

令和 6 年 6 月 14 日現在

機関番号：12101

研究種目：基盤研究(C) (一般)

研究期間：2021～2023

課題番号：21K02768

研究課題名(和文) プログラミング科目のレポート課題の採点効率化～類似答案の可視化と時系列追跡～

研究課題名(英文) Efficient Scoring of student reports in Programming Courses - Similarity
Visualizaiotn and Tracing over Time -

研究代表者

梅津 信幸 (Umezu, Nobuyuki)

茨城大学・理工学研究科(工学野)・准教授

研究者番号：30312771

交付決定額(研究期間全体)：(直接経費) 2,900,000円

研究成果の概要(和文)：大量の演習が必要なプログラミング科目において、学生から提出された大量のレポート課題を効率的に評価する手法を開発し、採点作業に要する時間を大幅に短縮することを目的とし、学生のプログラム(ソースコード)を抽象構文木(AST)に分解し、マルコフ過程行列の類似度としてクラスタリング結果をデンドログラム(樹形図)として可視化する手法を開発した。計200名が履修する初級プログラミングの講義でのレポート課題のべ4000件程度を匿名化したデータに提案手法を適用し、採点に要する時間がおよそ2割程度まで短縮された。研究成果について国際会議で2件の発表を行い、学術誌論文にも投稿中である。

研究成果の学術的意義や社会的意義

生成AIの興隆によりプログラミング作業は人間が行う必要はなくなるなどの意見もあるが、工学部学生にとってはむしろ、それらのAIを高度に制御する側の人材としてプログラミングの重要性はさらに増している。プログラミング能力の涵養、特に初期の段階にはとにかく大量の演習による経験の蓄積が重要であり、大学の講義においても学生の主体的学習を促すレポート課題は必須である。週に10時間、20時間と要するレポート採点作業が本手法により数分の1の時間に効率化されることで、学生一人一人の提出物により細かくコメントを与えて理解を助けるなど、より効率的な学習を支援できる。

研究成果の概要(英文)：In programming courses that require a large number of exercises, we developed a method to efficiently evaluate a large number of report assignments submitted by students and significantly reduce the time required for scoring. Our method visualizes the clustering results as a dendrogram (tree diagram) based on abstract syntax trees (ASTs) converted from student codes. We applied the proposed method to anonymized codes of more than 4,000 report assignments from 200 students in an elementary programming course, and found that the time required for grading was reduced to approximately 20%. We presented our results at two international conferences and submitted a paper to an academic journal.

研究分野：マンマシンインタフェース, 教育工学

キーワード：抽象構文木(AST) デンドログラム(樹形図) クラスタリング 類似度

1. 研究開始当初の背景

プログラミングを扱う演習系の科目では十分な量の演習課題を通じた学習が不可欠である。一方、課題を評価・採点する教員やTAの作業量が膨大となることから、現実的には課題の分量もかなり制限される。学生が各自でプログラムを作成して提出する課題は、採点作業を複数人で分担すると評価基準の統一が困難(=著しく煩雑)になるという問題点がある。特に、類似した提出物が複数あった場合にはそれらに類似(または同一)の評点を与える必要があり、この評価の一貫性(=類似内容には類似の評点)の確保には多大な労力を要する。この作業をもし複数人で分担したとしても膨大な時間がかかる(結局、各人がすべてのソースを見直しながら採点することになる)ため、現実的には特定の教員が単独で採点を担当することが多い。

そこで、本研究が目指したのは、プログラミングの演習系科目の提出物について

「統一的な基準による大量の評価は可能か、どこまで効率化できるか」

という問いに関する答えである。

研究代表者が現在担当する科目の例では、約200人の受講者に対して採点作業をすべて手動で行った場合は1課題あたり15~20時間を要する。研究開始以前から進めていた部分的な自動化により5~7時間に短縮されつつあるが、本研究課題により効率化をさらに進めて数分の一の作業時間を目指した。

2. 研究の目的

本研究では、学生から提出された大量のプログラミング課題の採点作業の効率化を目的とする。個別の提出物をそのまま採点するのでは、常に学生の人数分の作業が避けられない。一方、もし類似するプログラムごとに提出物が事前にグループ化してあれば、教員の採点作業はそのグループ数だけの作業にとどまり、著しく効率化される。例として、200人から提出されたプログラムが10グループに分類できれば、採点に要する時間も1/10程度に減少する。ここで得られた類似性グループの数を使って、たとえば

- クラス全体が単一のグループであれば、おそらく課題がシンプルすぎる
- 多様な解き方が予想される課題で、提出された内容が2, 3個のグループであれば、おそらく課題が難しすぎて、解答にたどり着いた数名のプログラム例がクラス全体で共有されている

などの判断も可能となる。

さらに、得られたグループについて、複数のレポート課題にまたがって時系列で変化を追いかければ学生間の交流関係が把握でき、より効果的な教科指導につながる。

3. 研究の方法

学生のレポート内容に関する既存のチェック手法は一般的に、他との類似や盗用・剽窃を、他の学生の提出物との比較や、ネット上や書籍などの情報源との比較により発見する。これらは、ある程度の長さがある日本語や英語などの自然言語の文章を対象としており、他のコピー・剽窃が基本的に許されない分野である。レポート課題だけでなく、特許や文学作品などの知的所有権で類似性の判定の需要があることから、すでに大量の研究例があり、商用のソフトウェアやサービスも多数ある。

一方、プログラミングに関するレポート課題には特殊な事情がある。他との類似度が高いことは、コピー・剽窃の可能性を示唆するだけであり、すぐには不正発見を意味しない。コンピュータのプログラムはある面で数学の定理や証明の記述と似ており、短いものだと誰が書いてもほぼ同様の内容になってしまう場合がある。また、特定の処理には定石の書き方があり、その部分には差が出ないことが望ましい場合も少なくない。つまり単に同一や類似というだけでは剽窃や不正とは判断できない。

加えて、研究代表者は講義中に「他人から教えてもらったプログラムでも自分自身で十分に理解した上であれば、提出することに問題はない」と伝えている。課題について自分一人で考えてわからないまま未提出にするよりは、他人に教えてもらい十分理解した上で提出する方が望ましいとの判断から、コピーそれ自体は特に問題視していない。

ただ、学生間でプログラムが共有される状況そのものには常に注視する必要がある。そこで、従来手法のように剽窃やコピーを検出すること自体を目的とするのではなく、類似性に基づいた可視化までを自動化し、そこから示唆される情報の評価に教員が専念できる環境を提供する点で他の研究例と異なる。

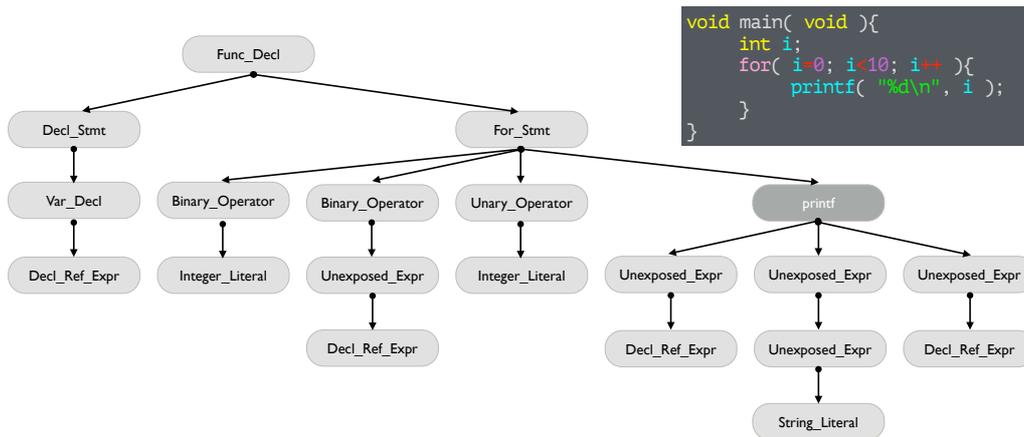


図1. ソースコードの抽象構文木 (AST) への変換例

提出されたレポート評価作業の効率化に際し、本研究ではまずソースコード間の類似度をどのように定義すべきかについて調査検討した。当初の時点 (文献1) では、トークン分解に基づいた手法が有望と考えていたが、途中で抽象構文木 (AST) を用いた後述の手法 (論文投稿中) に切り替えた (上の図1)。すべての提出課題についてASTに基づいて計算した特徴行列を用いて互いの類似度を計算し、互いに類似するグループへと分割する (クラスタリング、下の図2)。このとき、手作業で課題を評価した時に教員が適切と判断するクラスタ数と、クラスタリング結果を可視化した樹形図 (デンドログラム) での分岐の状況がどの程度一致するかを調べた。

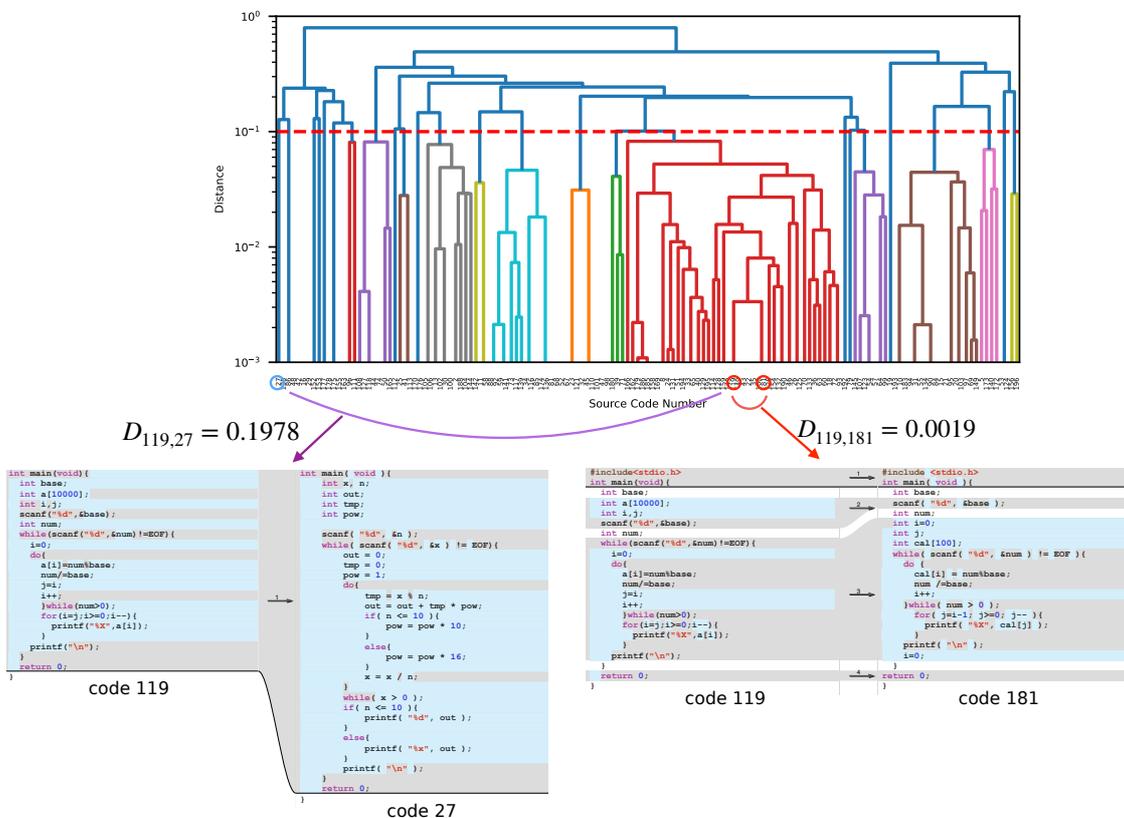


図2. ソースコードの類似度に基づく階層化クラスタリングの例。左のペア (学生119と27) には差が見られる (相違度0.198) のに対し、右のペア (学生119と181) にはほぼ差がない (相違度0.002)

4. 研究成果

- (1) ソースコード類似度の指標
当初は、コードをトークン化した後にレーベンシュタイン距離（編集距離）を用いて類似度を定義していたが、抽象構文木（Abstract Syntax Tree, AST）による表現の方が、ソースコード採点時の教員の感覚に近いことが判明し、ASTに対応する状態遷移行列の類似度として可視化する手法を考案した。
- (2) 採点作業支援システムの開発
Pythonと各種ライブラリを用いて構築したシステムにより課題1件（200提出物）あたり約2秒で処理可能である。図3に実行中の画面例を示す。
- (3) ソースコードのクラスタ数
提出されたソースコードについて、それらの間の距離（相違度）と実際のコード内容を目視で比較検討し、距離は線形ではなく対数軸での評価が妥当であること、および200名程度のクラスサイズに対して5~15のクラスタ数での分割が教員の経験に基づくグループ数と一致することを確認した。
- (4) レポート課題採点作業の効率化
講義TAを担当可能な大学院生、4年生20名を対象とした評価実験を実施した。提案手法により採点作業の品質を落とさず2~3倍の高速化が達成された（図4）。実際の講義での利用においては採点はTAではなく熟練の教員が担当するため、現実的には5倍程度の高速化につながる。
- (5) プログラミング科目の成績の事前指標
予測指標として基礎的な数学の能力（図形描画、関数の変形・解釈）について匿名で追跡した結果、初歩的なプログラミング科目では数学基礎との相関はほとんど見られず、明確な相関はおもに講義・学習への意欲にのみ現れた（文献2）。

これらの研究成果は文献1), 2)として国際会議で発表済みであり、さらに学術誌論文にも投稿中である。

参考文献

- 1) Umezu N., Iwai M., Ito H., Similarity-Based Grading Assistance for Assignments in Programming Lecture, Proc of IEEE GCCE 2023, Nara, Japan, pp.285-287, Oct 10-13, 2023, <https://ieeexplore.ieee.org/document/10315677/>
- 2) Umezu N., Ito H., Yanai HF, Performance in Programming Class and Basic Math Skills, Proc of IEEE GCCE 2023, Nara, Japan, pp.565-567, Poster, Oct 10-13, 2023, <https://ieeexplore.ieee.org/document/10315565/>



図3. 提案システムの画面例

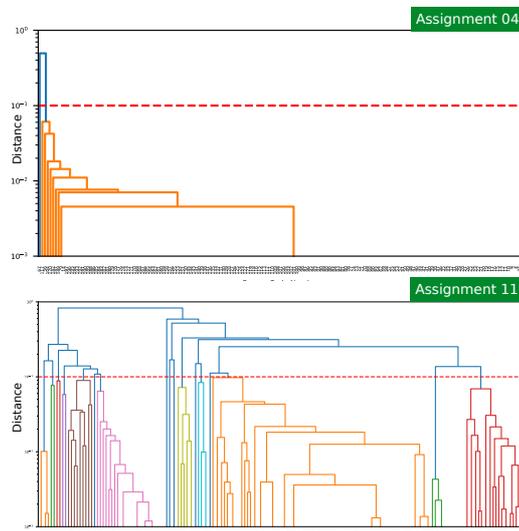


図3. 課題04, 11におけるソースコードの類似度によるデンドログラム。04は全ての提出内容が同一に近い簡単な課題である。

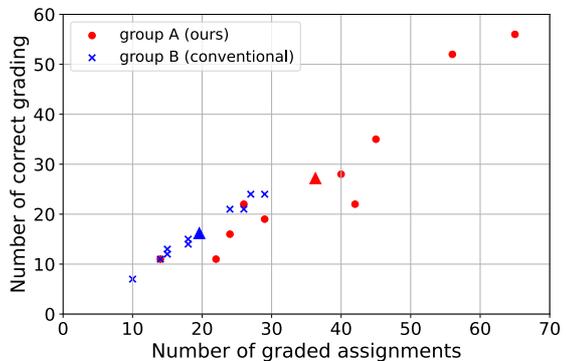


図4. 20名のTA相当学生による実験結果。従来手法（青のグループB）に比べ、提案手法を用いたグループA（赤）は単位時間あたりの採点数が大きく向上した。

5. 主な発表論文等

〔雑誌論文〕 計0件

〔学会発表〕 計4件（うち招待講演 0件 / うち国際学会 4件）

1. 発表者名 Motoki Iwai, Nobuyuki Umezu
2. 発表標題 Similarity Visualization of Source Code Assignments Submitted in Programming Classes
3. 学会等名 ICECE 2021 (国際学会)
4. 発表年 2021年

1. 発表者名 Motoki Iwai, Nobuyuki Umezu
2. 発表標題 Visualizing Similarity of Source Codes for a Programming Class
3. 学会等名 ISCI 2021 (国際学会)
4. 発表年 2021年

1. 発表者名 Nobuyuki Umezu, Motoki Iwai, Hiroto Ito
2. 発表標題 Similarity-Based Grading Assistance for Assignments in Programming Lecture
3. 学会等名 Proc. of IEEE GCCE 2023, pp.285-287, https://ieeexplore.ieee.org/document/10315677/ (国際学会)
4. 発表年 2023年

1. 発表者名 Nobuyuki Umezu, Hiroto Ito, Hiro-Fumi Yanai
2. 発表標題 Performance in Programming Class and Basic Math Skills
3. 学会等名 Proc. of IEEE GCCE 2023, pp. pp.565-567, https://ieeexplore.ieee.org/document/10315677/ (国際学会)
4. 発表年 2023年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
--	---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------