団に対する代表的な外れ値除去法である Mahalanobis Outlier Analysis (MOA) と Local Outlier Factor Method (LOFM) を 2 クラス (fault あり/なし) の判別に拡張した手法と, 2 クラスの判別を前提とした外れ値除去法である Rule-Based Modeling (RBM), 及び, 新たに提案する Cross-Class Mahalanobis Outlier Analysis (CC-MOA) の効果を実験的に比較した. 実験では, NASA (アメリカ航空宇宙局) が公開している 3 つのプロジェクトのデータセットを題材として, MOA, LOFM, RBM, CC-MOA のそれぞれの適用の効果を, 3 つの代表的な判別モデル (線形判別分析, ロジスティック回帰分析, 分類木) について交差検証法により評価した. 実験の結果, LOFM を除く 3 つの外れ値除去法を用いた場合, いずれの判別モデル, データセットに対しても判別精度が改善し, F1 値の平均向上幅は MOA では 0.139, RBM では 0.137, CC-MOA では 0.123 であった.

Comparison of Outlier Detection Methods in Fault-Prone Module Detection

SHINSUKE MATSUMOTO,^{†1} YASUTAKA KAMEI,^{†1} AKITO MONDEN^{†1}
 and KEN-ICHI MATSUMOTO^{†1}

The goal of this paper is to improve the prediction performance of fault-proneness models by removing outliers from a dataset used for model construction. We experimentally evaluated the effect of four outlier removal methods; Mahalanobis Outlier Analysis (MOA) and Local Outlier Factor Method (LOFM) which are well-known outlier detection methods for a single sample, and Rule-Based Modeling (RBM) suitable for two samples, and Cross-Class Mahalanobis Outlier Analysis (CC-MOA) proposed in this paper. In the experiment, we compared MOA, LOFM, RBM and CC-MOA each applied to three well-known fault-proneness models (linear discriminant analysis, logistic regression analysis and classification tree) using three NASA project datasets. As a result, three outlier detection methods excluding LOFM improved F1-values of all fault-proneness models for all datasets. The average improvements of F1-value by MOA, RBM and CC-MOA were 0.139, 0.137 and 0.123 respectively.

1. はじめに

ソフトウェア開発において, 限られた開発期間で信頼性を確保するためには, テスト工程の効率化が重要である¹⁴⁾. その一つの手段は, 欠陥 (fault) を含んでいる可能性の高いモジュール (fault-prone モジュール) を特定し, テスト工数を重点的に割り当てる^{*1} ことである^{12),16)}. そのために, 従来, 多数の fault-prone モジュール判別モデルが提案されている^{5),13),17)}. Fault-

prone モジュール判別モデルは, モジュールの特性値 (ソースコード行数や分岐の数, サイクロマティック数など) を説明変数とし, モジュールの fault の有無を目的変数とする数学的モデルであり, 線形判別分析, ロジスティック回帰分析, 分類木などの手法が用いられる. 判別モデルの構築 (モデルのパラメータの推定) は, 過去のソフトウェア開発における各モジュールの特性値と fault の有無を記したデータセット (以降, フィットデータ) を用いて行われる.

本論文では, 判別モデルの構築における課題の一つである外れ値の問題に着目する. 一般に, フィットデータには外れ値 (特異な特徴を持つモジュールのこと. 例えば, ソースコード行数と分岐の数が著しく大きい) にも関わらず fault が発見されていないモジュールな

^{†1} 奈良先端科学技術大学院大学 情報科学研究科
 Graduate School of Information Science, Nara Institute of Science and Technology

^{*1} テスト工数の割り当て方法とその効果については文献 18) を参照にされたい.

ど)が含まれており,この外れ値が判別モデルの精度を低下させる原因となることが指摘されている^{9),11)}.このため,外れ値はあらかじめフィットデータから除去した上で,判別モデルを構築することが望ましい.

従来,多変量データの標本から外れ値(特異な個体)を検出し,除去する手法(外れ値除去法)が多数提案されている^{2),10)}が, fault-prone モジュール判別モデルの精度を向上するという目的には必ずしも利用できない. Fault-prone モジュール判別モデルにおけるフィットデータでは, fault を含むか否かによって「特異である」と判断すべき特徴が異なると考えられるためである.例えば,ソースコード行数が短く分岐やループも少ないモジュールは,一般に fault が混入しにくい.そのため, fault を含まない場合には特異とはいえない.ところが,逆に, fault を含むモジュールとしては特異であるといえる.

そこで,本論文では, fault を含む標本と含まない標本のそれぞれに対し,外れ値除去法を独立に適用することを考える.これにより, fault を含む/含まない場合のそれぞれにおいて特異とみなせるモジュールの除去が期待できる.本論文では,代表的な外れ値除去法である Mahalanobis Outlier Analysis (MOA)⁸⁾と Local Outlier Factor Method (LOFM)²⁾を適用した場合の効果を評価する.

さらに,本論文では,標本の中に2つのクラス(本論文の場合は, fault あり/なしの2クラス)があることを前提とした,新たな外れ値除去法 Cross-Class Mahalanobis Outlier Analysis (CC-MOA)を提案する.この手法では,一方のクラスに含まれるモジュールが,他方のクラスの重心に近い場合に外れ値であるとみなす(3.4節).このような2クラスの判別問題を対象とした外れ値除去法としては,他に,Rule-Based Modeling (RBM)が提案されている⁹⁾(3.3節).RBMは,CBR (Case-Based Reasoning)モデルに対する効果は実験的に示されている⁹⁾が,他の判別モデルに対する効果は不明である.

そこで,本論文では,前述の4つの外れ値除去法(MOA, LOFM, RBM, CC-MOA)を,3つの代表的な fault-prone モジュール判別モデル(線形判別分析⁴⁾,ロジスティック回帰分析⁷⁾,分類木¹⁾)に適用した場合の計12通りの判別精度を実験的に比較し,いずれの外れ値除去法が fault-prone モジュール判別において最も効果があるかを明らかにする.実験にはNASA(アメリカ航空宇宙局)が公開しているモジュールの特性値と fault の有無を記したデータセット¹⁵⁾のうち,3つ(モジュール数の多いものから3プロジェクト

ト)のデータセットを用いた.

以降,2章で実験に用いる fault-prone モジュール判別モデルについて説明し,3章で比較対象とする外れ値除去法について説明する.4章で実験の方法と手順について述べ,5章でその実験結果について述べる.6章で実験結果についての考察を行い,最後に7章で本稿のまとめと今後の課題を述べる.

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,モジュールが fault を含んでいるか否かを判別することを目的とし,過去に開発されたモジュールの特性値と fault の有無を記したデータセットを用いて構築される.これまでに多数の判別モデルが提案されているが,本論文では fault-prone モジュール判別モデルとして広く用いられている3つのモデリング手法を採用する.

2.1 線形判別分析

線形判別分析では,2クラスの標本を判別する境界を直線として表現する⁴⁾.個体 x の説明変数を x_i ,説明変数の総数を p とした場合,判別関数は以下のように表される.

$$Z = \alpha + \beta_1 x_1 + \dots + \beta_p x_p \quad (1)$$

ここで α は定数, β_i は判別係数であり,判別モデルの構築時に α と β_i が決定される.判別対象の個体は,判別値 Z が正であるか負であるかによって2値のいずれに属するか判別される.

2.2 ロジスティック回帰分析

ロジスティック回帰分析では,2クラスの標本を判別する判別関数にロジスティック関数を用いて表現する⁷⁾.

$$P(y|x) = \frac{1}{1 + e^{-(\alpha + \beta_1 x_1 + \dots + \beta_p x_p)}} \quad (2)$$

ここで, $y \in \{0, 1\}$ は2値のクラスを表す目的変数であり, $P(y|x)$ は個体 x に対して y が1のクラスに属する確率である.本論文では, $P(y|x)$ の値が0.5を超える場合に,個体 x が1のクラスに属すると判別する.

2.3 分類木

分類木は説明変数と目的変数の関係を木構造で表す非線形モデルである¹⁾.木の各ノードは2個以上の子ノードを持ち,説明変数の値によっていずれの子ノードへ分岐し,リーフノードには2値のいずれかが割り当てられる.判別対象の個体は説明変数の値によってノードをたどり,リーフノードによって2値のいずれに属するか判別される.本論文では,分類木の構築アルゴリズムには CART (Classification And Regres-

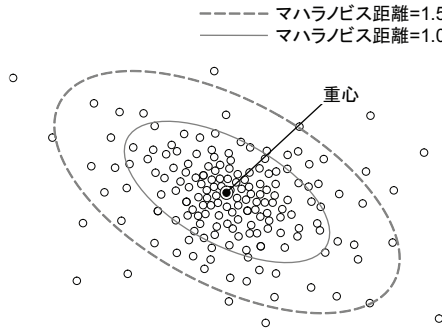


図 1 マハラノビス距離の例

Fig. 1 An example of Mahalanobis distance.

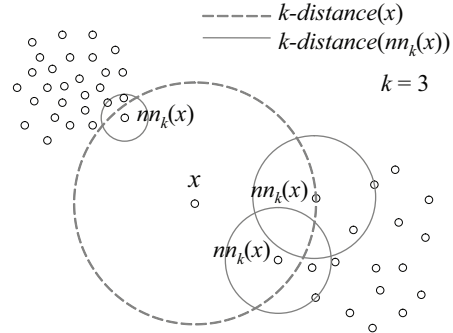


図 2 Local outlier factor の例

Fig. 2 An example of local outlier factor.

sion Trees)¹⁾ を、ノードの分岐の基準には Gini 係数を用いた¹⁾。

3. 外れ値除去法

外れ値除去法とは、標本の中から他の個体と比べて特異な傾向を持った個体を、外れ値として検出し除去する方法である。以降、実験に用いた 3 つの既存の外れ値除去法と本論文で提案する新たな外れ値除去法について説明する。

3.1 Mahalanobis Outlier Analysis

Mahalanobis Outlier Analysis (MOA) とは、多変量データを対象とし、マハラノビス距離が閾値 θ_{MOA} を超える個体を外れ値と見なし除去する手法である⁸⁾。

マハラノビス距離とは、標本の重心から計算対象とする個体までの距離を、標本のばらつき具合に基づいて算出される距離尺度である。図 1 に、ある標本に対するマハラノビス距離の例を示す。図 1 の点は個体を表し、楕円はマハラノビス距離の等高線を表す。このように個体が楕円状に分布している場合、マハラノビス距離の等高線は楕円を描くように算出される。

\bar{x} を各変数の平均値のベクトル、 x_i を個体 i の持つ変数のベクトル、 n を個体の総数、 T を転置ベクトルとしたとき、共分散行列 S は

$$S = (n - 1)^{-1} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (3)$$

と表され、個体 i のマハラノビス距離 MD_i は

$$MD_i = \sqrt{(x_i - \bar{x})^T S^{-1} (x_i - \bar{x})} \quad (4)$$

と表される。 MD_i は平均が 1 になるように正規化され、個体 i が重心と同じ場所に位置する場合 MD_i は最小 ($MD_i = 0$) となり、重心から離れた個体ほど大きな値をとる。これらのことから、MOA とは標本

の重心から離れた個体を外れ値と見なし除去する手法であるといえる。

本論文では、fault を含むクラスと含まないクラスのそれぞれに対し、MOA を独立に適用することで、各クラスの外れ値の除去を行う。

3.2 Local Outlier Factor Method

Local Outlier Factor (LOF) とは、個体ごとの k -nearest neighbor との分布密度に基づいて算出される値であり、各個体が標本の中からどの程度外れているかを表す尺度である²⁾。本論文では、LOF が閾値 θ_{LOFM} を超える個体を外れ値と見なし除去する手法を LOF Method (LOFM) と呼ぶ。

図 2 は $k = 3$ のときの LOF の例である。個体 x に対する k -nearest neighbor を $nn_k(x)$ 、個体 x と個体 $nn_k(x)$ の最大の距離を k -distance(x) と表す。ここで、個体 x の LOF の値 $LOF_k(x)$ は、 k -distance(x) が k -distance($nn_k(x)$) に比べて大きい場合に 1 を超える値をとるように算出される。他の個体から孤立して位置する個体は k -distance(x) が k -distance($nn_k(x)$) に比べて大きく、 $LOF_k(x) \gg 1$ となる。逆に一定の密度でばらついている個体は k -distance(x) と k -distance($nn_k(x)$) が近い値となるため、 $LOF_k(x) \approx 1$ となる。図 2 の個体 x の場合、 k -distance(x) が k -distance($nn_k(x)$) に比べ非常に大きいため、 $LOF_k(x) \gg 1$ となる。これらのことから、LOFM は疎な空間に存在する個体を外れ値とみなす手法であるといえる。

本論文では、fault を含むクラスと含まないクラスのそれぞれに対し、LOFM を独立に適用する。

3.3 Rule-Based Modeling

Khoshgoftaar らは 2 クラス判別問題のための外れ値除去法として、Rule-Based Modeling (RBM) を提案している⁹⁾。ここで 2 つのクラスとは fault を含

表 1 Rule-Based Modeling で用いる Boolean ルールの例
Table 1 An example of Boolean rules for Rule-Based Modeling.

#	ルールの内容
1	$(x_1 \leq c_1) \wedge (x_2 \leq c_2) \Rightarrow nfp$
2	$(x_1 \leq c_1) \wedge (x_2 > c_2) \Rightarrow nfp$
3	$(x_1 > c_1) \wedge (x_2 \leq c_2) \Rightarrow nfp$
4	$(x_1 > c_1) \wedge (x_2 > c_2) \Rightarrow fp$

む (fp) か, 含まない (nfp) かを指す. 従来, RBM は CBR (Case-Based Reasoning) モデルに対する効果は実験的に示されている⁹⁾ が, 他の判別モデルに対する効果は不明である.

RBM では, まず, 全ての変数に対して 2 つのクラスを最も良く分割する閾値を, fault ありモジュールの総数と fault なしモジュールの総数の割合に基づき 2 標本 Kolmogorov-Smirnov 検定³⁾ により計算する. 次に, 計算された閾値で区切られた空間ごとに, fault あり/なしのどちらに属するかを判別する Boolean ルールを構築する. 2 つの変数を持つ標本の場合の Boolean ルールの例を, 表 1 に示す. 変数の数が 2 つの場合, 2 つの閾値 (c_1, c_2) を用いた 4 つの Boolean ルールが構築される. 最後に, 標本の中から $(x_1 \leq c_1) \wedge (x_2 \leq c_2) \wedge fp$ のような, ルールに反する (この場合ルール 1 に反する) 個体を外れ値と見なし除去する.

つまり RBM は, 自身の属するクラスと異なったクラスの特徴を持つと判定された個体の全てを, 外れ値とみなす手法である. また, 変数ごとの閾値は Kolmogorov-Smirnov 検定を用いて自動的に決定されるため, MOA や LOFM と異なり閾値決定のための事前実験が不要である.

3.4 Cross-Class Mahalanobis Outlier Analysis

本論文ではこれまでに提案されている上記の 3 つの手法に加え, Cross-Class Mahalanobis Outlier Analysis (CC-MOA) を提案する. CC-MOA は, RBM と同様, 2 つのクラス (fault あり/なし) が存在することを前提とした手法である. 図 3 に CC-MOA の例を示す. CC-MOA では, あるクラスの個体 (例えば, fault を含むモジュール) が, もう一方のクラスの標本 (fault を含まないモジュール) の重心に近い個体を外れ値であるとみなす. つまり, 他のクラスからのマハラノビス距離が閾値 θ_{CC-MOA} を下回る個体を外れ値とみなす手法である.

CC-MOA の手順を以下に示す.

Step 1. クラス α に属する個体 x_α に対して, クラ

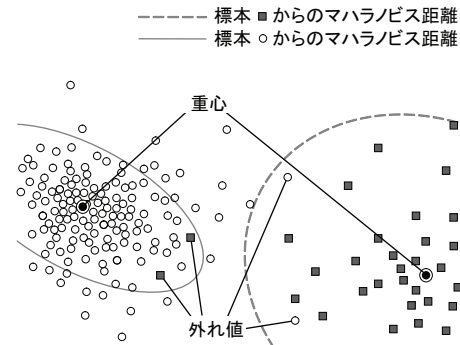


図 3 Cross-Class Mahalanobis Outlier Analysis の例
Fig. 3 An example of Cross-Class Mahalanobis Outlier Analysis.

ス β からのマハラノビス距離 MD_{x_α} を計算する.

Step 2. MD_{x_α} が θ_{CC-MOA} を下回る場合, x_α を外れ値とみなす.

Step 3. 全ての x_α に対して Step 1 から Step 2 を繰り返す.

Step 4. α と β を入れ替えて Step 1 から Step 3 を実行する.

MOA (3.1 節) は自身の属するクラスからのマハラノビス距離を用いるのに対して, CC-MOA では他のクラスからのマハラノビス距離を用いる点が異なる. これにより, 例えば fault を含むにも関わらず, fault を含まないモジュール群と似た特徴を持つような個体を外れ値として検出することが可能となる.

4. 実験

4.1 実験概要

実験の目的は, 代表的な fault-prone モジュール判別モデルに対して, 最も判別精度の向上につながる外れ値除去法を明らかにすることである. そのために, 4 つの外れ値除去法 (MOA, LOFM, RBM, CC-MOA) と, 3 つの fault-prone モジュール判別モデル (線形判別分析, ロジスティック回帰分析, 分類木) の組み合わせの合計 12 通りについて判別精度の比較を行った.

実験では, まず RBM を除く閾値を要する外れ値除去法 (MOA, LOFM, CC-MOA) に対して閾値 ($\theta_{MOA}, \theta_{LOFM}, \theta_{CC-MOA}$) を決定するための事前実験を行った. 次に, 本実験としてモデル構築用のフィットデータに対して, 事前実験で決定した閾値を用いて外れ値除去法を適用した. 最後に, 外れ値除去済みのフィットデータを用いて判別モデルを構築し, モデル評価用のテストデータを用いてその判別精度を求めた (交差検証法). 実験の信頼性を確保するために, 事前実験と本実験をそれぞれ 10 回ずつ繰り返した.

表 2 実験に用いたデータセットの概要

Table 2 Summary of datasets.

	KC1	JM1	PC5
モジュール総数	2,107	10,878	17,186
Fault なしモジュール数	1,782	8,776	16,670
Fault ありモジュール数	325	2,102	516
ソースコードメトリクス数	21	21	40

表 3 実験に用いたデータセットのメトリクス

Table 3 Source code metrics in each dataset.

	KC1	JM1	PC5
Branch count	*	*	*
Call pairs			*
Condition count			*
Cyclomatic complexity	*	*	*
Cyclomatic density			*
Decision count			*
Decision density			*
Design complexity	*	*	*
Design density			*
Edge count			*
Essential complexity	*	*	*
Essential density			*
Global data complexity			*
Global data density			*
Halstead content	*	*	*
Halstead difficulty	*	*	*
Halstead effort	*	*	*
Halstead error estimation	*	*	*
Halstead length	*	*	*
Halstead level	*	*	*
Halstead programming time	*	*	*
Halstead volume	*	*	*
LOC blank	*	*	*
LOC code and comment	*	*	*
LOC comment	*	*	*
LOC executable	*	*	*
LOC total	*	*	*
Maintenance severity			*
Modified condition count			*
Multiple condition count			*
Node count			*
Normalized cyclomatic count			*
Number of operands	*	*	*
Number of operators	*	*	*
Number of unique operands	*	*	*
Number of unique operators	*	*	*
Number of lines			*
Parameter count			*
Pathological complexity			*
Percent comment			*

LOF を算出する際に用いる k -nearest neighbor の k については, Breunig らの決定方法²⁾ に従ってあらかじめ実験を行い, $k = 30 \sim 50$ とした.

4.2 データセット

実験には NASA/WVU IV&V Facility Metrics Data Program (MDP) が公開しているデータセッ

表 4 判別結果の分類

Table 4 Classification of detection results.

		判別値	
		fault なし	fault あり
実測値	fault なし	n_{11}	n_{12}
	fault あり	n_{21}	n_{22}

ト¹⁵⁾のうち, モジュール数の多いものから順に3つのデータセット(プロジェクト KC1, JM1, PC5)を用いた^{*1}. 各データセットの概要を表2に示す. また, 説明変数として用いたソースコードメトリクスの一覧を表3に示す. 表3の「*」は各データセット中において記録されているソースコードメトリクスを表す.

実験では, fault の有無を目的変数, ソースコードメトリクスを説明変数として判別モデルを構築する. 判別モデルを構築する際には, 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, もう一方をテストデータとする.

4.3 評価基準

Fault-prone モジュール判別モデルの評価基準として, 再現率, 適合率, F1 値⁶⁾を用いる. 再現率とは fault を含んでいるモジュールのうち, 正しく fault-prone と判別したモジュールの割合であり, 表4に示す記号を用いると以下のように定義される.

$$Recall = \frac{n_{22}}{n_{21} + n_{22}} \quad (5)$$

適合率とは fault-prone と判別されたモジュールのうち, 実際に fault を含んでいるモジュールの割合であり, 以下のように定義される.

$$Precision = \frac{n_{22}}{n_{12} + n_{22}} \quad (6)$$

これら再現率と適合率は一方が高くなると, もう一方が低くなりやすいという関係にある. そこで本論文では, 再現率と適合率の調和平均である F1 値を評価基準として用いる. F1 値は以下のように定義される.

$$F\text{-measure} = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (7)$$

3つの評価基準は値域 $[0, 1]$ を取り, 値が高いほど判別精度が高いことを表す.

4.4 実験手順

4.4.1 閾値の決定手順

外れ値除去法の閾値 (θ_{MOA} , θ_{LOFM} , θ_{CC-MOA}) を決定するために事前実験を行う. 事前実験では本実

*1 MDP における公開データセットでは, プロジェクト PC2 と MC1 もモジュール数が多いが, fault を含むモジュールがほとんど含まれていなかった(1%未満)ため, 本論文では用いなかった.

験で用いるフィットデータのみを用いて、判別モデルの構築および評価を行い、最も F1 値が高くなる閾値を求める。この閾値は、fault-prone モジュール判別モデルと外れ値除去法の全ての組み合わせについて決定される。閾値の決定手順を以下に示す。

Step 1. フィットデータ fit を、 fit_A と fit_B の 2 つにランダムに二等分する。

Step 2. fit_A に対して外れ値除去法を適用し、 fit'_A を作成する。

Step 3. fit'_A を用いて、判別モデルを構築する。

Step 4. fit_B を用いて、構築した判別モデルの判別精度を求める。

Step 5. 閾値を変化させて、Step 2 から Step 4 を繰り返す。

Step 6. Step 1 から Step 5 を 10 回繰り返し、F1 値の平均値が最も高かった閾値を本実験に用いる閾値とする。

4.4.2 本実験の手順

4.4.1 項で決定した閾値を用いて、外れ値除去法と fault-prone モジュール判別モデルの全ての組み合わせ 12 通りについて判別精度を求める。本実験の手順を以下に示す。

Step 1. データセットを、フィットデータ fit とテストデータ $test$ の 2 つにランダムに二等分する。

Step 2. fit に対して事前実験で決定した閾値を用いて外れ値除去法を適用し、 fit' を作成する。

Step 3. fit' を用いて、判別モデルを構築する。

Step 4. $test$ を用いて、構築した判別モデルの判別精度を求める。

Step 5. Step 1 から Step 4 を 10 回繰り返し、判別精度（再現率、適合率、F1 値）の平均値をそれぞれ求める。

5. 実験結果

5.1 事前実験の結果

4.4.1 項で述べた事前実験により決定された各外れ値除去法ごとの閾値を表 5 に示す。PC5 に対して LOFM を適用した場合については、閾値の計算が終了しなかったため、記載していない^{*1}。この原因は、LOFM が他の手法より大きな計算量を要することと、PC5 のモジュール数とソースコードメトリクス数が他のデータセットより大きかったためと考えられる。また、RBM については、閾値決定のための事前実験が不要

表 5 事前実験で決定した閾値

Table 5 Thresholds determined in preliminarily experiments.

		線形判別 分析	ロジスティック 回帰分析	分類木
KC1	θ_{MOA}	0.3	0.3	0.4
	θ_{LOFM}	1.2	1.2	1.2
	θ_{CC-MOA}	1.3	1.2	1.2
JM1	θ_{MOA}	0.3	0.3	0.4
	θ_{LOFM}	1.0	1.0	1.0
	θ_{CC-MOA}	0.4	0.8	0.8
PC5	θ_{MOA}	9.8	5.7	2.3
	θ_{LOFM}	—	—	—
	θ_{CC-MOA}	0.7	0.9	0.5

であるため、記載していない。

表 5 より、KC1 と JM1 に対しての各外れ値除去法の最適な閾値は、判別モデル間で大きな違いがなかったが、PC5 については判別モデル間でばらつきがあった。また、データセット間でも、各外れ値除去法の閾値にはばらつきがあった。このことから、いずれの外れ値除去法においても、最適な閾値はデータセットと判別モデルに依存するため、事前実験により閾値を決定することが必須といえる。

5.2 本実験の結果

4.4.2 項で述べた本実験の結果をデータセット KC1, JM1, PC5 のそれぞれについて表 6, 表 7, 表 8 に示す。表中の未適用とは、外れ値を除去せずに判別モデルを構築した場合の判別精度を表す。また有意水準 5% で平均値の差を検定した結果、未適用と比べて判別精度が有意に向上したものを太字体で表し、有意に低下したものを斜字体で表す。以降、判別精度の変化について、外れ値除去法ごとと判別モデルごとに述べる。

5.2.1 外れ値除去法間の比較

表 6 に示すとおり KC1 では、MOA, RBM, CC-MOA はいずれの判別モデルに対しても適合率が若干減少した ($-0.043 \sim -0.333$) もの、再現率がそれ以上に向上し ($+0.099 \sim +0.538$)、これらの調和平均である F1 値も向上した ($+0.042 \sim +0.188$)。このことは、JM1 (表 7), PC5 (表 8)、においても同様であった。一方で、LOFM はいずれのデータセット、判別モデルにおいても判別精度の有意な変化が (適合率, 再現率, F1 値の全てにおいて) 見られなかった。以上のことから、MOA, RBM, CC-MOA はデータセットおよび判別モデルに関わらず判別精度向上の効果が、逆に、LOFM は、fault-prone モジュール判別問題に対しては効果が期待できないといえる。

LOFM 以外の外れ値除去手法 (MOA, RBM, CC-MOA) による判別精度向上の効果は、ほぼ同程度であった。4 つの外れ値除去法それぞれの F1 値平均

*1 Pentium 4, 3GHz の計算機を用いて 100 時間以内に結果が得られなかった。

表 6 判別モデルに対しての外れ値除去法適用の効果 (KC1)
Table 6 Result of outlier detection for dataset KC1.

		線形判別 分析	ロジスティック 回帰分析	分類木
再現率	未適用	0.277	0.154	0.241
	MOA	0.376	0.389	0.378
	LOFM	0.276	0.160	0.301
	RBM	0.622	0.692	0.711
	CC-MOA	0.743	0.479	0.640
適合率	未適用	0.557	0.652	0.483
	MOA	<i>0.452</i>	<i>0.452</i>	<i>0.440</i>
	LOFM	0.515	0.625	0.460
	RBM	<i>0.342</i>	<i>0.389</i>	<i>0.314</i>
	CC-MOA	<i>0.344</i>	<i>0.381</i>	<i>0.301</i>
F1 値	未適用	0.367	0.247	0.317
	MOA	0.409	0.416	0.402
	LOFM	0.356	0.252	0.357
	RBM	0.440	0.435	0.434
	CC-MOA	0.437	0.405	0.407

を (MOA, LOFM, RBM, CC-MOA) の順でデータセットごとに述べると, KC1 では (0.409, 0.322, 0.406, 0.416) であり, JM1 では (0.417, 0.164, 0.412, 0.361), PC5 では (0.460, —, 0.458, 0.4472) であった.

5.2.2 判別モデル間の比較

外れ値除去法を適用しない場合, 判別モデル間の判別精度には大きなばらつきが見られた. 特に, ロジスティック回帰分析は, いずれのデータセットにおいても線形判別分析と比べて F1 値が有意に小さかった. ところが, このモデル間の差は, 外れ値除去法を用いることで小さくなった. 例えば, KC1 (表 6) に対して外れ値除去法を適用しない場合の判別モデルごとの精度を (線形判別分析, ロジスティック回帰分析, 分類木) の順で並べると (0.367, 0.247, 0.317) であったが, MOA を用いた場合には (0.409, 0.416, 0.402) となり, 0.4 付近に収束した. 同様に, RBM では (0.440, 0.435, 0.434), CC-MOA では (0.437, 0.405, 0.407) となった. この傾向は, 他のデータセット JM1, PC5 でも同様であった. このことから, 外れ値による影響 (判別精度の低下の度合い) は判別モデルによって異なっていたが, 外れ値を除去することで, いずれの判別モデルも性能を発揮できるようになり, 結果として性能の違いがなくなったといえる.

5.3 本実験でのモジュール除去率

各外れ値除去法によって除去されたモジュールの割合 (除去率) を表 9 に示す. モジュールの総数を N_{all} , 除去されたモジュールの数を N_{rm} としたとき, 除去率 P_{rm} は以下のように表される.

$$P_{rm} = \frac{N_{rm}}{N_{all}} \quad (8)$$

除去率はデータセット間で大きな差があった. 例えば, KC1 では除去率が 20.3% ~ 74.0% であったのに

表 7 判別モデルに対しての外れ値除去法適用の効果 (JM1)
Table 7 Result of outlier detection for dataset JM1.

		線形判別 分析	ロジスティック 回帰分析	分類木
再現率	未適用	0.139	0.072	0.089
	MOA	0.483	0.474	0.402
	LOFM	0.146	0.067	0.100
	RBM	0.401	0.547	0.625
	CC-MOA	0.832	0.691	0.779
適合率	未適用	0.561	0.648	0.559
	MOA	<i>0.375</i>	<i>0.381</i>	<i>0.413</i>
	LOFM	<i>0.508</i>	<i>0.555</i>	<i>0.361</i>
	RBM	<i>0.398</i>	<i>0.358</i>	<i>0.323</i>
	CC-MOA	<i>0.228</i>	<i>0.226</i>	<i>0.251</i>
F1 値	未適用	0.222	0.129	0.158
	MOA	0.422	0.422	0.406
	LOFM	0.223	0.117	0.153
	RBM	0.399	0.432	0.406
	CC-MOA	0.357	0.351	0.374

表 8 判別モデルに対しての外れ値除去法適用の効果 (PC5)
Table 8 Result of outlier detection for dataset PC5.

		線形判別 分析	ロジスティック 回帰分析	分類木
再現率	未適用	0.538	0.182	0.369
	MOA	0.538	0.352	0.618
	LOFM	—	—	—
	RBM	0.747	0.824	0.820
	CC-MOA	0.743	0.742	0.602
適合率	未適用	0.393	0.615	0.542
	MOA	0.413	<i>0.523</i>	<i>0.415</i>
	LOFM	—	—	—
	RBM	<i>0.361</i>	<i>0.309</i>	<i>0.298</i>
	CC-MOA	<i>0.344</i>	<i>0.298</i>	<i>0.415</i>
F1 値	未適用	0.454	0.279	0.437
	MOA	0.467	0.420	0.493
	LOFM	—	—	—
	RBM	0.487	0.449	0.437
	CC-MOA	0.470	0.424	0.490

対し, PC5 では 0.7% ~ 21.7% と小さかった. データセット間の fault を含むモジュールの割合の違いが影響している可能性がある.

外れ値除去法の違いによっても, 除去率には大きな差があった. 例えば, MOA は除去率が 0.7% ~ 37.5% であったのに対し, CC-MOA では 80% を超える場合もあった. ただし, 前節で述べたように MOA と CC-MOA では判別精度への効果に大きな差がなかったことから, 除去率が判別精度へ与える影響は小さいと考えられる.

6. 考 察

実験結果より, MOA, RBM, CC-MOA はいずれのデータセット, 及び, 判別モデルに対しても, 適合率が若干減少したものの, 再現率がそれ以上に向上し, これらの調和平均である F1 値も向上した. 一般に, fault-prone モジュールの判別問題においては, 再現率の低下と比べると, 適合率の低下は許容される¹³⁾ ため, 適合率の少しの低下と引き換えに再現率をより大きく向上できたことは, テスト効率の改善に有効で

表 9 外れ値として除去されたモジュールの割合 (%)
Table 9 Percentage of removed modules (%).

		線形判別	ロジスティック	分類木
		分析	回帰分析	
KC1	MOA	20.3	37.5	27.7
	LOFM	55.5	55.5	55.5
	RBM	30.3	30.3	30.3
	CC-MOA	74.0	71.6	71.6
JM1	MOA	32.6	32.6	23.6
	LOFM	37.9	37.9	37.9
	RBM	33.1	33.1	33.1
	CC-MOA	69.5	80.5	80.5
PC5	MOA	0.7	1.4	5.5
	LOFM	—	—	—
	RBM	6.9	6.9	6.9
	CC-MOA	18.7	21.7	11.2

あると考える。

さらに、MOA、RBM、CC-MOA は、モデル間の判別精度の差を小さくする効果があった。いずれのデータセット、判別モデルにおいても、MOA、RBM、CC-MOA により外れ値を除去した場合、F1 値は向上し、0.4 付近に収束した。従来、最適な判別モデルはデータセットによって異なることが指摘されており⁵⁾、データセットごとに適した判別モデルを見つけることが一つの課題であったが、外れ値除去によって判別モデル間の性能差がなくなり、モデル利用者にとって利便性が増すことが期待される。特に、RBM は閾値決定のための事前実験が不要であるという特徴を持つため、最初に試す手法として有力と考えられる。

外れ値除去によってモデル間の性能差が小さくなったことに対する一つの解釈としては、外れ値のないデータセットはそもそも 2 クラスの判別が容易であるため、いずれの種類のモデルにおいても、性能の良い (2 クラスをうまく判別する) モデルが得られた可能性がある。一方で、判別対象のテストデータには依然として外れ値が含まれており、たとえ性能の良いモデルであっても達成可能な判別精度には限界があったために、得られた F1 値は 0.4 付近にとどまった可能性がある。

次に、LOFM では判別精度向上の効果がほとんど見られなかった原因について、外れ値除去による各クラスのモジュールの分布の変化に着目して考察を行う。判別モデルと外れ値除去法の全ての組合せについて確認することは困難であるため、ここではデータセット KC1 に対して線形判別分析を用いた場合を取り上げる。

外れ値除去法を適用しない場合のモジュールのヒストグラムを図 4 に、各外れ値除去法によって除去されたモジュールのヒストグラムを図 5 に、除去法適用後の残ったモジュールのヒストグラムを図 6 に示す。白

の棒は fault ありモジュール、灰の棒は fault なしモジュールを表す。横軸には、議論の簡単化のために最も直感的な特性値である SLOC (ソースコード行数) を取った。

まず、図 4 より、MOA では fault なしモジュール群の重心は SLOC = 0 付近にあり、fault ありモジュール群の重心は SLOC = 10 ~ 20 付近であった。図 5 より、各群において、重心付近のモジュールは除去されず、重心から離れたモジュールが除去されたことが窺える。SLOC 以外の特性値においても、fault あり / なしによって重心位置に差があり、各クラスの重心付近のモジュールが除去されず残された結果、2 クラスの判別精度が向上した可能性がある。

次に、LOFM に着目すると、fault あり / なしのいずれのモジュール群についても、外れ値除去前のモジュール群 (図 4)、除去されたモジュール群 (図 5)、除去後に残ったモジュール群 (図 6) のヒストグラムはほぼ同じ形であった。このことから、LOFM では、モジュールのサイズや fault あり / なしに関わらずほぼランダムに除去が行われ、その結果、判別精度に変化が見られなかったことが窺える。LOFM は、「近隣の個体と比べて疎な空間にある個体を除去する」方式であり、クラスタから外れた個体を外れ値として特定するのに向いている。ところが、今回のデータセットでは、サイズの小さなモジュールから大きなモジュールまで途切れなく存在していたため、大きなクラスタがそもそも存在せず、巨視的にはランダムに近い除去が行われた可能性がある。

次に、RBM では SLOC = 0 ~ 30 の辺りを閾値として、サイズの大きな fault なしモジュール、及び、サイズの小さな fault ありモジュールの全てが除去されている。これは、RBM における Boolean ルールの一つに、SLOC を含むものがあったためと考えられる。一般に、サイズの小さいモジュールほど fault を含む割合が低く、また、サイズが大きくなるほど fault を含む割合が高くなることから、「サイズの大きな fault なしモジュール」と「サイズの小さな fault ありモジュール」は、ともに外れ値とみなすことが妥当であり、RBM は、SLOC に関しては、外れ値の除去が効果的に行えたことが示唆される。

CC-MOA に関しては、fault なしモジュールのうち、サイズの非常に小さなモジュール以外を大量に除去している。また、fault ありモジュールについては、サイズの小さなモジュールを全般的に除去している。その結果、RBM と同様のモジュールが除去され、判別精度の向上につながったことが窺える。なお、表 9 に

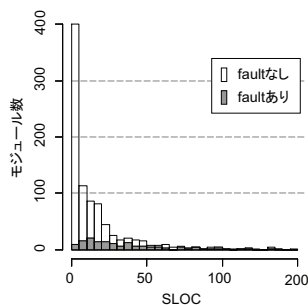


図 4 フィットデータ内の全モジュールのヒストグラム

Fig. 4 Histogram of all modules in a fit dataset.

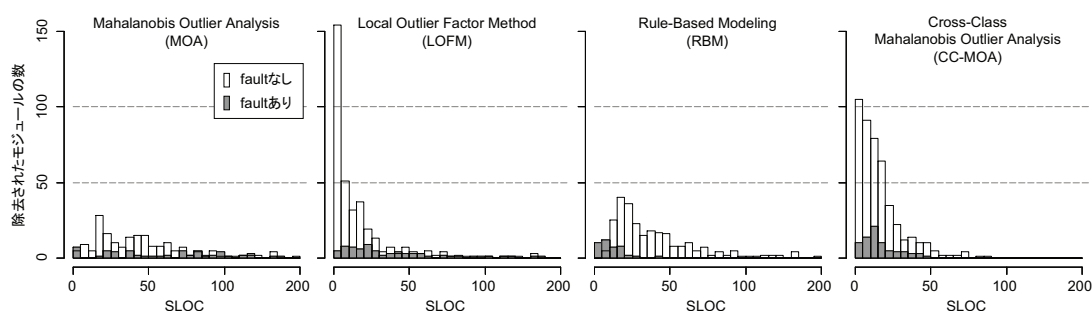


図 5 外れ値除去法によって除去されたモジュールのヒストグラム

Fig. 5 Histogram of removed modules.

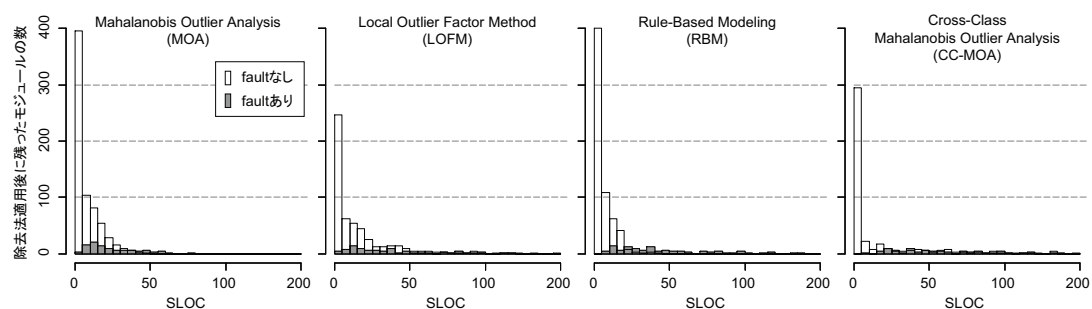


図 6 外れ値除去法適用後に残ったモジュールのヒストグラム

Fig. 6 Histogram of remaining modules.

示すとおり、CC-MOA のモジュール除去率は最大で約 8 割と大きく、外れ値を除去しているというよりはむしろ、判別に有用なモジュールのみを残してそれ以外を削除していると考えられる。

7. おわりに

本論文では、fault-prone モジュール判別モデル構築の前処理として、モデル構築用のフィットデータに外れ値除去法を適用した効果を、交差検証法により実験的に比較した。実験には 4 つの外れ値除去法 (MOA, LOFM, RBM, CC-MOA) と、fault-prone モジュール判別モデルとして一般的に用いられる 3 つのモデル

(線形判別分析、ロジスティック回帰分析、分類木) を用いた。得られた主な結果は次のとおりである。

- LOFM を除く MOA, RBM, CC-MOA は、いずれのデータセット、及び、判別モデルにおいても、適合率が若干減少するものの、再現率が大きく向上し、これらの調和平均である F1 値も向上した。F1 値の平均向上幅を (MOA, RBM, CC-MOA) の順で並べると (0.139, 0.137, 0.123) であった。一般に、fault-prone モジュールの判別問題では、再現率よりも適合率の方が重要である¹³⁾ ことから、テスト効率の改善に有効であると考えられる。

- MOA, RBM, CC-MOA では, 判別モデル間の性能差を小さくする効果があった. いずれのデータセット, 判別モデルにおいても, MOA, RBM, CC-MOA により外れ値を除去した場合, F1 値は向上し, 0.4 付近に収束した. 外れ値除去によって, データセットごとに最適なモデルが異なるという問題が緩和され, モデル利用者にとって利便性が増すことが期待される.
- LOFM は, いずれのデータセットおよび判別モデルに対しても, ほとんど効果がなかった. LOFM は, fault-prone モジュール判別問題に不向きであることが示唆された.

今後の課題としては, より多くのデータセットを用いて同様の実験を繰り返すことで, 実験結果の一般性と信頼性を向上させていくことが重要となる.

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) Breiman, L., Friedman, J., Olshen, R. and Stone, C.: *Classification and Regression Trees*, Wadsworth, California (1984).
- 2) Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, J.: LOF: Identifying Density-based Local Outliers, *Proc. 19th ACM International Conference on Management of Data (SIGMOD'00)*, pp.93–104 (2000).
- 3) Conover, W. J.: *Practical Nonparametric Statistics*, Wiley, New York (1971).
- 4) Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems, *Annals Eugenics*, Vol.7, Part II, pp.179–188 (1936).
- 5) Gray, A. R. and MacDonell, S. G.: Software Metrics Data Analysis—Exploring the Relative Performance of Some Commonly Used Modeling Techniques, *Empirical Software Engineering*, Vol.4, No.4, pp.297–316 (1999).
- 6) Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems, *ACM Trans. Information Systems*, Vol. 22, No. 1, pp. 5–53 (2004).
- 7) Hosmer, D.W. and Lemeshow, S.: *Applied Logistic Regression*, Wiley, New York (1989).
- 8) Jimenez-Marquez, S. A., Lacroix, C. and Thibault, J.: Statistical Data Validation Methods for Large Cheese Plant Database, *Journal of Dairy Science*, Vol.85, No.9, pp.2081–2097 (2002).
- 9) Khoshgoftaar, T.M., Bullard, L.A. and Gao, K.: Detecting Outliers Using Rule-Based Modeling for Improving CBR-Based Software Quality Classification Models, *Proc. 5th International Conference on Case-Based Reasoning (ICCB'03)*, pp.216–230 (2003).
- 10) Knorr, E. M. and Ng, R. T.: Algorithms for Mining Distance-Based Outliers in Large Datasets, *Proc. 24th International Conference on Very Large Data Bases (VLDB'98)*, pp.392–403 (1998).
- 11) Matsumoto, S., Kamei, Y., Monden, A. and Matsumoto, K.: Comparison of Outlier Detection Methods in Fault-Proneness Models, *Proc. 1st International Symposium on Empirical Software Engineering and Measurement (ESEM'07)*, pp.461–463 (2007).
- 12) Mizuno, O., Ikami, S., Nakaichi, S. and Kikuno, T.: Fault-Prone Filtering: Detection of Fault-Prone Modules Using Spam Filtering Technique, *Proc. 1st International Symposium on Empirical Software Engineering and Measurement (ESEM'07)*, pp.374–383 (2007).
- 13) Munson, J. C. and Khoshgoftaar, T. M.: The Detection of Fault-Prone Programs, *IEEE Trans. Softw. Eng.*, Vol.18, No.5, pp.423–433 (1992).
- 14) Myers, G.J.: *The Art of Software Testing*, Wiley, New York (1979).
- 15) NASA/WVU: IV&V Facility, Metrics Data Program (online), available from (<http://mdp.ivv.nasa.gov/>) (accessed 2007-09-08).
- 16) Ostrand, T.J., Weyuker, E.J. and Bell, R.M.: Predicting the Location and Number of Faults in Large Software Systems, *IEEE Trans. Softw. Eng.*, Vol.31, No.4, pp.340–355 (2005).
- 17) Pighin, M. and Zamolo, R.: A Predictive Metric Based on Discriminant Statistical Analysis, *Proc. 19th International Conference on Software Engineering (ICSE'97)*, pp.262–270 (1997).
- 18) 柿元 健, 門田 暁人, 亀井 靖高, 杉本 真佑, 松本 健一: Fault-Prone モジュール判別における F1 値とソフトウェア信頼性の関係, ソフトウェア工学の基礎 XIV, 日本ソフトウェア科学会 (FOSE2007), pp.75–83 (2007).

(平成 18 年 11 月 28 日受付)

(平成 19 年 2 月 4 日採録)



松本 真佑 (学生会員)

平成 18 年京都産業大学理学部卒業。現在、奈良先端科学技術大学院大学情報科学研究科博士前期課程在籍。エンピリカルソフトウェア工学，特にソフトウェアメトリクスの研究

に従事。電子情報通信学会，IEEE 各会員



門田 暁人 (正会員)

平成 6 年名古屋大学工学部電気学科卒業。平成 10 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同年同大学同研究科助手。平成 16 年同大学助教授。平成

19 年同大学准教授。平成 15~16 年 Auckland 大学客員研究員。博士 (工学)。ソフトウェアメトリクス，ソフトウェアセキュリティ，ヒューマンファクタなどの研究に従事。



亀井 靖高 (学生会員)

平成 17 年関西大学総合情報学部卒業。平成 19 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。現在，同大学博士後期課程在籍。エンピリカルソフトウェア

工学，特にソフトウェア信頼性の研究に従事。電子情報通信学会，IEEE 各会員。



松本 健一 (正会員)

昭和 60 年大阪大学基礎工学部情報工学科卒業。平成元年同大学大学院博士課程中退。同年同大学基礎工学部情報工学科助手。平成 5 年奈良先端科学技術大学院大学助教授。平成

13 年同大学教授。工学博士。エンピリカルソフトウェア工学，特に，プロジェクトデータ収集 / 利用支援の研究に従事。電子情報通信学会，日本ソフトウェア科学会，ACM 各会員，IEEE Senior Member。