

# 学習データの時間的変化に伴う 欠陥モジュール予測モデルの評価

内垣 聖史<sup>1,a)</sup> 伊原 彰紀<sup>1,b)</sup> 門田 暁人<sup>1,c)</sup> 松本 健一<sup>1,d)</sup>

**概要:** ソフトウェア開発におけるテスト工程の効率化を目的とした、欠陥が含まれている可能性の高いモジュールを予測する(欠陥モジュール予測)研究では、一般的に過去のバージョン開発終了時の学習データを用いてモデルが構築される。しかし昨今、アジャイル開発を導入するケースが増加し、開発途中であっても学習データの計測が可能となった。本論文では、学習データ計測時期による予測精度の違いを分析する。結果、予測時点に近い開発データを用いるほど、モデルの予測精度が高いことが分かった。

## 1. はじめに

ソフトウェア開発プロセスのテスト工程において、欠陥を含みやすいソースコード(以下、欠陥モジュール)を特定することはテストの効率化のために重要であり、近年盛んに研究がおこなわれている [1][2]。従来、ウォーターフォール型の開発プロジェクトを対象とした欠陥モジュール予測では、過去バージョンの開発終了時のスナップショット(ある時点のソフトウェアを構成するソースコードの集合)を用いて予測モデルが構築されることが多かった。一方で、短いサイクルで機能を拡張していくアジャイル開発では、開発途中であっても欠陥モジュール予測に必要なソフトウェアメトリクス(コード行数、分岐数)を取得し、予測モデルを構築することができる。

本論文では、開発途中のスナップショットを用いて欠陥モジュール予測モデルを構築し、学習データの取得時期による予測精度の違いを分析する。

## 2. 欠陥モジュール予測

欠陥モジュール予測モデルは、過去に開発されたプロジェクトのモジュールに関するソフトウェアメトリクスを説明変数とし、モジュールに含まれる欠陥情報(欠陥の有無や欠陥密度(単位行数あたりの欠陥数))を目的変数として構築される。構築した予測モデルに対して新規に開発したプロジェクトのモジュールから計測したソフトウェアメトリクスの値を入力することで、モジュールに欠陥が含

まれる確率や欠陥密度を予測する事ができる。しかしながら、モジュールは頻繁に変更され、それに伴い欠陥が含まれる可能性の高いモジュールも変化すると考えられる。

## 3. 実験

本論文では、Eclipse プロジェクトの 2005 年 7 月から 2006 年 7 月までの 13 か月分(各月 1 日)のスナップショットを用いて、それぞれの時期のデータを学習データとして欠陥モジュール予測モデルを構築し、学習データの違いによる予測精度を比較する。モデルを構築するために、スナップショットのソースコード群から計測したコード行数やクラス数など 13 種のメトリクスを計測する。実験の概略図を図 1 に示す。本論文ではモデル構築アルゴリズムとして、ランダムフォレスト [3] を採用する。予測モデルの評価指標には Mende[4] らの提案した  $P_{opt}$  を用いる。 $P_{opt}$  の値域は [0,1] であり、値が 1 に近づくほど予測精度が高い事を示す。本実験は以下に示す 3 つの手順で行った。

### 手順 1: 予測モデルの構築

スナップショットの内、取得時期が最も遅いスナップショット(2006 年 7 月時)を評価データとし、その他のスナップショットを学習データとしてそれぞれランダムフォレスト法を用いて予測モデルを構築する。

### 手順 2: 欠陥密度の予測

構築した 12 個の予測モデルに評価データのメトリクス値を適用し、各モジュールの欠陥密度を予測する。

### 手順 3: $P_{opt}$ の算出

各モデルから得られた欠陥密度の予測値と評価データから計測した欠陥密度の実測値から  $P_{opt}$  をそれぞれ算出する。

<sup>1</sup> 奈良先端科学技術大学院大学, 情報科学研究科

a) satoshi-u@is.naist.jp

b) akinori-i@is.naist.jp

c) akito-m@is.naist.jp

d) matumoto@is.naist.jp

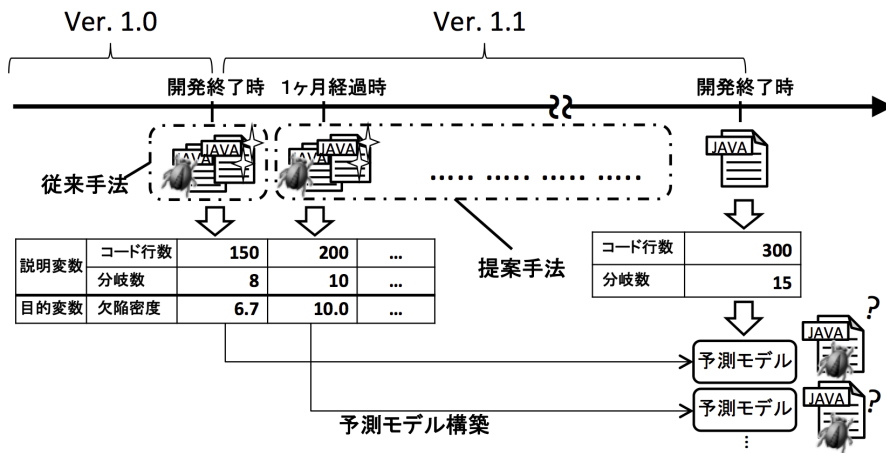


図 1 実験の概略図

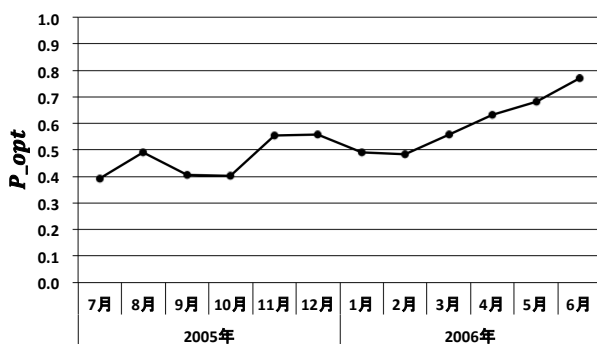


図 2  $P_{opt}$  の推移 (評価データ：2006年7月)

#### 4. 実験結果

学習データ計測時期による予測精度を図2に示す。評価データは2006年7月とする。横軸は予測モデルを構築したスナップショットの取得対象時期、縦軸は  $P_{opt}$  の値を示す。図2より、取得時期が最も早いスナップショット(2005年7月時)から構築されたモデルを利用した場合に最も予測精度が低くなり、予測値(欠陥密度)の高い上位50%のソースコードから発見された欠陥の67%を検出した。一方で、最も遅いスナップショットの場合に最も予測精度が高く、同様に予測値(欠陥密度)の高い上位50%のソースコードから発見された欠陥の83%の欠陥を検出できた。予測精度の時系列推移を最小二乗法を用いて傾向推定した結果、スナップショットの取得対象時期が遅くなるほど、予測精度が上昇する傾向にあった(有意水準1%)。従って、開発終了時に近い時期のソースコードを用いて学習データを作成することで、欠陥が混入しているソースコードにテスト工数を効率的に与えられと考えられる。

#### 5. 考察

開発終了時により近い時期のソースコードを学習データとして構築した予測モデルの精度が高くなった理由は、時

間経過に伴い徐々にソースコードが変更され、評価データが予測モデルに適合するようになったことが考えられる。各スナップショットにおいて全ソースコードを対象にメトリクスの値域を計測すると、13個中9個のメトリクスの値域は開発終了時に近い時期よりも遠い時期の値域の方が狭いことが分かった。従って、開発終了時期から遠い時期の学習データでは、評価データの一部しか予測することができない。この結果は、時間の経過と共にソースコードが徐々に変更され、予測モデルに評価データが適合しにくくなるという筆者らの考えを支持するものである。

#### 6. おわりに

本論文では、学習データの取得時期による欠陥モジュール予測精度の違いを分析した。その結果、学習データ及び評価データとして利用する各スナップショットの取得間隔が短いほど高い予測精度が得られる事が分かった。今後はメトリクス値の変動が大きいモジュールのみを対象とした予測実験を行い、更に本実験で得られた知見の妥当性を向上させるために他のプロジェクトデータを用いた実験を行う。

#### 参考文献

- [1] O. Mizuno, and H. Hata, " Prediction of fault-prone modules using a text filtering based metric," International Journal of Software Engineering and Its Application, Vol.4, No.1, pp.43-52, 2010.
- [2] T. Gyimothy, R. Ferenc, and I. Siket, " Empirical validation of object-oriented metrics on open source software for fault prediction," IEEE Transactions on Software Engineering, Vol.31, No.10, pp.897-910, 2005.
- [3] L. Breiman, "Random forests," Machine Learning, Vol.45, No.1, pp. 5-32, 2001.
- [4] T. Mende, and R. Koschke, "Revisiting the evaluation of defect prediction models," Proceedings of International Conference on Predictor Models in Software Engineering (PROMISE'09), pp.1-10, 2009.