コードクローンメトリクスの差異が fault-prone モジュール判別に与える影響

角田雅照[†1]　梶村和輝[†1]　亀井靖高[†2]　沢田篤史[†3]

fault-prone モジュール判別モデルの説明変数にコードクローンメトリクスを用いることにより，予測精度が高まることが指摘されている．ただし，クローン検出方法の違いにより，コードクローンメトリクスに差異が生じる．本稿では，この差異がモデルの予測精度に与える影響に着目する．

Influence of difference of code clone metrics on fault-prone module prediction

Masateru Tsunoda[†1]　Kazuki Kajimura[†1]　Yasutaka Kamei[†2]　Atsushi Sawada[†3]

Past researches showed that using code clone metrics as independent variables enhances accuracy of a fault-prone module prediction model. Different code clone detection methods produces different code clone metrics. We focus on influence of the difference on accuracy of the model.

1. はじめに

近年，ソフトウェア開発は大規模化，短納期化しており，ソフトウェアの品質を高める重要性がますます高まっている．そのような開発において，ソフトウェアの品質を高めるためには定量的なプロジェクト管理が不可欠であり，そのために，ソフトウェアのテスト実施前に fault-prone モジュールを判別することが行われる．

fault-prone モジュール判別とは，あるモジュールに fault が含まれているかどうかを，テスト前にあらかじめ予測（判別）することであり，予測結果に基づいてテスト計画が立案される．判別モデルはソースコードのメトリクスと，過去に発見された fault のデータに基づいて構築される．構築されたモデルに判別対象のモジュールのメトリクスを入力することにより，対象に fault を含んでいるかどうかを予測する．判別モデルの説明変数として，fault と関連を持つと考えられる，モジュールの複雑度やコード行数を用いることが一般的である．

いくつかの研究では[1), 2)]，モデルの説明変数としてコードクローンメトリクスを用いることにより，判別予測の精度が向上することを示している．コードクローンとは，モジュール内，またはモジュール間における類似したコード片のことを指す．コードクローンの箇所は，一般に修正時に全てを更新する必要性があることが多く，一部に修正漏れが発生する可能性がある．すなわち，コードクローンが多いほど保守性が低下し，修正時に fault を混入する危険性が高まる．

コードクローンメトリクスとは，各モジュールおけるコードクローンの含有率などを指す．コードクローンメトリクスはモジュールの保守性を示す特性の一つであり，fault と関連を持つと考えられる．そのため，コードクローンメトリクスを説明変数として用いることにより，モデルの精度が向上すると考えられる．

2. クローンメトリクスの差異

2.1. コードクローン検出手法

従来研究[1), 2)]では，コードクローンメトリクスの算出に Kamiya ら[4)]の提案したアルゴリズムに基づくツールを用いている．Kamiya らの手法では，コードクローン検出時にソースコードをトークンに置き換え，トークンが一定長以上連続して一致している場合，コードクローンとみなして検出している．Kamiya らの手法の特長は非常にスケーラブルなことであり，大規模なソフトウェアに適用してコードクローンを検出することが比較的容易である．

Kamiya らの手法以外にも，コードクローン検出方法がいつくか提案されている．コードクローンの検出手法は，主に**文字/行の並び**，**トークンの並び**，AST（Abstract Syntax Tree），**依存関係**に分類される[3)]．

文字/行の並びに基づく手法では，ソースコードをそ

†1 近畿大学
Kinki University
†2 九州大学
Kyushu University
†3 南山大学
Nanzan University

のまま文字列として扱う. Kamiya らの手法はトークンの並びに基づく手法の1つである. ASTに基づく手法は, ソースコードを AST に変換し, 構文木を比較することにより類似かどうかを判定する方法である. 依存関係に基づく手法では, ソースコードから PDG (Program Dependence Graph)を作成し, グラフを比較することにより類似かどうかを判定する方法である.

コードクローン検出方法により, コードクローンの定義が異なり, 検出されるコードクローンが異なる. 従って, 同一のソースコードを計測対象とした場合でも, 検出方法の違いにより, コードクローンメトリクスに差異が生じる. 例えば, 文字/行の並びに基づく手法では, 2 つのコード片において変数名が異なる場合, コードクローンとみなされないが, その他の手法ではコードクローンとみなされる. また, トークンの並びに基づく手法では, 2 つのコード片の中身は同じでも順序が異なる場合, コードクローンと見なされないが, 依存関係に基づく手法の場合, コードクローンとみなされる.

2.2. Research questions

本稿では, コードクローン検出手法の違いから生じる, コードクローンメトリクスの差異に着目する. 我々は, fault-prone モジュール判別モデルの構築, 利用時に, この差異が影響するのかどうかを明らかにする必要があると考える. もし影響を与えない場合, スケーラブルなコードクローン検出方法を用いてコードクローンメトリクスを算出し, モデルを構築すればよいことになる. 逆に影響を与える場合, モデルの予測精度とスケーラビリティとのバランスを考慮してコードクローン検出方法を選択する必要が生じる. よって本稿では, 以下の research questions に答える必要があると考える.

- RQ1: コードクローン検出／メトリクス算出手法の違いにより, fault-prone モジュール判別モデルの精度に違いが生じるのか?
- RQ2: (RQ1 に対する答えが yes の場合)どのコードクローン検出／メトリクス算出手法を用いると, 最も fault-prone モジュール判別モデルの精度が高まるのか?
- RQ3: 複数のコードクローン検出／メトリクス算出手法を適用して, それぞれから得られたメトリクスを同時に用いた場合, fault-prone モジュール判別モデルの精度は高まるのか?

上記の research questions に答えるためは, 以下の3種類の fault-prone モジュール判別モデルを構築し, それぞれの精度を比較する必要がある.

- モデル 1: コードクローンメトリクスを用いないモデル
- モデル 2: 単一のコードクローン検出手法に基づくメトリクスを用いたモデル
- モデル 3: 2 つ以上のコードクローン検出手法に基づくメトリクスを用いたモデル

モデル 1 は, モデル 2, 3 と比較し, コードクローンメトリクスが fault-prone モジュール判別モデルの予測精度向上に寄与することを再確認(追実験)するためのモデルである. モデル 2 は RQ2 に答えるため, モデル 3 は RQ3 に答えるためのモデルである.

実験では各モデルの予測精度の比較だけではなく, 詳細な分析が必要となる. 例えば, あるコードクローン検出手法を用いてモデルの予測精度が高まった場合, その手法が特定したコードクローンに, 実際にバグが含まれていることが多いのかなどを確かめる必要がある.

3. おわりに

本稿では, コードクローンメトリクスの差異が, fault-prone モジュール判別予測モデルの精度に与える影響を分析する必要性について述べた. ワークショップでは, コードクローンメトリクスの開発現場での活用方法と, 実験結果を補強するために必要な分析について議論したい.

謝辞 本研究の一部は, 文部科学省科学研究補助費(基盤 C:課題番号 25330090, 24500049), 2013 年度南山大学パッヘ奨励金 I-A-2 による助成を受けた.

参 考 文 献

1) 馬場慎太郎, 吉田則裕, 楠本真二, 井上克郎: Fault-Proneモジュール予測へのコードクローン情報の適用, 電子情報通信学会論文誌D, J91-D, no. 10, pp. 2559-2561 (2008).
2) 亀井靖高, 左藤裕紀, 門田暁人, 川口真司, 上野秀剛, 名倉正剛, 松本健一:クローンメトリクスを用いた fault-prone モジュール判別の追実験, 電子情報通信学会論文誌 D, J93-D, no. 4, pp. 544-547 (2010).
3) 神谷年洋, 肥後芳樹, 吉田則裕:コードクローン検出技術の展開, コンピュータソフトウェア, vol. 28, no. 3, pp. 29-42 (2011).
4) Kamiya, T., Kusumoto, S. and Inoue, K.: CCFinder: a multilinguistic token-based code clone detection system for large scale source code, *IEEE Trans. on Software Eng.*, vol. 28, no. 7, pp. 654-670 (2002).