

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 データセット

本研究では、GitHub から Python で実装されており Pylint を用いて開発している transitions, schematics, schema_salad, python-bugzilla, python-cloudant, pyscard, pynput,

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

OWSLib, howdoi, hickle の 10 リポジトリを対象とする。これらのリポジトリからそれぞれ 1,000 日間のコミット（中央値で 501 コミット分）を分析対象とする。

2.2 分析手法

規約違反が修正されるかの予測には、従来研究で使用されたものや本研究で加えた変更行数や複雑度などソースコードや編集についての 44 個の特徴量を抽出し、FedAvg アルゴリズムを用いた連合学習を行う [2]。目的変数は、コミットにおいて規約違反が修正有無の 2 クラス分類とし、説明変数には規約違反の種類やソースコードの複雑度などの 44 個のパラメータを用いる。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場合があるので学習データとテストデータは、コミットの古い順から 7 割を学習データとし、残り 3 割を予測データとすることで、時系列を考慮した予測を行う。

図 1 は、FedAvg を用いた連合学習の概要を示し、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手法である。実装には、オープンソースフレームワークの OpenFL を用いる。

本論文で使用する FedAvg 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。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。

2.3 結果

図 2, 図 3 は schematics, transitions リポジトリに含まれる、Pylint で検出された規約違

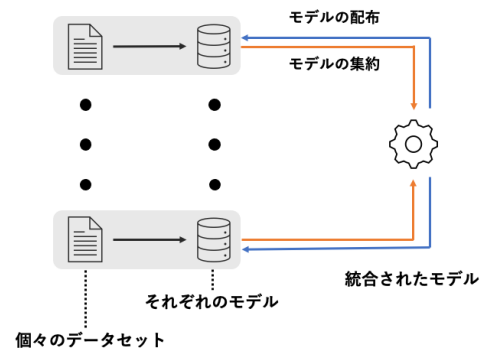


図 1 連合学習の概略図

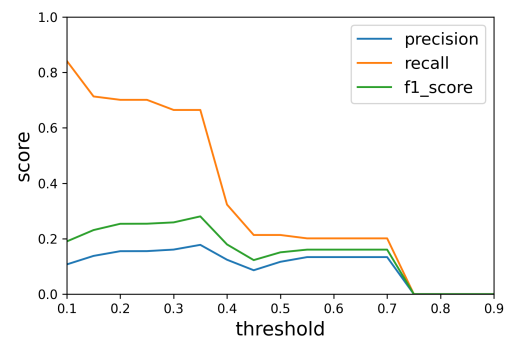


図 2 連合学習による予測結果

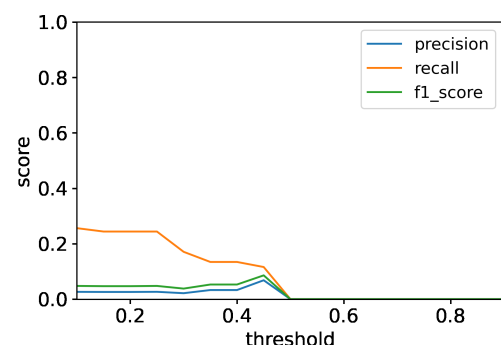


図 3 深層学習による予測結果

反のうち検出頻度に偏りがある規約 1 種類についての修正予測の評価結果である。閾値が低い場合に連合学習の再現率が高く、深層学習のモデルは閾値に関わらず全体的にすべての値が低いことがわかる。閾値が低い場合にも深層学習では再現率が低いことから、修正予測に必要な重みづけができていないことがわかる。

図 2 の結果において閾値が 0.35 の場合に最も適合率、再現率、F1 値のバランスが取れていることがわかる。しかし、閾値が 0.4 以上の場合には、適合率に大きな変化がないが、再現率が下がっている。したがって、連合学習によって規約違反修正の予測に効果がある兆候が確認できるが、モデルとして不十分である。

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、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

～～～

参考文献

[1] Joseph R. Ruthruff, John Penix, J. David Morgenthaler, Sebastian Elbaum, and Gregg Rothermel: Predicting Accurate and Actionable Static Analysis Warnings: An Experimental Approach: Proceedings of the 30th international conference on Software engineering(ICSE), pp. 341-350,

(2008)

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Agueray Arcas: Communication-Efficient Learning of Deep Networks from Decentralized Data: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (PMLR), No54, pp. 1273-1282, (2017)