

コードパズルを用いたプログラミング的思考過程の時系列解析 Time Series Analysis of Programming Thinking Process Based on Code Puzzles

伊東 大輝[‡]
Hiroki Ito

島川 博光[†]
Hiromitsu Shimakawa

1. はじめに

本研究は、プログラミング教育における理解度をコードパズルにより推定する手法を提案する。昨今、知識を組み合わせる力であるプログラミング的思考力の育成が重要視されているが、現在の教育支援システムの多くはこれを考慮できていない。プログラミング的思考を用いて課題を解く過程を評価するには、思考の遷移を考慮する必要があると考えられる。そこで本手法はコードパズルと呼ばれるプログラミングの並び替え問題の解答過程を収集し、時系列解析により理解状況を推定する。本研究によって、学習者と指導者はプログラミング的思考力に準じた理解度を知ることができるほか、解析結果から不理解要因を特定し、指導者は新たな指導案を考案できる。

2. プログラミング教育を支援する関連研究

2.1 プログラミング教育を支援する関連研究

Crow ら [1] は、近年の Intelligent tutoring system について、ステップごとにヒントを与えるものや、提出されたコードに対して総括的なフィードバックを与えるような適応型フィードバックが普及してきていると報告している。Nesbit ら [2] も同様に、現状の研究の風潮は概念的な学習目標の達成を考慮できていないと推測している。すなわち、多くの既存研究は、知識の側面にばかり着目しており、論理に基づく、その思考過程を考慮した理解度推定を行っていない。

2.2 論理上の構成要素を考慮するコードパズル

プログラミングは、必要な知識と考え方のパターンがなければ問題解決の道筋をつけることはできない。Parsons ら [3] の提案した Parson's programming puzzle は、初学者が取り組みやすく、論理上の構成要素の習得に着目できるツールとして紹介されている。本手法では問題解決のための考えのパターンや知識の関係性に言及するスキーマ理論に基づき、また、データ収集の手軽さや回答のパターンに一定の制約がかけられる点から、コードパズルの問題形式を用いることに決定した。

スキーマ理論をもとにすれば、問題解決の道筋に見通しが立っている学習者のコードパズル操作は行動が一貫していると考えられる。反対に、見通しが立っていない場合は学習者の操作は行動に一貫性がなく、さまざまな思考に遷移すると考えられる。

3. ふるまいを用いた理解度推定

3.1 状態空間モデルによる学習のモデル化

本研究では、学習者がコードパズルを解くさいのふるまいから、学習者がどのような思考過程を考えられているか、また、その思考過程は適切かを推定する方法を提案する。

図 1 に手法の概要を図示する。

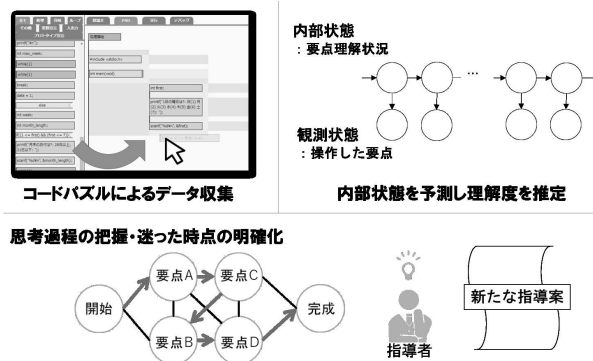


図 1: コードパズルの解答過程による思考過程の分析方法

本手法は、学習者の思考過程とそのときの理解度を分析するために、複数の学習者からコードパズルを解くさいのふるまいのデータセットを収集する。初学者向けのプログラミングの課題には、それぞれ理解すべき要点が設定されている。本手法では、この要点のことをサブゴールと呼称する。プログラミング課題を解くとき、順調な学習者はこのようなサブゴールを順番に解いていくと考えられる。反対に、見通しがついていなかったり、迷っている学習者はサブゴールを順番に解いていくことができないと考えられる。

すなわち、プログラミング課題を解く過程は”操作”を観測し、”理解度”を内部状態に持つ状態空間モデルとして表現できると考えられる。本研究では、この仮説に基づき、”コードパズルのどのサブゴールに属するブロックを操作したか”を観測状態とし、”そのサブゴールに取り組んでいるときの理解度”を隠れ状態とした隠れマルコフモデルを作成する。

本手法は、コードパズルに取り組んでいるときの要点に対する理解度を時系列情報から推測できることから、視線情報を取得する外部センサーなどを用いることなく「どの時点で何を悩んでいたか」を明らかにすることができる。これにより、指導者は学習者に足りないプログラミングの要素が特定できる。

3.2 操作ログを収集するためのインターフェース

本手法は、理解度を推定するためにコードパズルの操作ログを収集する。操作ログの記録に用いる WEB 上で動作するオリジナルのアプリケーションのインターフェースを図 1 左上に示す。

学習者は図 1 左上のように画面左側のコードブロック群から選択し、PAD[4]の形式でプログラムを記述する。

課題ごとに、コードブロックに対して、そのブロックがどのサブゴールに関するものなのかを示す属性を設定し、これを観測系列とする。他にも、ツールはコンパイル時の成否や画面切り替え状態などを保存している。本手

[†]立命館大学情報理工学部

[‡]立命館大学大学院理工学研究科

法でのツールは、ブロックを配置するさいのふるまいを収集し、ふるまいの中の特徴を分析し理解度を推定する。

3.3 理解度の推定と要因の同定

コードパズルの操作ログから学習者の思考過程を分析するため、隠れマルコフモデルにより、ある時点で見通しが立っているかを判別するモデルを作成する。本手法は、2種類の理解度推定を行う。

第一に、操作の観測系列から復号した隠れ状態から判断する、「見通しが立っているか」という理解度である。隠れマルコフモデルでは、バウム-ウェルチ・アルゴリズムによって、観測系列を与えると未知のパラメータを決定することができる。そして、遷移確率行列、観測系列が分かっているとき、ビタビ・アルゴリズムによって観測系列に対応する隠れ状態を復号することができる。観測できない「理解度」によって「ふるまい」は観測されるという仮説が正しければ、このときの隠れ状態は「見通しが立っているか」を示すような理解度を表すはずである。

第二に、ある操作を観測した時点で他の学習者とのふるまいを比較することにより得られる、「その見通しは正しいか」という理解度である。この方法では、まず収集した学習者のデータを理解度が高い者と低い者に選別する。このラベリングは課題の成果や支援ツールによる可視化からヒトが行う必要がある。次に、2つに選別したデータ群でそれぞれバウム-ウェルチアルゴリズムによるパラメータ推定を行い、隠れマルコフモデルを構築する。ここで、新規学習者から観測した系列をそれぞれのモデルで尤度関数により評価を行う。これはすなわち、観測した系列がどちらのモデルから取得できる確率が高いかを比較していることに値する。このように、尤度を用いて既存学習者と比較した場合の理解度を数量化できる。このように、理解度が高い学習者のモデルを習合してモデルを作成すれば、理想的なサブゴールの順序を明らかにすることができる。これにより、理解度の低い学習者が見通しの立っていない状態に陥った状態を検知できる。そして、理解度が高い学習者の見通しをあらかじめ明らかにしておけば、指導法も判る。

4. コードブロック課題取り組みの可視化

ふるまいを可視化する支援ツールを図2に示す。

2つのグラフは、コードパズルの成果が良かった者と良くなかった者の2例の代表である。横軸は時間軸であり、縦軸はサブゴールカテゴリである。サブゴールカテゴリを触った時間にプロットが行われる図になっている。図上部の成果が良かった者のグラフでは、一貫したサブゴールを操作していることが観測された。一方、下部の成果が良くなかった者のグラフでは、一貫性がなく、特定の時間枠にさまざまなサブゴールを操作してしまっていることが観測された。ただし、一貫した操作が行われていたとしても正しい構造になっているとは限らない。あくまで見通しを持っているかを判別できそうであることが、可視化により判った。今後は、観測の可視化により明らかになった理解度とふるまいの違いを隠れマルコフモデルによりモデル化し、自動で理解度を推定できるか実験により検証する。また、プログラムの構造の正しさがコードパズルから取得できるデータから把握できるか検証し、モデルに組み込む。

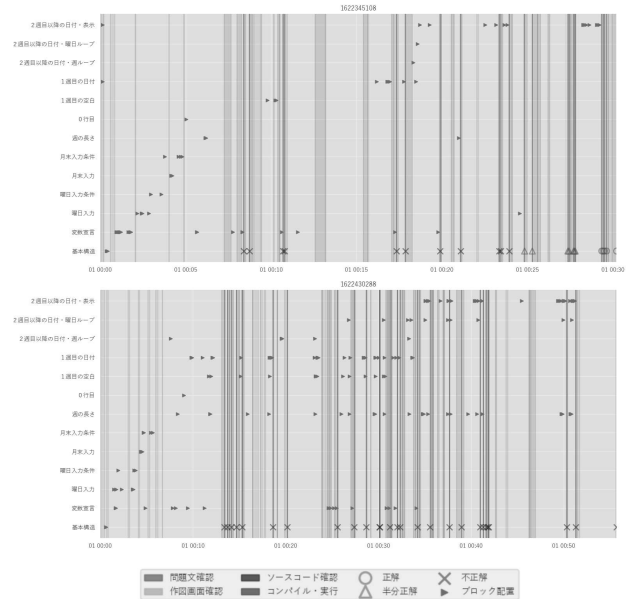


図2: カレンダー作成問題におけるふるまい可視化

5. おわりに

本論文では、プログラミング教育の現場で学習者の思考過程を考慮した理解度を推定する手法を提案した。本手法は、コードパズル課題の解答過程から、学習者に解答の見通しが立っているかで見通しの正しさを明らかにする。指導者は理解に躓いている学習者を早期に発見することができ、より適切な対策を講じることができる。

参考文献

- [1] Tyne Crow, Andrew Luxton-Reilly, and Burkhard Wuensche. Intelligent tutoring systems for programming education: A systematic review. In *Proceedings of the 20th Australasian Computing Education Conference, ACE '18*, p. 53–62, New York, NY, USA, 2018. Association for Computing Machinery.
- [2] John C Nesbit, Qing Liu Arita Liu, and Olusola O Adesope. Work in progress: Intelligent tutoring systems in computer science and software engineering education. *Proceeding 122nd Am. Soc. Eng. Education Ann*, 2015.
- [3] Dale Parsons and Patricia Haden. Parson's programming puzzles: A fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52, ACE '06*, p. 157–163, AUS, 2006. Australian Computer Society, Inc.
- [4] Y. Futamura, T. Kawai, H. Horikoshi, and M. Tsutsumi. Development of computer programs by problem analysis diagram(pad). In *Proceedings of the 5th International Conference on Software Engineering, ICSE '81*, p. 325–332. IEEE Press, 1981.