リポジトリマイニングにおけるCIツールの影響除去のための前処理手法の検討

戸　田　航　史[†1]

近年の Open Source Software (OSS) 開発の隆盛と共に，OSS 開発で用いられている各種開発ツール (git, Rietveld 等) のリポジトリの統計的解析（マイニング）も盛んになってきている．しかし，それらのリポジトリに統計処理を実施するためには様々な前処理が必要になる．本稿ではリポジトリから CI ツールの行動を除去するための前処理とそれに伴う問題について検討する．

Removing CI tools' action method from management system for Repository Mining

Koji Toda [†1]

Development tools (git, Rietveld, etc.) have been used in open Source Software and their repository are used for repository mining. However, it requires preprocessing for the quantification by various data to statistically processing the data from the repository. In this paper, It is discussed about removing CI tools' action from management system for repository mining.

1. はじめに

近年の Open Source Software(OSS) 開発の隆盛に従い，OSS 開発で用いられる各種開発管理システムのリポジトリを統計的に解析する，リポジトリマイニングが盛んになってきている．その背景には OSS 開発データの入手容易性があると考えられる．企業での開発管理ツールのデータに比べ，OSS の開発管理ツールのデータは誰でも容易に入手が可能という点がまず挙げられる．それに加え，git, rietveld 等の OSS 開発で用いられるツールの一部にはログを出力する機能を備えているものもあり，リポジトリマイニングのためのデータ入手を更に容易にしている．

リポジトリマイニングの対象としては git が多くを占める．その理由の 1 つとしては，前述のとおり，git には開発ログの出力機能，具体的には git log コマンドと，それに付随するオプションの設定の利用による自由度の高い開発ログの出力機能があるためと考えられる．それに加え，git のホスティングサービスである GitHub の全公開プロジェクトのデータが公開されている[1)2)] 事も影響していると考えられる．

git のリポジトリを対象としてリポジトリマイニングを行う場合，git log コマンドで出力された内容を特に統計処理に用いるためには，数量化に代表される前処理が必要となる．この前処理の内容は多岐にわたり，フィールドの区切り記号設定のための置換作業といった単純なものから，コメント解析のための自然言語処理，各種管理システム間の紐付け等様々である．しかしながら本稿では，前処理として継続的インテグレーション（CI）ツールの出力ログへの影響の除外において表れる問題を取り扱い，対象としてその影響が表れ易いバージョン管理システム，およびレビュー管理システムを扱う．

2. 出力ログへの前処理

大規模な OSS 開発プロジェクトでは，多くの場合 Jenkins に代表される CI ツールが導入，利用されている．ただし CI ツールの動作結果はプロジェクトでの運用によってはしばしばバージョン管理ツールやレビュー管理ツールのログにパッチ製作者やコミッター，レビュアーとして記録され，リポジトリマイニングにおける障害となることがある．動作結果は特にコミッター，レビュアーとして表れる事が多く，コミッターとして表れるのは各パッチのレビューページを監視し，特定の条件（LGTM[⋆1] が一定以上，コアメンバがレビューでプラス評価を付けている等）が満たされた時に，自動的にパッチをコミットする，という形で運用されている場合である．同様にレビュアーとして表れるのは，レビューページを監視し，パッチが登録される，またはパッチがアップデートされるたびに自動テ

†1 福岡工業大学
Fukuoka Institute of Technology

⋆1 Looks Good To Me の略．肯定的評価の慣習表現であり，OSS 開発においてはレビューの評価として広く用いられている

ストを行い，その結果をレビューコメントとして返す場合である．git（バージョン管理システム）では前者が，レビュー管理システムでは後者が CI ツールのログへの影響として表れるため，その影響をログから除外する必要がある．

CI ツールの影響を除外するにあたっては運用の個別性が問題となる．前述の置換作業や自然言語処理といった前処理は，一度習得すれば同じ開発管理システムを用いていさえすれば異なるプロジェクトでも流用可能である．これに対し，CI ツールはプロジェクトごとにどのように運用するかが異なっており，自動テストと自動コミットの例だけでも，自動テストには用いないが自動コミットには用いる場合，その逆，両方に用いる場合の 3 通りがあり得る．さらに自動テストを行うにしても単体テストだけの場合やある程度の結合テストまで行うものなどの派生があり，さらにバージョン管理ツール，レビュー管理ツール，CI ツールの種類や設定によってログへの影響（出力内容）が異なる場合がある．このため CI ツールの影響を除外するにあたっては，プロジェクトごとにその運用状況を個別に確認し，その内容に応じて処理対象を変更する必要がある．

3. CI ツールのログへの影響の対処

CI ツールが自動テストを実施している場合にはレビュー管理システムのリポジトリにその影響が表れ，自動コミットを実施している場合にはバージョン管理システムのリポジトリにその影響が表れる．実際のログにおいては，各管理システムの都合上，CI ツールも開発者として取り扱われる，すなわちバージョン管理システムではコミッターの 1 人として，レビュー管理システムではレビュアーの 1 人として，他の開発者と同様の扱いでログに記録される．このため，リポジトリから CI ツールの影響を除外するためには，それがどのような名前（もしくはメールアドレス）の開発者として駆動しているかを調査し，除外する必要がある．

CI ツールが管理ツール上で用いている名前，メールアドレスの扱いはプロジェクトごとに異なる．リストとしてまとめられ，公開されているプロジェクトもあればリスト等にまとめられていないプロジェクトもある．そして，リポジトリマイニングの実施あたっては駆動している CI ツールが公開されていない（まとめられていない）プロジェクトに対しては，分析者が逐一ログを手作業で確認，検出する以外に方法が無いのが現状である．

ただし，CI ツールはメールアドレスに「ci」「bot」「jenkins」といった語を含んでいる事が多いため，コミッター，レビュアーのメールアドレスに対し，これらの語で検索をかける事で，検出の手間をある程度省くことはできるただし，特に bot, ci については人名の一部として使われる事も多いため，目視による確認は必要である．

また，CI ツールのユーザ名，メールアドレスがまとめられ，公開されているプロジェクトの場合，その影響の除外は容易である．しかしそのようなプロジェクトでは多数の CI ツールが駆動している可能性があり（むしろ多数駆動しているためリスト化せざるを得なかった可能性がある），手作業による検出の必要は無いにせよ，除外に時間がかかる場合もある．例えば OpenStack の場合，駆動中の CI ツールはまとめられてはいるが，そこに記述されているメールアドレスが必ずしも CI ツールが用いているメールアドレスとは一致していない，具体的には単にツールの責任者のメールアドレスが記載されており，かつ責任者はそのメールアドレスでレビューや開発に参加している，という場合もあり [⋆1]，やはり分析者が手作業で確認しなければならない場合もあり，むしろ非公開の場合に比べて作業量が増大する可能性がある点に注意が必要である．

4. おわりに

本稿では CI ツールが駆動している OSS 開発においてリポジトリマイニングを実施する際に，各種開発管理ツールのログからその影響を除外する前処理について述べた．ただし，プロジェクトごとに CI ツールやその運用は異なり，実際の影響の除外においては本稿の前処理をプロジェクトの特性に応じて変更する必要がある．より汎用的に利用可能な影響の除去手段の考案が今後の課題である．

参考文献

1) Georgios Gousios. The ghtorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pp. 233–236, Piscataway, NJ, USA, 2013. IEEE Press.
2) The GHTorrent project. `http://ghtorrent.org/`.

⋆1 https://wiki.openstack.org/wiki/ThirdPartySystems