一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SS2018-4,KBSE2018-14（2018-07）

開発者の活動量の経時的変化が コミッター候補者予測に与える影響の分析

山崎　大輝†　大平　雅雄††　伊原　彰紀††　柏　祐太郎†,††† 宮崎　智己†

† 和歌山大学大学院システム工学研究科　〒640–8510 和歌山県和歌山市栄谷 930
†† 和歌山大学システム工学部　〒640–8510 和歌山県和歌山市栄谷 930
††† 日本学術振興会 特別研究員（DC）
E-mail: †{yamasaki.daiki,kashiwa.yutaro,miyazaki.tomoki}@g.wakayama-u.jp,
††{masao,ihara}@sys.wakayama-u.ac.jp

あらまし　多くの大規模 OSS プロジェクトでは，変更したコードを検証し版管理システムに反映するための特別な権限を持つコミッターの不足が問題になっている．本研究の目的は，既存研究が着目してこなかったコミッター候補者予測モデルの予測精度とデータサイズとの関係を明らかにすることである．本稿では，Eclipse Platform プロジェクトを対象にデータサイズが予測精度にどのような影響を与えるのかを調べるための実験を行った．実験の結果，取得可能な全期間のデータを用いるよりも直近の期間のデータで予測を行った方が予測精度が高くなることが分かった．
キーワード　OSS，コミッター候補者予測，ロジスティック回帰

An Analysis of the Impact of Temporal Changes of Developers' Activities on the Committer Candidate Prediction Model

Daiki YAMASAKI†, Masao OHIRA††, Akinori IHARA††, Yutaro KASHIWA†,†††, and Tomoki MIYAZAKI†

† Graduate School of Systems Engineering, Wakayama University　930 Sakaedani, Wakayama-city, Wakayama, 640–8510 Japan
†† Faculty of Systems Engineering, Wakayama University　930 Sakaedani, Wakayama-city, Wakayama, 640–8510 Japan
††† Japan Society for the Promotion of Science
E-mail: †{yamasaki.daiki,kashiwa.yutaro,miyazaki.tomoki}@g.wakayama-u.jp,
††{masao,ihara}@sys.wakayama-u.ac.jp

Abstract　Many of large-scale open source projects have a serious problem in recruiting committers who have a privilege to commit changed and verified code to a version management system. The goal of this study is to reveal the relationship between the accuracy of the committer candidate prediction model and the size of data, which were not studied in the past. In this paper, we conduct an experiment to investigate the relationship using data extracted from the Eclipse Platform project. As a result, we found that the accuracy of the prediction model became better when using data during a recent one or two year period than when using all the available data.
Key words　OSS, Committer Candidate Prediction, Logistic Regression

1. はじめに

近年，OSS (Open Source Software) の普及とともに OSS プロジェクトへの不具合報告が増加している [1]．不具合報告に対処するため，パッチと呼ばれる修正前後のソースコードの差分を表すファイルが多くの開発者からプロジェクトに投稿される．投稿されたパッチはコミッターと呼ばれる一部の開発者の手によって検証され，版管理システムに反映（コミットと呼ぶ）される．コミット権限を有するコミッターは一般に少数であり，パッチが多数投稿される状況ではコミッター 1 人当たりの負担

が集中する．結果的に，大規模プロジェクトではパッチの検証作業が滞る，すなわち，不具合の修正がプロダクトに反映されるまでに時間がかかる傾向にある [2]．

既存研究では [3]，開発者の過去の活動量に着目したコミッター候補者予測モデルが提案されている．コミッターの増員を支援することでコミッター 1 人当たりの負荷を軽減することを狙いとしている．[3] では，データセットに含まれる開発者の活動量メトリクス（パッチ投稿数やコメント数）をすべて計測し，学習データとテストデータとに 2 分割した上で，予測モデルの構築および評価を行っている．しかしながら，大規模プロジェクトは長期間にわたり継続しているものが多く，プロダクトの成熟度やプロセス改善によりコミッターの昇格基準が変わっている可能性がある．そのため，計測した全期間の活動量メトリクスを用いて構築した予測モデルは直近のコミッター昇格基準を上手く反映できない可能性がある．言い換えると，直近のデータのみを用いる方が直近のコミッター昇格基準を反映したより精度の高い予測モデルを構築できる可能性がある．

そこで本研究では，コミッター候補者予測のさらなる精度向上を目的として，計測対象となるメトリクスの経時的変化に着目し，データセットのサイズ変更が予測精度へ与える影響を分析する．特に本稿では，長期継続かつ大規模プロジェクトである Eclipse ロジェクトを対象として行った予測精度評価実験について報告する．本稿の構成は次の通りである．続く 2 章では，OSS 開発におけるコミッターの役割と既存研究について述べ，本研究の動機を説明する．3 章では，本研究におけるリサーチクエスチョンを示し，4 章においてリサーチクエスションを明らかにするために行った評価実験の結果を示す．5 章では，評価実験により得られた結果に基づいて考察を行う．最後に 6 章において本稿のまとめと今後の研究の課題について述べる．

2. 研究の動機

2.1 OSS 開発におけるコミッターの役割

OSS 開発では，不具合修正あるいは機能追加・改良を行うために，開発者は元のソースコードに対する変更差分，すなわち，パッチをプロジェクトに投稿する必要がある．また，投稿されたパッチがプロダクトに反映されるためには，版管理システムへのコミット権限を有するコミッターの検証作業を経る必要がある．コミッターによる検証はプロダクトの品質を保証する上で非常に重要な作業である．しかしながら一般に，大規模プロジェクトであってもコミッターは少数であるため，不具合修正等を目的として多数のパッチが投稿される状況においては，コミッター 1 人当たりの作業負担が集中する傾向にある [4]．そのため大規模プロジェクトでは，パッチが反映されるまでに多くの時間が必要となることが多く，不具合修正時間が長期化したり [2]，パッチ投稿等によりプロジェクトに貢献する一般開発者の離脱を招く要因となっている [5], [6]．

2.2 既 存 研 究

コミッター不足を解消することを目的とする研究アプローチには大きく二通りの方法がある．1 つは，有用な開発者がプロジェクトを離脱する前にできるだけ多くコミッター候補者として推薦する方法である．もう一方は，得る有能な開発者がプロジェクトを離脱しないようにするための環境や要因を明らかにする方法である．

前者の研究は，プロジェクトに参加する多数の一般開発者の中からコミッターに適した開発者を分類する問題として定式化することでコミッター候補者予測モデルを構築するものである．数多く存在する予測モデルの中で最もシンプルなものとして，伊原らのコミッター候補者予測モデルがある [3]．コミッターの昇格基準として開発者の活動量（活動期間やパッチ投稿数，コメント投稿数）に着目することで，コミッター候補者と一般開発者を弁別するモデルを構築している．一方，後者の研究は，将来コミッターに昇格するような有能な開発者がプロジェクトを離脱しないよう過去の長期貢献者の活動量やプロジェクトの環境を分析 [7] したり，早期にコミッター候補者を特定するための分析 [8] をするものである．

しかしながら，これら既存研究では，長期的に継続している大規模プロジェクトから取得した過去の全データを用いて予測モデルの構築や分析が行われているため，実際のプロジェクト内での運用においては直近のトレンドに即したコミッター昇格基準とは異なる可能性がある．次節では Motivating Example として具体例を示しつつ，本研究の動機を説明する．

2.3 Motivating Example

図 1 (a) (b) はそれぞれ，Eclipse プロジェクトにおける 2002 年から 2014 年までにコミッター昇格前の開発者が行ったパッチ投稿およびコメント投稿（不具合管理システムで行った議論の数）の総数を 1 年ごとに示したものである．年によりそれぞれの投稿数にばらつきが存在すること，プロジェクトの前半と比べると後半は投稿数が減少していることなどが見て取れる．大規模な機能拡張が行われる時期にはパッチやコメント投稿が多い開発者がコミッターとして昇格する可能性が高く，プロダクトが成熟してきたと思われる 2009 年以降はそれ以前に比べるとコミッターに昇格するための基準として多くの投稿数は必ずしも必要はないことを示唆している．

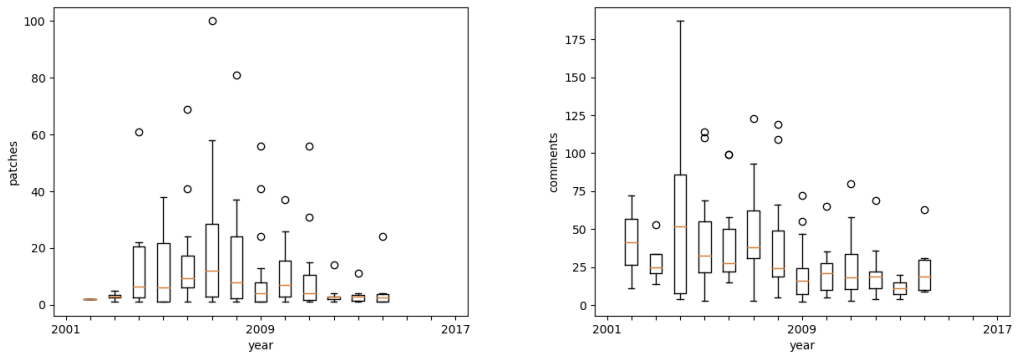
このように，プロジェクト内でコミッター昇格の基準は，投稿数の絶対評価ではなく相対評価によるものに変化している可能性が高く，既存研究のように全データを用いて予測モデルを構築したり分析を行った場合，プロジェクト内で暗黙的に運用されているコミッター昇格基準とは異なるものになる可能性が高い．そこで本研究では，年月の経過によるデータの変化を考慮してコミッター候補者予測モデルの構築をすることで，予測に必要なデータを削減しつつも予測精度の向上が図れるのではないかと考えた．

3. リサーチクエスチョン

本章では，既存研究のコミッター候補者予測モデルについて説明するとともに，本研究で取り組むリサーチクエスチョンについて説明する．

3.1 既存研究のコミッター候補者予測モデル

コミッター候補者予測モデルの研究は多数存在するが，本稿では最もシンプルなものとして伊原らのコミッター候補者予測



(a) パッチ投稿数　　(b) コメント投稿数

図 1　Eclipse におけるコミッタ昇格前の一般開発者の活動量の経時変化

表 1　開発者の活動量メトリクス [3]

メトリクス名	内容
活動期間	パッチ (コメント投稿) を行った月の総数
総パッチ投稿数	活動期間内でのパッチ投稿の総数
月パッチ投稿数	1 ヶ月当たりのパッチ投稿の中央値
総コメント投稿数	活動期間内でのコメント投稿の総数
月コメント投稿数	1 ヶ月当たりのコメント投稿の中央値

モデル [3] を取り上げる．[3] では，コミッターへの昇格基準が OSS プロジェクト内では暗黙的に存在すると仮定し，昇格基準として開発者の代表的な活動量に着目した予測モデルを構築している．具体的には，表 1 に示した OSS プロジェクト内での開発者の活動量を計測するメトリクスを定義し，ロジスティック回帰分析により予測モデルを構築する．

ロジスティック回帰分析は以下の式 1 で表現される．

$$P(y|x_1,\cdots,x_p) = \frac{1}{1+e^{-(\alpha_1 x_1+\cdots+\alpha_r x_r+\beta)}} \quad (1)$$

$y \in \{0,1\}$ は 2 つのクラスを表す目的変数であり，本研究では 1 がコミッター，0 が一般開発者を表している．x_i は説明変数，α_i および β は回帰係数である．説明変数は活動期間やパッチ投稿数などの活動量メトリクスを表しており，回帰係数は各メトリクスの重みを表している．$P(y|x_1,\cdots,x_p)$ は目的変数の条件付き確率であり，コミッターになる確率を示している．

予測モデルを実際に構築する際には，取得したデータセットをランダムに二分割し，半分を訓練データ，残りの半分をテストデータとしてモデルの構築・評価を行う．他のコミッター候補者予測モデルでは利用するメトリクスが異なったり，10-fold cross validation などを用いて予測モデルを構築するなどの差異は存在するが，基本的には予測モデルを構築する時点で取得可能な全期間のデータを用いている．

3.2　リサーチクエスチョン

コミッター昇格についての大まかな方針を提示しているプロジェクトは存在するが，数値等で具体的にコミッター昇格基準を示しているプロジェクトは皆無と言っても過言ではない．本研究では，コミッターの必要性と役割はプロジェクトの状況やプロダクトの成熟度などにより変化するものと仮定し，本稿では以下のリサーチクエスションに取り組む．

表 2　Eclipse のコミッター候補者と一般開発者の数

対象期間	コミッター候補者数	一般開発者数
2001/01/01~2017/10/31	138	7,139

RQ1: データサイズは予測精度に影響するか？

コミッターの昇格基準が一定ではなく変化するのであれば，予測モデルを構築する時点から過去のすべてのデータを用いるのではなく，直近のデータのみを用いる方が予測精度が向上する可能性がある．また，予測精度が向上するのであれば既存研究のように全期間のデータを用いる必要がないため，データ収集コストが削減できるとともに，予測モデル構築の際の計算量も削減できる可能性がある．そのため RQ1 では，直近の 1 年分，2 年分，3 年分とデータサイズを増やしながら予測精度の評価を行う．

RQ2: 予測精度に影響するメトリクスは期間によって異なるか？

コミッターの昇格基準が一定ではなく変化するのであれば，データサイズのみならずコミッターの昇格に重要な因子（本研究では活動量メトリクス）も予測精度に影響するはずである．期間毎にコミッター昇格に重要な因子が異なること，また，直近ではどの因子が最も重要視されているか（現在プロジェクトで最も求められている活動）を明らかにすることができれば，コミッター昇格を目指す一般開発者に有用な指針を提供することができる．そのため，RQ2 では，予測精度に影響する活動量メトリクスが期間毎に異なるかどうかを確認する．

4.　評 価 実 験

本章では，3.2 節で述べたリサーチクエスションに答えるために行った評価実験について述べる．

4.1　データセット

本研究では，Eclipse Platform プロジェクト（以降では単に Eclipse と呼ぶ）を対象としてデータセットの構築を行う．Eclipse を選定した理由は，大規模かつ長期間継続しているプロジェクトであり，コミッターの昇格基準の経時的変化を詳細に分析する上で十分なデータを取得できるためである．対象プロジェクトのデータ収集期間と開発者の内訳を表 2 に示す．

データの収集・整形の手順は以下のように行った．

手順 1：開発者データの収集　不具合管理システムから不具合票を収集する．不具合票には不具合の症状についての報告のみならず，不具合修正パッチの投稿履歴や不具合修正のための議論の内容（コメント）が記録されている．既存研究と同様に，不具合票からパッチやコメントの投稿を 1 回以上行った開発者の名前とそれぞれの開発者が行った投稿の日時を収集し，コミッターを含むすべての開発者の活動量メトリクス一覧を作成する．

手順 2：開発者の分類　開発者がコミッターに昇格した日時を版管理システムを用いて調べる．版管理システムにはコミットを行った開発者名や日時が記録されている．既存研究同様，版管理システムに初めてコミットした日時をその開発者のコミッター昇格日時として定義し，コミッターの一覧表を作成する．

こうして作成したコミッターの一覧表をもとに，活動メトリクス一覧表の開発者がコミッターまたは一般開発者のどちらであるか分類を行う．分類の際，開発者がコミッターである場合は，最初にパッチまたはコメント投稿を行った日時からコミッターに昇格した日時までの活動のみを活動量メトリクスとして用いる．

4.2 実 験 手 順

評価実験の手順は以下の通りである．

手順 1：予測実験用のデータセットの作成　コミッター候補者を予測するタイミングとして 2014 年および 2017 年の 2 時点を想定して，それぞれ直近の 1 年分から 5 年分のデータセットを作成する．また，既存研究との比較として，2001 年から 2016 年までのデータセットを作成する．

手順 2：データセットの調整　表 2 の通り，コミッターに比べ一般開発者は数十倍存在している．データセットを単純に二分割して一方を訓練データとしてそのまま学習させるとモデルに偏った学習をさせてしまい，予測精度が低下する可能性がある [9]．そのため，一般開発者の数がコミッターと同数になるようランダムに一般開発者を抽出する．

手順 3：予測モデルの構築と検証　手順 2 で作成した訓練データを元に予測モデルの構築を行う．予測モデルの構築にはロジスティック回帰分析を用いる．ロジスティック回帰分析で 2 クラスの分類を行うため，本研究ではコミッターを 1，一般開発者を 0 として定義し予測モデルに学習させる．この時，手順 2 で述べたようにコミッターの数は少数であるため，交差検証の一つである Leave-One-Out 法を用いてモデルの構築および検証を行う．コミッターになるかどうかを割合で表し，0.5 以上の開発者をコミッター，それ以下の開発者を一般開発者と定義する．なお，手順 2 でランダムに一般開発者を抽出しているため，予測結果が毎回異なる．そのため，本実験では手順 2~3 を 1,000 回行いその中央値を結果として用いる．

4.3 評 価 方 法

4.3.1 RQ1

RQ1 では，データセットの経時的変化がどのようにコミッター候補者予測の精度に影響するかを調査する．そのため，従来研究 [3] と同じ活動量メトリクス（表 1），かつ，ロジスティック回帰による予測モデルの構築を行う．さらに，前節で示したように直近 1 年分のデータ，直近 2 年分のデータと 1 年ずつデータサイズを増やしながら予測モデルをそれぞれ構築し，予測モデルの精度を評価する．予測モデルの識別性能は適合率および再現率を用いて評価する．

表 3　予測精度の推移（2014 年の予測）

対象期間	1 年	2 年	3 年	4 年	5 年	全期間
適合率	0.883	0.644	0.785	0.725	0.780	0.723
再現率	0.890	0.590	0.607	0.564	0.599	0.522

表 4　予測精度の推移（2017 年の予測）

対象期間	1 年	2 年	3 年	4 年	5 年	全期間
適合率	0.480	0.472	0.599	0.697	0.687	0.739
再現率	0.493	0.490	0.595	0.784	0.784	0.833

4.3.2 RQ2

RQ2 では，データセットの経時的変化をより詳しく調査し，コミッター昇格に重要な因子（プロジェクトでコミッターに昇格するために重要視されている活動）が時期によって異なるかどうかを明らかにする．そのため，ロジスティック回帰分析により求められる各変数のオッズ比を比較分析する．

4.4 実 験 結 果

4.4.1 RQ1: データサイズは予測精度に影響するか？

表 3 および表 4 にそれぞれ，2014 年および 2017 年に昇格するコミッター候補者を予測した結果を示す．なお，いずれの表にも既存研究との比較として，全期間のデータを用いた場合（2001 年から 2013 年までのデータを用いた場合（表 3）と 2001 年から 2016 年までのデータを用いた場合（表 4））の結果も合わせて示している．

表 3 より，2014 年に昇格するコミッター候補者を予測する場合は直近の 1 年分のみのデータを用いるのが最も予測精度が高くなることが分かる（適合率 0.883，再現率 0.890）．また，2 年分のデータを用いた場合の適合率（0.644）以外は，全期間のデータを用いた場合（適合率 0.723，再現率 0.522）よりも予測精度が向上している．

一方，表 4 より，2017 年に昇格するコミッター候補者を予測する場合は直近の 1 年分あるいは 2 年分のデータを用いた場合が最も予測精度が低くなることが分かる．また，より長い期間のデータを用いるについて予測精度が向上していくことも見て取れる．特に，2017 年に昇格するコミッター候補者を予測する場合は，全期間のデータを用いる方が最も予測精度が高くなることが示されている．

RQ1 の回答: 予測モデルの構築に用いるデータサイズは予測精度に影響を与える．Eclipse では，2014 年に昇格するコミッター候補者を予測する場合は，直近 1 年分のデータ（2013 年のデータ）を用いた場合，最も予測精度が向上した．一方，2017 年に昇格するコミッター候補者を予測する場合は，直近のデータを用いるよりも計測可能なすべての期間のデータを用いた場合，最も予測精度が向上した．

表 5　オッズ比の推移（2014 年）

対象期間	1 年	2 年	3 年	4 年	5 年	全期間
活動期間	1.284	1.281	1.231	1.395	1.525	2.547
総コメント数	5.523	2.428	2.282	2.291	2.387	1.420
月コメント数	0.829	1.050	1.182	1.250	1.323	1.441
総パッチ数	1.000	1.247	1.689	1.523	1.537	1.652
月パッチ数	1.000	1.000	1.929	1.637	1.919	1.988

表 6　オッズ比の推移（2017 年）

対象期間	1 年	2 年	3 年	4 年	5 年	全期間
活動期間	1.542	1.786	1.831	1.732	1.852	1.674
総コメント数	1.068	1.133	1.449	1.320	1.266	1.988
月コメント数	0.593	0.712	0.625	0.470	0.494	0.896
総パッチ数	1.000	1.000	1.000	1.000	1.000	1.000
月パッチ数	1.000	1.000	1.000	1.000	1.000	1.000

4.4.2　RQ2: 予測精度に影響するメトリクスは期間によって異なるか？

表 5 および表 6 にそれぞれ，2014 年および 2017 年にコミッター候補者を予測した結果から得られるオッズ比の推移を示す．RQ1 と同様に，全期間のデータを用いた場合の結果も合わせて示している．

表 5 より，2014 年に昇格するコミッター候補者を直近の 1 年分のみのデータで予測した場合の「総コメント数」のオッズ比が最も高い（オッズ比 5.523）ことが分かる．また，「総コメント数」のオッズ比は他の変数に比べ一貫して高い傾向にあるが，1 年分のみのデータを用いた場合に比べ 2 年分以上のデータを用いた場合には「総コメント数」のオッズ比は半減することも見て取れる．さらに，全期間のデータを用いた場合には，「活動期間」のオッズ比が最も高くなることも見て取れる．

一方，表 6 より，2014 年に昇格するコミッター候補者を予測するモデルに比べ，2017 年に昇格するコミッター候補者を予測するモデルでは，一貫して「活動期間」が最もオッズ比が高い傾向にあるが，変数間でオッズ比に大きなばらつきは観察できない．全期間のデータを用いた場合には，「総コメント数」のオッズ比が最も高くなることも見て取れるが，表 5 の結果ほどの大きな差はない．

RQ2 の回答: 予測モデルの構築に用いるデータサイズによりコミッターの昇格に重要な因子は異なることがある．Eclipse では，2014 年に昇格するコミッター候補者を予測する場合は，直近 1 年分のデータ（2013 年のデータ）を用いた場合，「総コメント数」が最も重要な活動量となる．一方，2017 年に昇格するコミッター候補者を予測する場合は，データサイズによって重要な因子が大きく異なることはなく，「活動期間」がその他の活動量よりも比較的重要な活動量となる．

5. 考　　察

本章では，4. 章での実験結果を踏まえてコミッター候補者予測モデルにおけるデータサイズの影響について考察する．

5.1　RQ1 の結果について

2014 年に昇格するコミッター候補者を予測する場合は，我々の当初の予想通り，直近のデータを用いる方が予測精度が向上することが分かった．一方，2017 年に昇格するコミッター候補者を予測する場合は，我々の予想に反して，より多くのデータを用いる方が予測精度が向上することが分かった．このような結果になった理由としては，以下の 2 つの原因が考えられる．

1 つは，2014 年に昇格したコミッターと 2017 年に昇格したコミッターの人数の違いによる影響である．2014 年に昇格したコミッターは 15 名，2017 年に昇格したコミッターはわずか 3 名であった．3 名をコミッター候補者とし，同数の一般開発者 3 名をランダムに選出してデータセットを構築しているため，1000 回分の試行の平均値を求めているとはいえ，3 名のコミッター候補者の活動量のばらつきにより精度の高い予測モデルを構築することができなかった可能性がある．

もう 1 つは，プロジェクトの成熟度の影響である．コミッターに昇格した開発者が 2017 年には 3 名のみであることとも関係するが，プロジェクトが成熟し大きな機能拡張やバグ修正の必要がなく，2017 年にはコミッター昇格の基準としてパッチ投稿やコメント投稿は寄与していない可能性がある．実際，表 6 から見て取れるように，「活動期間」以外の活動量はオッズ比が一貫して低い傾向にある．

これらから，データサイズを削減する方が予測精度を向上させる場合もあるが，データサイズの削減・変更が予測精度の向上には寄与しない場合もあると言える．コミッター予測モデルを実際に構築し適用する際には，開発者の活動量の経時変化に着目し大きな変動が観察されない場合には従来研究通りのより大きなデータセットを，開発者の活動量に大きな変動が観察される場合にはより直近のデータを用いてプロジェクトのトレンドに追従した予測モデルを構築するのが効果的であると考える．

5.2　RQ2 の結果について

2014 年に昇格するコミッター候補者を予測する場合に，「総コメント数」のオッズ比が他の変数および他のデータサイズの結果に比べて大きな値をとる理由を確かめるために，コミッター候補者および一般開発者の「総コメント数」の分布を比較した（図 2）．図 2 の縦軸は「総コメント数」を年数毎に比較

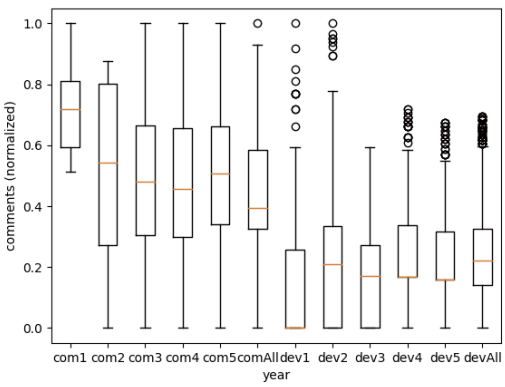


図 2　開発者の総コメント数の分布（2014 年）

するために正規化した値を示す．横軸の com1, com2... は，コミッター候補者の 1 年分，2 年分... のデータ，dev1, dev2... は，一般開発者の 1 年分，2 年分... のデータであることを表す．図 2 から，2014 年に昇格するコミッター候補者は直近 1 年，すなわち，2013 年の「総コメント数」が他の期間のデータとは分布が大きく異なり，「総コメント数」が多い傾向にある．さらに，一般開発者についても，2013 年の「総コメント数」が他の期間のデータとは分布が大きく異なり，「総コメント数」が少ない傾向にある．このように，コミッター候補者および一般開発者の「総コメント数」の分布の両方が他の期間の分布とは異なるため，オッズ比の大きな違いとして現れたものと思われる．結果的に，2014 年に昇格するコミッター候補者を予測する場合，「総コメント数」が最も寄与するため 1 年分のデータのみを用いた場合が予測精度としても良好な結果をもたらしたと考えられる．前述したように，コミッター予測モデルを実際に構築し適用する際には，コミッター候補者のみならず一般開発者の活動量の経時変化に着目することが，予測精度の向上およびデータサイズの削減につながるものと考えられる．

5.3 妥当性への脅威

5.3.1 コミッターの区別

本稿では，開発者がコミッターまたは一般開発者か定義する際に，版管理システムへのコミット履歴を元に分類を行っている．よって，既存コミッターであるにも関わらず，コミット履歴がパッチ投稿以前に行われていなかったためにコミッターとして分類できていない開発者が存在する可能性がある．実際に OSS プロジェクトで予測を行う際にはコミッターと一般開発者の区別は容易に行えるため，より正確に分類されたデータセットで予測可能であると考えられる．

5.3.2 メトリクスについて

実際の OSS プロジェクトでは，コミッター昇格候補者を検討する際にはどれだけ開発したかといった量的観点だけでなく，どのような開発やコミュニケーションを行ったかといった質的観点も考慮されると予想される．現在用いている活動量メトリクスでは開発者の技術力やコミュニケーション内容を元に予測できておらず，今後はこういった内容を分析できるメトリクスを調査する必要があると考えられる．

5.3.3 データセットの作成方法

本実験では年代を元にデータセットを減らしたが，年代毎ではなくパッチやコメントの投稿件数ごとにデータを区切る方法も考えられる．しかし，件数で区切った場合，時期やプロジェクトごとに件数が異なり，データの経時的変化をうまく捉えることができない可能性がある．また，実際の OSS プロジェクトにおいて予測を行う場合は，プロジェクトのプロセスや構造改善の時期を元に，基準となる年代の区切り方も変更できるため，より正確な予測を行える可能性がある

6. おわりに

本稿では，コミッター候補者予測の精度向上に向け，データサイズの変更による予測精度への影響を調べるための評価実験を行った．さらに予測精度の変化を元にデータに経時的変化が存在するか分析を行った．大規模 OSS プロジェクトである Eclipse を対象として実験を行った結果，取得可能な全期間のデータでモデル構築するよりも，直近の期間に絞ったデータでモデル構築した方が予測精度が向上する場合があることを明らかにした．また，予測する時期によってコミッター昇格のために重要視される活動の種類が異なる場合があることを明らかにした．

本稿では，既存研究 [3] に合わせてパッチやコメントの投稿数といった開発者の活動量メトリクスを用いて予測を行った．実際のコミッター候補者推薦ではパッチやコメントの投稿数だけでなく，それぞれの内容も推薦において重要な要因になっていると考えられる．今後の研究方針として，パッチやコメントの内容を考慮するためのメトリクスがないかどうか分析する予定である．その他に，近年は git による開発が主流であり，それらの開発形態においてもコミッター候補者予測において同様の活動量メトリクスが有意であるか分析を行う予定である．

謝　辞

本研究の一部は，本研究は JSPS 科研費 JP16K16037，JP17H00731 および JP18K11243 の助成を受けた．また，本研究の一部は，JSPS 特別研究員奨励費（JP17J03330）による助成を受けた．

文　献

[1] N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim, “Duplicate bug reports considered harmful . . . really?,” Proceedings of the 24th International Conference on Software Maintenance (ICSM ’08), pp.337–345, 2008.

[2] G. Jeong, S. Kim, and T. Zimmermann, “Improving bug triage with bug tossing graphs,” Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering (ESEC/FSE ’09), pp.111–120, 2009.

[3] 伊原彰紀，亀井靖高，大平雅雄，松本健一，鵜林尚靖，“Oss プロジェクトにおける開発者の活動量を用いたコミッター候補者予測，” 電子情報通信学会論文誌，vol.J95-D，no.2，pp.237–249，0 2012.

[4] 柏祐太郎，大平雅雄，阿萬裕久，亀井靖高，“大規模 oss 開発における不具合修正時間の短縮化を目的としたバグトリアージ手法，” 情報処理学会論文誌，vol.56，no.2，pp.669–681，Feb. 2015.

[5] C. Bird, A. Gourley, P. Devanbu, A. Swaminathan, and G. Hsu, “Open borders? immigration in open source projects,” Proceedings of the Fourth International Workshop on Mining Software Repositories (MSR ’07), p.No.6, 2007.

[6] A. Lee, J.C. Carver, and A. Bosu, “Understanding the impressions, motivations, and barriers of one time code contributors to floss projects: A survey,” Proceedings of the 39th International Conference on Software Engineering (ICSE ’17), pp.187–197, 2017.

[7] M. Zhou and A. Mockus, “What make long term contributors: Willingness and opportunity in oss community,” Proceedings of the 34th International Conference on Software Engineering (ICSE ’12), pp.518–528, 2012.

[8] A. Ihara, Y. Kamei, M. Ohira, A.E. Hassan, N. Ubayashi, and K. Matsumoto, “Early identification of future committers in open source software projects,” 14th International Conference onQuality Software (QSIC ’14), pp.47–56, 2014.

[9] H. He and E.A. Garcia, “Learning from imbalanced data,” IEEE Transactions on Knowledge and Data Engineering, vol.21, no.9, pp.1263–1284, 2009.