

## ソフトウェアバグデータへの拡張アソシエーションルールマイニングの適用 Applying Extended Association Analysis to Software Defect Data

齊藤 英和<sup>†</sup>  
Hidekazu Saito

門田 暁人<sup>†</sup>  
Akito Monden

### 1. はじめに

ソフトウェアテストにおいては、網羅的にバグを発見するために、規模あたりの実行テストケース数に基準を設けることが行われている（例えば、単体テストであれば100件/KSLOCなど[4]）。このような基準は、ソフトウェアの規模とバグ数が比例関係にあることを暗黙的に仮定しているが、現実には、規模とは独立にバグが混入する場合があります。そのような場合には異なるテスト戦略が必要となると考えられる。

本研究では、著者らが提案している、結論部に相関関数を持つ拡張アソシエーションルール[5]を用いた実験により、ソフトウェアの規模とバグ数の相関が大きくなる条件、および、小さくなる条件についてのルールの抽出を試みる。これにより、網羅的なテストが効果的となる場合とそうでない場合の切り分けを試みる。実験では、オープンソースソフトウェアの Mylyn を対象としてルール抽出を行い、得られたルールについて考察する。

### 2. アソシエーションルールマイニング

#### 2.1 アソシエーションルール

アソシエーションルールマイニングは、Agrawal らによって提案されたデータ分析手法である[1]。アソシエーションルールは  $A \Rightarrow B$  の形式で表現され、事象  $A$  が現れたとき、高い確率で  $B$  も現れることを表す。 $A$  を前提部（条件部）、 $B$  を結論部（帰結部）と呼ぶ。

アソシエーションルールを選定するための指標として、次の尺度が知られている。

$$\text{支持度}(A, B) = \frac{A \text{ と } B \text{ が同時に出現したケース数}}{\text{全ケース数}}$$

$$\text{信頼度}(A, B) = \frac{A \text{ の発生時に } B \text{ が出現したケース数}}{A \text{ が発生したケース数}}$$

支持度が大きいほどよく起こる事象を表す。信頼度は  $A$  と  $B$  の関連の強さを表す。一般に、支持度、および、信頼度に下限値を設けてルールの抽出・絞り込みが行われる。

#### 2.2 アソシエーションルールの拡張

アソシエーションルールの拡張として、筆者らは相関関数を扱うアソシエーションルールを提案している[5]。このアソシエーションルールは次のように表される。

$$A \Rightarrow \text{Correl}(X, Y)$$

$$A \Rightarrow \text{TestCorrel}(X, Y)$$

表1 変数とその説明

変数名	説明
FOUT	ファンアウトの多さ
VG	サイクロマティックの複雑さ
NSF	静的クラスの数
CBO	クラス間の結合度
RFC	クラスの応答の規模
DIT	継承ツリーの深さ
LCOM	凝集性欠如の度合い
WMC	クラス内のメソッド数
TPC	過去の変更回数
BFC	バグ修正が行われた回数
TLOC	ソースコード行数
BUG	バグの数

ここで、変数  $X$ ,  $Y$  はデータセット中の任意の量的変数を表す。 $\text{Correl}(X, Y)$  は2変数  $X$  と  $Y$  の相関係数を表す。この拡張により、任意の変数  $X$  と  $Y$  の相関が高く（もしくは低く）なるような前提部を発見することが可能となる。また、 $\text{TestCorrel}(X, Y)$  は  $X$  と  $Y$  の無相関検定（両側検定）の有意確率（ $p$  値）を計算する関数である。これにより、 $\text{Correl}(X, Y)$  で得られた相関が統計的に有意か否かを判断できるようになる。

### 3. 実験

本実験では、拡張アソシエーションルールをソフトウェアバグデータに適用し、ソフトウェアの規模とバグの相関が強くなる条件、および、弱くなる条件についてのルールを抽出する。ルールの抽出にはアソシエーションルールマイニングツールである NEEDLE[3]を拡張したものを使用する。

#### 3.1 データセット

データは Mylyn の version 2.0 におけるメトリクスとバグを計測したデータセット[2]を使用する。本データセットは、1230件のモジュールと、24個の変数から構成されている。本論文で示すアソシエーションルールに出現する変数とその説明を表1に示す。結論部の変数  $X$  は TLOC、変数  $Y$  は BUG である。

<sup>†</sup> 岡山大学大学院自然科学研究科, Graduate School of Natural Science and Technology, Okayama University

表2 抽出されたルールのうち、相関係数が大きいもの上位5件

番号	ルール	支持度	P 値
1	(LCOM=低)&(BFC=高) $\Rightarrow$ Correl(BUG,TLOC)=0.688	0.324	3.82E-57
2	(LCOM=低) $\Rightarrow$ Correl(BUG,TLOC)=0.649	0.547	1.09E-81
3	(RFC=高)&(BFC=高) $\Rightarrow$ Correl(BUG,TLOC)=0.647	0.324	1.26E-48
4	(DIT=高)&(LCOM=低) $\Rightarrow$ Correl(BUG,TLOC)=0.647	0.359	1.42E-53
5	(DIT=高)&(TPC=高) $\Rightarrow$ Correl(BUG,TLOC)=0.634	0.369	1.66E-52

表3 抽出されたルールのうち、相関係数が小さいもの下位5件

番号	ルール	支持度	P 値
1	(TPC=低) $\Rightarrow$ Correl(BUG,TLOC)=0.0985	0.472	3.51E-02
2	(VG=低)&(CBO=低) $\Rightarrow$ Correl(BUG,TLOC)=0.114	0.337	4.01E-02
3	(FOUT=低)&(WMC=低) $\Rightarrow$ Correl(BUG,TLOC)=0.119	0.402	1.66E-02
4	(LCOM=高) $\Rightarrow$ Correl(BUG,TLOC)=0.122	0.453	7.91E-03
5	(FOUT=低)&(NSF=低) $\Rightarrow$ Correl(BUG,TLOC)=0.123	0.308	3.32E-02

### 3.2 実験手順

[手順 1] データの離散化：結論部に指定しない量的変数について、カテゴリごとのデータの数が同数に近くなるように大・小の2カテゴリに分割する。

[手順 2] データのクリーニング：大小のカテゴリ間でデータ数に大きな偏りがある（1つのカテゴリが全データの75%以上を占めている場合）、その変数を予め削除する。

[手順 3] ルールの絞り込み：ルールの最低支持度を0.3、前提部における変数結合数の上限を1に設定してルールの抽出を行う。

### 3.3 結果と考察

抽出されたルールのうち、相関係数が大きいもの上位5件を表2に、下位5件を表3に示す。なお、無相関検定によって算出されるp値が0.05を上回るものは除外している。また、データセット全体において、規模とバグの量の相関係数は0.416であった。

表2のうち、最も多く出現する条件は“LCOM=低”である。これは凝集性欠如の度合いが低いことを表しており、一般に良い設計とされている。このことから、凝集性の観点で良い設計であるほど規模に応じてバグが混入する傾向にあり、逆に、悪い設計の場合は規模以外の要因でバグ混入する傾向があると考えられる。

また、この条件に“BFC=高”が加わったものが相関係数の1番大きいルールである。BFCはバグ修正が行われた回数を示している。“BFC=高”のみを条件に持つルールの相関係数は0.622と高い値であった。

以上より、“LCOM=低”と“BFC=高”はともにTLOCとBUGの相関が大きくなる要因と言え、また、両方を満たした場合、その相関はより大きくなる。よってそのようなモジュールは規模に応じてバグが増える可能性が高いため、網羅的なテストが効果的であると言える。

次に、表3において、最も相関係数が低いルールの条件は“TPC=低”であり、これは過去の変更回数が少ないことを表しており、そのようなモジュールはそもそもバグが少ないので、相関が低くなったと考えられる。

また、“VG=低” & “CBO=低”が2番目に相関係数が小さいルールになっている。これらの指標はそれぞれ、マイクロマティックの複雑さとクラス間の結合度を表しており、これらの条件に合致するモジュールの多くは規模の小さいものであり、かつ、複雑さが低いものである。そのようなモジュールはそもそもバグが少ないことから、規模とバグの相関が小さくなったと考えられる。実際に規模(TLOC)の平均を比べると“VG=低”かつ“CBO=低”の場合は45.97、それ以外の場合は212.25であった。

### 4. まとめ

本論文では、拡張アソシエーションルールを用いた実験により、ソフトウェアの規模とバグの量の相関が大きくなる条件、および、小さくなる条件についてのルール抽出を試みた。これにより、網羅的なテストが効果的となる条件とそうでない条件を得ることができた。

今後は、より多くのデータセットを用いた分析を行い、結果の一般性についても調査したい。

### 謝辞

本研究の一部は、科学研究費補助金(17K00102)の助成を受けたものである。

### 参考文献

- [1] R. Agrawal, T. Imielinski, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” Proc. International Conference on Management of Data, pp.207-216 (1993).
- [2] A. Monden, J. Keung, S. Morisaki, Y. Kamei, K. Matsumoto, “A heuristic rule reduction approach to software fault-proneness prediction,” Proc. Asia-Pacific Software Engineering Conference (APSEC 2012) pp.838-847 (2012)
- [3] S. Morisaki, A. Monden, H. Tamada, T. Matsumura, and K. Matsumoto, “Mining quantitative rules in a software project data set,” IPSJ Journal, Vol.48, No.8, pp.2725-2734 (2007).
- [4] 奈良隆正, “ソフトウェア品質保証, 評価技術の勘所,” JaSST’09 Kansai (2009).
- [5] 齊藤 英和, 門田 暁人, Zeynep Yücel, 森崎 修司, “相関関数を扱うアソシエーションルールの提案とソフトウェア開発データへの適用,” コンピュータソフトウェア, Vol. 36, No. 3, 2019.