2M-3

学習に基づく error-prone モジュール予測器選択 -メトリクス種別の考察-

嶋 大輔　　小形 真平　　海谷 治彦　　海尻 賢二
信州大学大学院理工学系研究科情報工学専攻

1. はじめに

Error-prone モジュールとは、バグの発生する可能性の高いモジュールのことである。Error-prone モジュールを的確に予測し、それを重点的に検査することで、ソフトウェアの保守・点検の作業効率が向上する。メトリクスを用いた Error-prone モジュールの予測手法としては、図１に示すように過去のプロジェクトを訓練データとし、アルゴリズムに学習させることで予測器を作成、出来た予測器を用いて予測対象プロジェクトのバグを予測する手法が採られている。しかし、過去の研究から、予測対象プロジェクトによって予測精度の高い訓練データ・アルゴリズムの組は異なっており、一律に最適な予測器は存在しないことがわかっている。そこで、Zhimin の研究[1]では、最適な予測器を識別するための識別器を作り、識別器によって選択された予測器を利用する手法が提案されている。
本研究では、Zhimin の手法を適用するにあたって、２つのリサーチクエスチョンを提案する。

RQ1：検査対象のプロジェクトに対する最適な予測器（訓練データに使用するプロジェクト＋メトリクス種別＋アルゴリズム）は、検査プロジェクトによって異なる。

RQ2：メトリクス種別の混合は、予測性能の向上に効果がある

この 2 つの RQ に答えることによって、予測器のマイニングの必要性、および、予測における異なった種別のメトリクスの組み合わせの効果を探る。

2. データセット

本研究では、Marco D'Ambros の論文[2]で公開されている 5 種類のプロジェクトを対象とする（以降、このデータセットを Ambros データセットと呼ぶ）。

プロジェクト	クラス数	バグ数	バグ率
Eclipse	997	207	0.208
Equinox	324	129	0.398
Lucene	691	64	0.0926
Mylyn	1863	246	0.132
Pde	1498	209	0.14

表 1.　使用プロジェクトとバグについての詳細

使用メトリクス種別は、Change metrics (from CVS change logs) as proposed by Moser, Previous defects as proposed by Kim, Source code metrics as proposed by Basili, Entropy of changes as proposed by Hassan, Churn of source code metrics as proposed by D'Ambros, Entropy of source code metrics as proposed by D'Ambro の６種類を用いる。学習アルゴリズムは、BayesNet, IB1,MultiClassClassifier, MultilayerPerceptron, J48 の 5 種類を使用し、予測器の作成・予測には Weka を用いる。各プロジェクトの諸量を表１に示す。

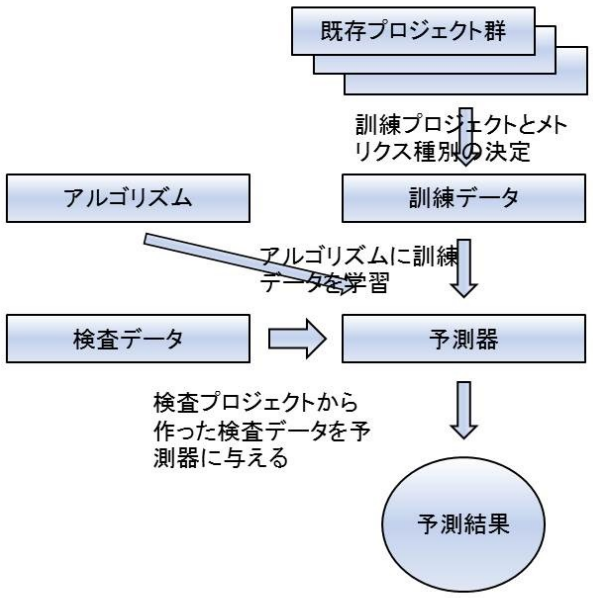

図 1.　メトリクスを用いた予測の流れ

3. 実験方法

RQ1 のための実験方法

Ambros データセットの各プロジェクトデータを

Selection of error prone module predictor - consideration of the kind of metrics-
Daisuke Shima, Shinpei Ogata, Haruhiko Kaiya, Kenji Kaijiri (Faculty of Engineering, Shinshu University)

訓練、検査データとし、[訓練データ-アルゴリズム-検査データ-使用メトリクス種別]のすべての組み合わせで予測実験を行い、予測結果を収集する。収集した予測結果を分析し、検査対象プロジェクトごとに予測精度の高い予測器の調査を行う。各検査対象プロジェクトに対する[訓練データ-アルゴリズム-使用メトリクス種別]の組み合わせ（予測器）の数は、訓練プロジェクト 4 種×アルゴリズム 5 種×メトリクス種別 6 種の計 120 個になる。

RQ2 のための実験方法

Ambros データセットは6種類のメトリクス種別から成る。そこで上記実験に、使用するメトリクス種別の混合を追加する。メトリクス種別を混合して予測した場合の予測結果と、メトリクス種別を単一で予測した場合の予測結果を比較する。

なお、本研究では、Recall・Precision の両値を重要視したいため、２つの調和平均である F-measure を予測精度として使用する。

4. 実験結果

RQ1 のための実験

各検査対象プロジェクトに対する予測結果の F-measure の値を図２に箱ひげ図にして示す。

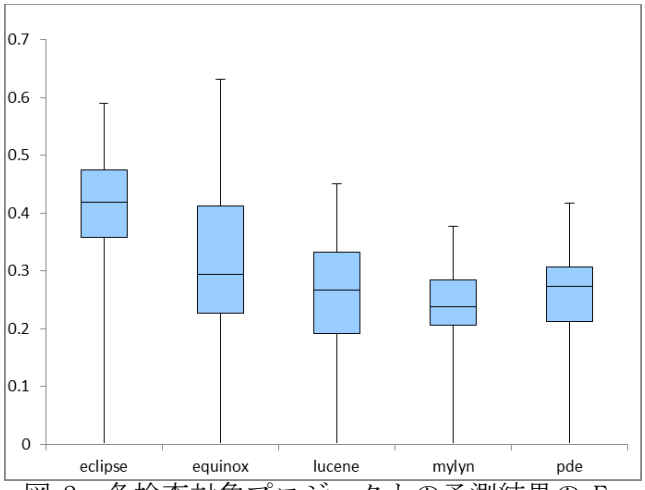

図 2. 各検査対象プロジェクトの予測結果の F-measure の箱ひげ図

図２から、各検査対象プロジェクトにおいて、予測器の精度には幅があるため、実際に予測を行う際には、プロジェクトごとに最適な予測器を選択する必要があることがわかる。

次に、検査プロジェクトごとに F-measure が上位 5 つの予測器を調べてみた結果、複数の検査プロジェクトで、同一の[訓練データ-アルゴリズム-使用メトリクス種別]の組がいくつか存在した。これらの組み合わせが上位 5 つに存在しなかった検査プロジェクトでも、これらの予測器は平均以上の精度を示していた。このことから、今回の対象データセットにおいては、どの検査対象プロジェクトに対しても最適な予測器は存在するという結果となった。

RQ2 のための実験

全体的に予測精度が高い結果を示した訓練データが eclipse、アルゴリズムが BayesNet を用いた予測器に絞って、メトリクス種別の混合を行った際の予測精度について調査を行った。メトリクス種別が、Change metrics, Previous defects 単体での予測結果は全体的に F-measure が低い結果となった。逆に Entropy of source code metrics, Churn of source code metrics のメトリクス種別は、単体でも F-measure が高い結果を示した。また、各検査プロジェクトに対する F-measure の値が高い予測器を確認してみたところ、使用されているメトリクス種別に傾向があった。しかし、メトリクス種別を単体で予測したものと、混合して予測したものに、実用上の差は見られなかった。これらのことから、メトリクス種別の混合は必ずしも予測精度の向上に効果があるとは言えない結果となった。

5. 終わりに

本論文では、Error-prone モジュール予測に関する 2 つのリサーチクエスチョンについて Ambros データセットを使って検証した。RQ1 については、既存研究と異なる結果になった。RQ2 についても、異なるメトリクス種別の混合が予測精度の向上に効果があるとは言えなかった。しかし、これらは、あくまで予測精度として F-measure を用いた場合の結果である。F-measure は、Recall, Precision の平均であるため、どちらかの値を重要視するかによって、全く別の結果になる可能性がある。また、どちらの結果も、5 種類のプロジェクト、6 種別のメトリクスのデータセットのみを対象とした実験結果であり、一般性を考えればさらに対象範囲を広げた実験を行う必要がある。

なお、実験結果はページ数の関係で以下の WEB ページにアップしているので参照されたい。

http://cwww.cs.shinshu-u.ac.jp/~shima/

参考文献

[1]Zhimin He, et al: An investigation on the feasibility of cross-project defect prediction, ASE 19, 2 2012.

[2]Marco D'Ambros, et a: Evaluating defect prediction approaches: a benchmark and an extensive comparison, Emp. Soft. Eng, 17,4-5 2012