プログラム正誤判定における プログラムのベクトル化と類似度評価の関係について

大嶋　琉太[1]　阿萬　裕久[1]　川原　稔[1]

概要：プログラミング自主学習支援サービスでは，解答プログラムの正誤が自動判定される．その際には多種多様なテストケースが必要不可欠であるが，それらを人手で作成するのは決して容易な作業ではない．この問題に対し，さまざまなプログラムを用意して記号実行解析を行うことでテストケースの生成を支援するという方法がある．本稿では，効率的なテストケース生成に向けて，プログラムの特徴ベクトル化とそれによる類似度評価の関係性について，データ分析の観点から検討を行う．

On Relationship Between Program Vectorization and Similarity Evaluation for Right/Wrong Judging

1. はじめに

近年，プログラミング学習に対するニーズの高まりに伴い，オンラインジャッジシステムのようなプログラミングを自主学習できるシステムが注目されている．オンラインジャッジシステムでは，プログラムの正誤を自動判定するために多数のテストケースが使用される．しかし，全てのテストケースを人手で準備することは容易な作業ではないため，記号実行技術 [1] と機械学習によるプログラムの類似性判定を用いた支援が研究されてきている [2]．その際には，プログラムの類似度を適切に評価することが重要である．そこで本稿では，プログラムの特徴ベクトル化とそれによる類似度評価に着目し，実際のプログラミング学習支援サイトに投稿されていたプログラムについてデータ分析を行う．

2. 正誤判定の効率化のためのプログラム類似度評価

あるプログラミング問題の正誤判定を行う場合，本来であれば多数のテストケースが必要とされるが，模範解答となるプログラムを出題者が用意しておけば，その後は解答プログラムとの等価性を記号実行によって判定するだけで正誤判定並びにテストケース生成を自動化できるという利点がある．しかしながら，記号実行という処理は決して軽量ではないため，より少ない処理回数で全てのプログラムの正誤判定を行う手法が従来から研究されてきている [2]．

従来手法では，はじめにいくつかの判定対象プログラムについて記号実行解析を行い，模範解答プログラムとの等価性を判定するとともにテストケース集合 T を生成する．そして残りのプログラムに関しては，いったん T に含まれるテストケースでもってテストを行い，不具合が見つからなかった場合にはさらに記号実行解析も行って正誤判定を実施する．ただし，その際にはプログラムを特徴ベクトル化しておき，その時点での正解プログラム群との類似度を機械学習モデルにより評価して，一定以上類似していれば記号実行解析を省略して “正解とみなす” という手法をとっている．

しかしながら，この手法では “本来は不正解であるはずのプログラムを誤って正解とみなしてしまう” 場合がある．その主な要因として，次の二つが挙げられる．一つ目は，テストケース集合 T の多様性が乏しく，多種多様なテストケースを用意できていないことが考えられる．もう一つは，プログラムの類似性を適切に評価できていないことが挙げられる．前者の問題を解決するには，はじめに可能な限り多様なプログラムをサンプルとして用意（選択）し，記号実行解析を行うことが重要である．後者の問題を解決するには，より良い特徴ベクトル化手法とそれによる類似性評価法を模索する必要がある．いずれについても，プロ

[1] 愛媛大学
Ehime Uniersity

グラムの類似度を適切に評価することが鍵となる．そこで我々は，“プログラムを特徴ベクトル化することで類似度・非類似度を適切に評価できるのか” という問いの下，データ分析を行うこととした．

3. データ分析

本分析では，二種類のベクトル化手法を用いて解答プログラムの特徴をベクトル化し，機械学習モデルで “正解とみなす” にはどちらの手法が優れているのかを比較・検討する．具体的には，プログラミングコンテストサイト codeforces 上の “一つの整数を入力とし，それに応じた文字列を出力する” というある問題に対して投稿されたプログラムの中で，正解・不正解と判断されたものをそれぞれ 100 個ずつ抽出し，Bag-of-N-gram と Doc2vec を用いた特徴ベクトル化を行う．さらに，特徴ベクトルをクラスタリングし，クラスタ内の不純度（正解・不正解の占める割合）を算出する．この不純度が高いと，ベクトル空間上では近い（同じクラスタに属する）プログラム同士であるにも関わらず，正誤が異なっているということになる．つまり，そのようなデータは機械学習モデルでもって適切に “正解とみなす” ことは難しいといえる．

以下に k 平均法を用いてクラスタリングを行った結果を示す．Bag-of-N-gram 及び Doc2vec による特徴ベクトルのクラスタリング結果は，それぞれ**図 1** 及び**図 2** に示す通

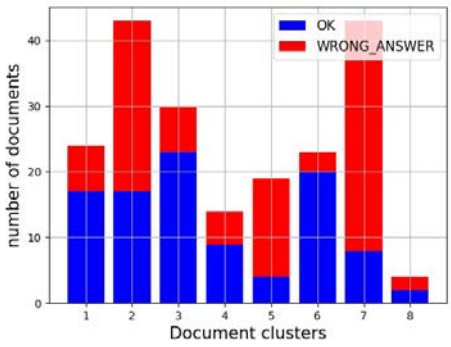

図 1: クラスタリング結果（Bag-of-N-gram）

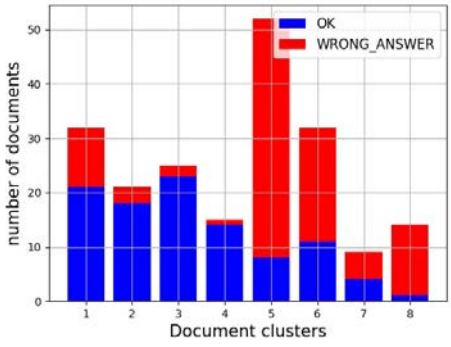

図 2: クラスタリング結果（Doc2vec）

表 1: 評価結果

ベクトル化手法	ジニ不純度 G の合計
Bag-of-N-gram	3.070
Doc2Vec	2.306

りとなった．この際，エルボー法を用いて最適なクラスタ数を算出しようと試みたが，誤差指標の急激な変化点が見られなかったため，経験的にクラスタ数を 8 とした．図 1 及び図 2 では，横軸はクラスタ番号，縦軸はクラスタに含まれるプログラムの個数を表しており，クラスタ内での正解プログラムを青色，不正解プログラムを赤色で示している．また，本分析では各クラスタの不純度をジニ不純度 G で表し，G の和でもってベクトル化手法の良さを評価することとした．結果を**表 1** に示す．

4. 考察

分析結果から，両手法間で各クラスタの不純度に差があることがわかった．例えば，図 1 におけるクラスタ 2 とクラスタ 8 は正解・不正解の占める割合がほぼ半々であるが，そのようなクラスタは図 2 ではクラスタ 7 のみである．ジニ不純度を比較すると，表 1 に示すように Doc2vec は Bag-of-N-gram よりもジニ不純度 G が低く，Doc2vec は “みなし正解” の精度向上により有効であると考えられる．これは，我々の先行研究 [3] の結果にも符合するものであった．今回は二種類のみの比較を報告したが，さらにさまざまなベクトル化手法の使用を検討していけば，プログラム正誤判定の判定精度向上と効率化へつなげることができると期待される．

5. おわりに

本稿では，プログラムの特徴ベクトル化とそれによる類似度評価の関係性について，データ分析の観点から検討を行った．ワークショップでは，他のベクトル化手法の活用や類似度評価の方法について議論したい．

謝辞 本研究の一部は JSPS 科研費 20H04184, 21K11831, 21K11833 の助成を受けたものです．

参考文献

[1] 酒井政裕, 岩政幹人：記号実行によるプログラム改造支援技術, 東芝レビュー, Vol. 67, No. 12, pp. 35–38 (2012).
[2] Rastogi, I., Kanade, A. and Shevade, S.: Active Learning for Efficient Testing of Student Programs, in Penstein Rosé C. et al., ed., *Artificial Intelligence in Education*, Vol. 10948 of *Lecture Notes in Computer Science*, pp. 296–300, Springer (2018).
[3] 大嶋琉太, 阿萬裕久, 川原稔：プログラムのベクトル化と記号実行を活用した正誤判定の効率化, ソフトウェア工学の基礎 29, 近代科学社 Digital, pp. 85–90 (2022).