

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[†] 伊原 彰紀[‡] 南 雄太[§] 大森 楓己[¶]
和歌山大学[†] 和歌山大学[‡] 和歌山大学[§] 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。コーディング規約はプロジェクト開発のような複数人でコードを書く際に、可読性や保守性を高めるために定められた記法や、ソースコードの特微量についての規定のことである。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、修正率が低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。本研究では、プログラムの編集履歴からソースコード本体や変更に関する特微量を計測し、特微量を学習データとして連合学習の代表的な手法である FedAvg を用いてモデルを構築する。

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9個の説明変数に、変更行数や複雑度などソースコードの特微量を加えた44個の特微量を用いて、FedAvg 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、特定のコミットにおいて編集されたファイルに含まれる規約違反の修正有無の2クラス分類に取り組む。本研究で用いた連合学習は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

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

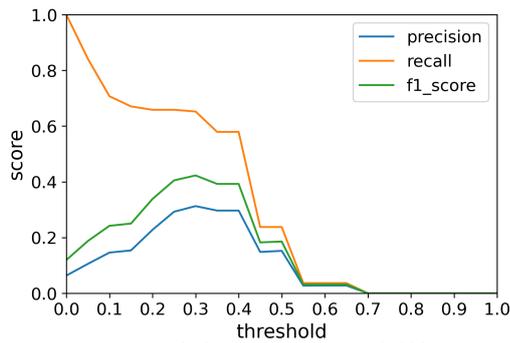


図1 連合学習による予測結果

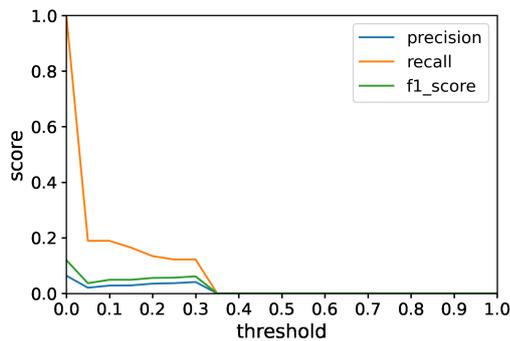


図2 深層学習による予測結果

ソースフレームワークの OpenFL を用いて実装する。その際、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り、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 Python 言語のための静的解析ツールである Pylint を使用する。対象プロジェクトは、Libraries.io から Python を主要言語とする 72,691 プロジェクトから、SourceRank(OSS の品質や活発度合いを評価する指標) 上位 1,500 件のプロジェクトの中で Pylint の設定ファイルを有し、かつ違反発生数の多い 10 件のプロジェクトを対象とする。

図 1 は、2 つのリポジトリ (schematics, transitions) において、修正回数が少ない規約 1 種類 (E0602/undefined-variable) について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、連合学習の際に用いたモデルと同じモデルを利用した深層学習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。紙面の都合上、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、クロスプロジェクトによって規約違反の修正予測は、単純に学習データを結合し深層学習する場合より、効果的に学習を行うことができ、コードスタート問題の解消への兆候を確認することができた。

3 おわりに

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、連合学習によって修正されるかの予測を試みた。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.