

プログラミング演習における学生の プログラミング行動推定手法の検討

筒井 善規¹ 堀口 諒人¹ 井垣 宏¹

概要：

リモート環境でのプログラミング演習では、対面で実施する場合と比較してより学生がどこに詰まっているかを早期に教員が確認できることが重要となる。そこで本研究では、ユーザのプログラミング行動特定のための様々な仮説を検証するため、学生がプログラミングを行う際のファイル編集やコマンド実行の履歴を収集して仮説にもとづいて分析し、実際の行動と比較する手法を検討する。

1. はじめに

大学生を対象とするプログラミングの教育を目的としたプログラミング演習と呼ばれる形式の授業が数多くの大学で行われている。一般に、プログラミング演習は対象のプログラミング言語に関する課題が学生に対して課され、少数の教員とアシスタントが数十名～100名程度の学生を指導するといった形式で実施されている [1]。

これまでの多くのプログラミング演習は対面での授業を前提としている。しかしながら、今後 With コロナの状況下においてリモートでプログラミングを指導することが前提となると、一人ひとりの学生がどの課題のどの構文要素でどの程度行き詰まっているか、こちらからのフォローが必要かどうか、どのような課題が個々の学生に求められているかといったより詳細な学生のプログラミング行動の特定が重要となると考えられる。オンラインで学生のプログラミング行動を推定する様々な手法が提案されているが、教育機関によって学生のスキルレベルや課題の実施方法が異なるため、そのまま適用できるとは限らない。

そこで本稿では、ファイル編集履歴やコンパイルなどのコマンド実行履歴を記録可能なプログラム開発環境を学生に配布し、各種ログから学生のプログラミング行動を様々な仮説と組み合わせて検証する手法について検討する。

2. プログラミング行動検証のための開発環境

学生のプログラミング行動を特定するためには、プログラミング環境、ログ収集環境が必要となる。さらに、様々

な仮説を立てて学生の行動を検証し、その仮説が正しいかどうかを分析するためには、より詳細に学生の行動を観測し、具体的なプログラミング行動をラベル付けできる環境が必要となる。

そこで我々は実際のプログラミング演習で利用する開発環境をプログラミング開発環境、ラベル付けのための環境をプログラミング開発実験環境と呼ぶ。本研究では Visual Studio Code, Java プログラミングのための Amazon Corretto11, コンパイル等を行う Windows 用の bash ターミナルをプログラミング開発環境として用意する。Visual Studio Code にはエディタに対して何の入力も行われなくなると自動的に変更されたファイルのスナップショットを保存する機能がある。その機能を利用して5秒以上学生が編集作業を停止した場合にそれまでのファイルの編集内容を履歴として保存する。また、bash ターミナルを利用してコンパイルや実行が行われた際に、bash_history 機能を利用し、どのコマンドがいつ実行されたかを記録する。プログラミング開発実験環境は、以上のプログラミング開発環境にあわせて、開発中の画面をキャプチャするソフトウェアが追加されている。以降の節では、我々が実施した実験内容と仮説検証の結果について述べる。

3. プログラミング開発実験

13名の大学3,4年生を対象にプログラミング開発実験環境を利用した実験を行った。被験者にはJava言語を対象とした4つの課題を提示し、すべてを解けるまで取り組んでもらった。実験中は被験者のパソコン全画面を動画としてキャプチャし、実験終了後にそれぞれいつ何をしていたかを確認するインタビューを行った。収集したログは、2

¹ 大阪工業大学
Osaka Institute of Technology 1-79-1 Kitayama, Hirakata
City, Osaka, 573-0196 Japan

表 1 ある被験者のプログラミング行動例

start	end	課題名	action
00:10:45	00:11:03	kadai1	プログラミング
00:11:03	00:11:44	kadai1	プログラミング
00:12:45	00:14:01	kadai1	検索 (教科書)
00:14:01	00:14:14	kadai1	プログラミング
00:14:14	00:15:22	kadai1	検索 (教科書)

節で述べたファイル編集履歴とコマンド実行履歴の 2 種類と画面キャプチャである。1 時間程度の実験終了後に、以下の手順で被験者ごとのプログラミング行動を特定した。

Step1. ファイル編集履歴からファイルごとのスナップショットが保存された時刻をすべて抽出する

Step2. 実験開始時刻から最初のスナップショット保存時刻まで、最初のスナップショット保存時刻から次のスナップショット保存時刻まで、といった形で実験中の時系列を区切り、区切りごとに被験者が何をしてたかをキャプチャした動画及びインタビューで特定する
実際に作成したある被験者のプログラミング行動を表 1 に示す。start と end は Step2 で述べた連続するスナップショット保存時刻（開始時刻からの差分）より特定した区切りを表している。action には動画及びインタビューから得られたプログラミング行動ラベルが記述されている。今回の実験では、プログラミング、問題確認、休憩、質問、検索 (Web)、検索 (教科書) の 6 種類のラベルを付与している。なお、プログラミングラベルについては、ファイルを編集しており、他の 5 種類のいずれでもない場合にラベル付けを行っている。表で色がついている区切り（表 1 の 3 行目と 5 行目）は被験者が自力で課題を解くことができず、質問、検索 (Web)、検索 (教科書) のラベルが割り当てられたことを示している（以降行き詰まりと呼ぶ）。

4. プログラミング行動推定のための仮説検証

前節で求めた被験者の実際のプログラミング行動とファイル編集履歴やコマンド実行履歴から仮説にもとづいて抽出した被験者のプログラミング行動を比較することで、仮説の精度を評価することができるようになる。ここでは「ファイルの編集時間が 30 秒以上たっている場合、行き詰まっている」という仮説について、実際に検証を行う。ファイルの編集時間、すなわちデータセットの start と end の差分が 30 秒以上の場合に行き詰まっているものと推定する。表 2 に仮説にもとづいてファイル編集履歴から推定した結果を示す。推定した区切りは表 1 と同一である。仮説によって行き詰まっていると判断された区切りには action 列に「詰まっている」というラベルを付与している。

このようにユーザのプログラミング行動の正解データセットと仮説にもとづいて算出した結果とを比較することで、仮説の精度を計算することができる。表 1 と表 2 にもとづいて適合率と再現率を算出してみる。まず適合率は

表 2 抽出されたデータセット例

start	end	課題名	action
00:10:45	00:11:03	kadai1	
00:11:03	00:11:44	kadai1	詰まっている
00:12:45	00:14:01	kadai1	詰まっている
00:14:01	00:14:14	kadai1	
00:14:14	00:15:22	kadai1	詰まっている

表 3 実験結果 (%)

	kadai1	kadai2	kadai3	kadai4
適合率	21.64	24.67	9.14	36.03
再現率	90.06	77.67	50.00	85.00

(仮説にもとづいて詰まっていると判断された中で正解データセットでも詰まっていると判定された区切りの数) を (仮説にもとづいて詰まっていると判断した数) で割ったもので計算している。表 2 で考えると $2/3=0.67$ が適合率となる。同様に再現率は、(仮説にもとづいて詰まっていると判断された中で正解データセットでも詰まっていると判定された区切りの数) を (正解データセットで詰まっていると判定された区切りの数) で割ったもので計算している。上記例で考えると、 $2/2=1$ となる。

実際に今回の仮説によって得られた 13 名の被験者それぞれについて課題ごとの適合率と再現率を求め、課題ごとに適合率と再現率の平均値を取った値を実験結果として表 3 に示す。

表 3 より仮説の適合率、再現率を見ていくと条件が「ファイルの編集時間が 30 秒以上たった場合、行き詰まっている」と容易な条件であるため再現率が高くなるが、条件にあってしまうログが多く取られてしまうため適合率が下がってしまうことが分かる。これより、さらに条件を追加していくことで、適合率、再現率が上昇していくと考えられる。

5. おわりに

13 名の被験者を対象に、プログラミング開発環境及び開発実験環境を配布し、複数のログ及び画面キャプチャ、インタビューを通じてプログラミング行動の正解データセットを作成し、仮説検証を実際に行った。実際のプログラミング演習では、一部の学生を対象に開発実験環境を利用してもらうなどしてデータを収集し、様々な仮説を立ててより正確なプログラミング行動の検証を継続して行っていくことを考えている。

謝辞 本研究の一部は JSPS 科研費 17K00500 の助成を受けた。

参考文献

- [1] 榎原絵里奈, 藤原賢二, 井垣宏, 吉田則裕, 飯田元ほか: 初学者向けプログラミング演習のための探索的プログラミング支援環境 Pockets の提案, 情報処理学会論文誌, Vol. 57, No. 1, pp. 236-247 (2016).