

発生頻度の少ないコーディング規約違反データ統合による 検出精度向上への試み

亀岡 令[†]
和歌山大学[†]

伊原 彰紀[‡]
和歌山大学[‡]

南 雄太[§]
和歌山大学[§]

大森 楓木[¶]
和歌山大学[¶]

1 はじめに

ソフトウェアを共同開発するプロジェクトでは、コーディング規約を遵守する可読性の高いソースコードの実装が期待される。具体的には、Python の PEP8 や Java の Code Conventions for the Java Programming Language など、各言語が標準的な規約を公開している。さらにプロジェクトは、保守性や可読性を高めるために独自の記法やプログラムの複雑度などの規約を定めることもある。 **TODO: [独自の記法の話背景すべきでしょうか？クロスプロジェクトできないと言ってるようなものな気が...]** 多くのプロジェクトでは静的解析ツールを用いることで規約を違反するソースコードの検出を自動化している。しかし、静的解析ツールが検出する違反は膨大であるため、優先的に修正すべき違反の選択が余儀なくされる。

従来研究では、機械学習を用いて優先的に修正すべき規約違反を推薦する手法を提案している [1]。しかし、一部の規約は発生頻度が低く、修正率や発生率が低い。このような規約は、推薦精度の低下に影響する。特に、新規プロジェ

クトでは過去に発生した違反修正履歴を持たないため、規約違反の推薦におけるコールドスタート問題の解決は容易でない。

本研究では、優先的に修正する規約違反コード断片の推薦におけるコールドスタート問題の解決に向けて、他のプロジェクトにおける違反修正履歴を用いて推薦モデルを構築する。さらに、発生頻度や修正率 **TODO: [発生頻度は関係ない？]** が低い規約違反の予測も実現するために、連合学習によってモデルを構築する。連合学習は、学習データから得られたモデルに加えて、別の学習データから得られたモデルも利用してモデルを構築するため、修正率が少ないような規約に対しても、他プロジェクトの学習データを用いることで予測を可能にする。 **TODO: [文章を少し変えたけど合ってる？]** 本研究では、プログラムの編集履歴から **TODO: [XX の]** 特徴量を計測し、特徴量を学習データとして連合学習の一つ **TODO: [FedAvg?]** を用いてモデルを構築する。 **TODO: [検証のことまでここに書くか結果などの分量によって要検討。]**

2 分析

2.1 モデル構築方法

モデル構築には、従来研究で使用された **TODO: [XX 件]** の説明変数に、変更行数や複雑度などソースコードの特徴量を加えた 44 個の特徴量を用いて、FedAvg アルゴリズム

Toward improving detection accuracy by integrating occasional coding convention violations datasets

[†] Ryo Kameoka, Wakayama University

[‡] Akinori Ihara, Wakayama University

[§] Yuta Minami, Wakayama University

[¶] Fuki Omori, Wakayama University

を用いた連合学習によりモデルを構築する [2]. 目的変数は, **TODO: [特定の?] コミット**において規約違反が修正有無の 2 クラス分類に取り組む. 図 ??は, 連合学習の概要を示す. リポジトリごとの深層学習によって得られたモデルを集約し加重平均を取り, 出力されたモデルをそれぞれの深層学習に配布する. 配布されたモデルを用いて個々のノードは再度学習を行う. モデルの構築, 得られたモデルの集約, 再度学習を繰り返すことで, より精度の高いモデルを得る. 本研究では, オープンソースフレームワークの OpenFL を用いて実装する. FebAvg において, 深層学習のレイヤーは 3 層で, 1, 2 層目は ReLU 層であり, 3 層目にシグモイド関数を用いた出力層となっている. **TODO: [この 3 層の説明は必要?]** 最適化アルゴリズムには Adam を利用し, 学習率は 0.01 に設定した.

評価実験では, 規約違反を修正したコミットが偏っていることがあるため, 予測データに正例が極端に含まれていないことを避けるため, コミットの古い順に 7 割を学習データ, 残り 3 割を予測データとする.

2.2 ケーススタディ

本研究では, GitHub で管理される **TODO: [スター数の多い?]** 10 リポジトリ (transitions, schematics, schema_salad, python-bugzilla, python-cloudant, pycard, pynput, OWSLib, howdoi, hickle) を対象とする. これらソフトウェアは全て Python で実装され, Pylint をしようしている. 本研究では, それぞれ 1,000 日間のコミット (中央値で 501 コミット) を分析対象とする.

図 1 は, 3 つのリポジトリ (schematics, transitions **TODO: [検証は 3 つのリポジトリでは?]**) において, 検出頻度に偏りがある

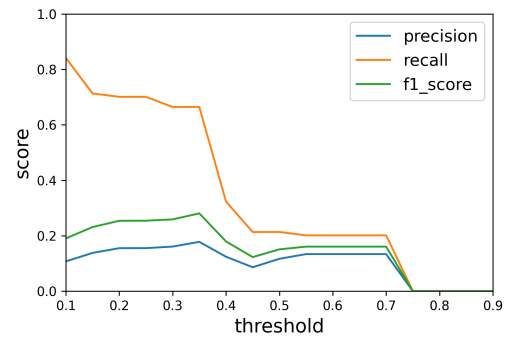


図 1 連合学習による予測結果

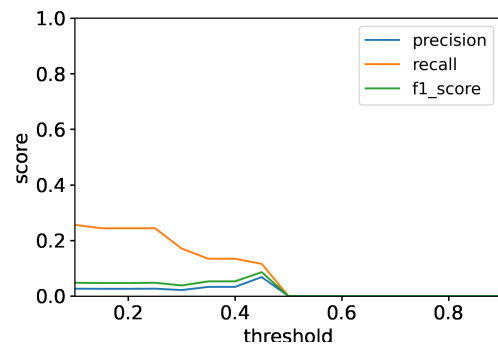


図 2 深層学習による予測結果

る **TODO: [修正率が低いではない?]** 規約 1 種類 (**TODO: [具体的な規約名ある? 修正率は?]**) の修正有無を予測した結果である. 横軸は判別閾値を 0 から 1 まで変化させたときの適合率, 再現率, F1 値を示す. 図 2 は比較対象として, 深層学習 (**TODO: [アルゴリズムは何?]**) を用いた結果を示す. 深層学習に比べ連合学習の方が高い精度で予測しているが, 紙面の都合上, 修正率 **TODO: [検出頻度?]** が低い規約の結果のみ示しているが, 修正率が高い規約であっても連合学習の方が精度が高いことを示した. 従って, **TODO: [結論を書く.]**

3 おわりに **TODO: [最後に読み直す]**

本研究では, 静的解析ツールによって検出された規約違反のうち特に修正回数や発生回数が低いものについて, 予測地点において修正されるかを連合学習によって予測精度向上を試み

た．今後の方針として，データセットの拡張，違反頻度の低い規約違反への重み付けなどを用いて予測精度の高いモデルの構築を目指す．

謝辞

本研究は和歌山大学「萌芽的個別研究支援」の助成を受けたものです．

参考文献

- [1] J. R. Ruthruff, J. Penix, J. D. Morgenthaler, S. Elbaum, and G. Rothermel: Predicting Accurate and Actionable Static Analysis Warnings: An Experimental Approach: Proc. of the Inter. Conf. on Software engineering (ICSE), pp. 341-350, 2008.
- [2] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas: Communication-Efficient Learning of Deep Networks from Decentralized Data: Proc. of the Inter. Conf. on Artificial Intelligence and Statistics (PMLR), pp. 1273-1282, 2017.