

プログラミング学習における理解度特定のための基礎検討 A Study of Identifying Comprehension Level in Programming Learning

酒井 泰地[†]
Taichi Sakai

浦野 昌一[†]
Shoichi Urano

1. はじめに

近年、学校の情報科目ではプログラミングをはじめ情報通信技術やデータ活用などの専門的な学習をするようになってきている。しかし、現状は教育者の不足が問題視されており、学習を支援するシステムの重要性が高まっている。

従来の教育支援システムは、学習前と学習後のプログラミングゲームにおけるテストスコアから適性を調査するといった解答結果の分析を行っているものが多い^[1]。

本研究では、実際の教育現場で指導者が解答過程を見て学習者に個別の指導を行っていることになり、「解答過程」に着目して個人に適応したプログラミングの教育改善を目標とする。解答過程を分析するためにプログラミングの問題解答時の視線情報を用いる。

本稿の目的は、以下の二点である。

- 1) 視線情報の時間推移を考慮した特徴抽出を行う
- 2) 視線の特徴から解答過程を分析し理解度を推定する

これまでに、プログラミング学習中に視線が集中した要素を苦手箇所として解答のヒントを出す研究などが報告されている^[2]。しかし、プログラミング問題解答においては、重要でない箇所にも視線が集中することがあるため、視線集中のみに着目するだけでは不十分であり、解答手順を考慮して分析を行う必要がある。

筆者らはこれまでの研究において、視線集中に加え、視線の時間推移を考慮した分析として自己組織化マップによる可視化を行った^[3]。また、被験者の分類を行い、誤答傾向を持つ被験者を除いた正答者(本稿では正答傾向者と呼ぶ)を基準に特徴抽出を行った。

今回は、個人の理解度に合わせたフィードバックを行うために、視線情報から理解度を特定するための基礎検討を行う。自己組織化マップを用いて得られた正答傾向者の視線特徴と同様の特徴が見られるか判定する。また、視線情報の判定に適切な条件設定を検討する。

2. 視線情報計測

視線情報は被験者の注意を引く要素を客観的に分析することができる指標の一つとして知られており、問題解決プロセスの観察や無意識に注意を引かれる要素の発見といった目的で使用される。

本研究におけるプログラミング問題解答時の視線情報計測には、Tobii eye tracker nano pro を使用した。視線情報測定の様子を図 1 に示す。この計測装置は計測画面に取り付けて使用し、赤外線を照射しカメラ画像処理と合わせて被験者の注視点を計測するセンサーである。サンプリングレートは 60Hz である。

Tobii eye tracker nano pro は計測前に被験者に画面の特定箇所を注視させることによりキャリブレーションを行う。

[†] 明治大学 Meiji University

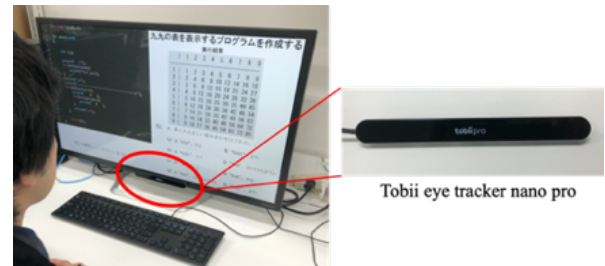


図 1 視線情報計測の様子

これにより、画面サイズや画面からの距離に合わせることに加え、被験者の注視点と計測点のずれを調整する。計測時にはキャリブレーションにより生成された角膜の赤外線反射パターンと画像処理によって注視点を計算している。

3. 分析手法

視線情報分類および個人分類の手法として K-medoids クラスタリングを適用する。また、分類したデータを可視化し特徴抽出を行うために自己組織化マップを用いる。

3.1 K-medoids クラスタリング

クラスタリングは分類対象のデータ群から類似するデータが同じクラスになるように分類するデータ解析手法の一つである。K-medoids クラスタリングは、K-means クラスタリングを一部修正したものであり、クラスタ中心をクラスタ内の全ての点との距離の総和が最小になる点とする手法である。そのため、K-means クラスタリングと比べて外れ値の影響を受けにくいという特徴がある。

3.2 自己組織化マップ

自己組織化マップは高次元の観測データを低次元に写像する教師なしニューラルネットワークの一種である。出力平面のノードと入力空間の一点を指す参照ベクトルによって構成される。各入力データに対し、最も近い参照ベクトルを持つノードを勝者ノードとし、勝者ノードと近傍ノードの参照ベクトルを学習によって更新することで、性質の近いものが隣り合うようにマッピングする。

4. 理解度特定

指導者なしで学習者の理解度を正確に把握することは個人に最適化された学習を実現するために重要である。視線情報を分析することで解答過程の傾向から理解度を特定することを考える。

4.1 本研究における理解度特定の位置付け

研究の流れを図 2 に示す。被験者ごとにキャリブレーション、問題解答時の視線情報計測を行い、計測された視線情報に K-medoids クラスタリングを適用して視線分類を行う。時間をフェーズに分割して自己組織化マップを適用し、

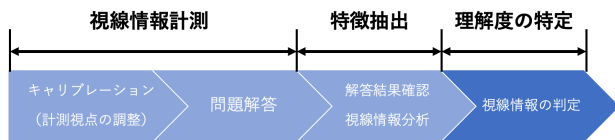


図2 研究の流れ

解答過程の特徴抽出を行う。こうして得られた特徴と同様の特徴が見られるか判定することで、理解度を推定する。

4.2 理解度推定の方法

理解度を推定するために、抽出された正答傾向および誤答傾向の特徴から注視領域と時間のチェックポイントを設定する。抽出された特徴からは注目していた領域とフェーズを把握できるため、チェックポイントと同様のフェーズ領域を注視していた場合に誤答傾向のフラグを立てることで各被験者の理解度を推定する。

注視していることを判定するために、一般的な一箇所を注視する注視時間の長さ(本稿では注視時間基準と呼ぶ)を検証する必要がある。被験者ごとに解答時間が大きく異なっていたこと、一箇所を極端に長時間注視する視線がみられなかったことを考慮して複数の注視時間基準を設定し、それぞれのフラグを確認する。

5. 実験

実験条件と実験結果を以下に示す。

5.1 実験条件

被験者は10人とする。視線移動から解答手順を把握するために、ソースコード、プログラムの実行結果、問題を同時に画面表示し解答する。

視線情報の特徴抽出と理解度推定の条件を以下に示す。

5.1.1 視線情報の特徴抽出条件

K-medoids クラスタリングの条件は以下の通りである。

クラス数:6 初期値:注目領域中心の座標

今回各クラスが意味する領域は以下の通りとする。

クラス1: ソースコード(for 文1) クラス4: 実行結果

クラス2: ソースコード(for 文2) クラス5: 解答選択肢

クラス3: 問題文 クラス6: 手元のメモ

自己組織化マップを用いて正答傾向者に共通してみられた特徴と誤答者のみにみられた特徴を抽出する。解答時間を等間隔の20フェーズに分割し、各フェーズで最も長時間連続して注視されていたクラスを抜き出し、自己組織化マップに反映させる。

自己組織化マップの条件は以下の通りである。

入力データ: 視線情報のクラス、時間のフェーズ

学習回数: 1000 近傍関数: ガウス関数

初期更新半径: 10 ノード数: 200

5.1.2 理解度推定の条件

特徴抽出に基づくチェックポイントを表1に示す。設定したチェックポイントにおいて、正答傾向者に共通してみられた特徴以外が注視時間基準以上に注視された時と、誤答者のみにみられた特徴が注視時間基準以上に注視された時に誤答傾向が検出されたとしてフラグを立てる。

注視時間基準を設定するために、各フェーズで最も長時間注視されていた領域の注視時間を抜き出して被験者全体平均、正答者平均、誤答者平均を算出し、これを共通の注

表1 チェックポイント設定

	クラス	フェーズ
正答傾向者共通	5	1, 9, 15, 16, 17, 18
誤答者のみ	1	3, 4
	2	10
	4	3, 5, 6, 8, 14, 20

表2 共通の注視時間基準で検出されたフラグ数

	注視時間基準(s)	誤答者	正答者
誤答者平均	4.46	7	0
全体平均	3.87	10	0
正答者平均	3.29	17	1

表3 注視時間基準4.46(s)のフラグ数

	誤答者				
	A	B	C	D	E
被験者					
フラグ数	4	2	0	0	1
	正答者				
	F	G	H	I	J
被験者					
フラグ数	0	0	0	0	0

視時間基準として検出を行う。また、全体時間に対する注視時間の割合平均を算出し、被験者ごとに個別の注視時間基準を設定して検出を行う。全体時間に対する注視時間の割合では、解答時間により注視時間基準が極端に大きく、もしくは小さくなることを考慮し、上限値を誤答者平均、下限値を正答者平均として設定した場合の検出を行う。

5.2 実験結果

全被験者共通の注視時間基準で検出されたフラグの数の合計を表2に示す。被験者ごと検出されたフラグの例を表3に示す。共通の注視時間基準を設定した場合、誤答者に多くの誤答傾向が検出されることがわかった。また、個別の注視時間基準を設定した場合でも同様であった。

しかし、被験者個人のフラグを確認すると、一箇所を注視する時間が極端に短く、ばらついていた誤答者に対してはフラグが立たなかったことから、視線のばらつきを考慮した検出が必要であると考えられる。

6. おわりに

本稿では、自己組織化マップを用いて得られた特徴をもとに、理解度を判別する適切な条件設定の検討を行った。

今後は、ばらつきを考慮して視線情報から明確に理解度を特定する方法を検討するとともに、特定された理解度に応じてリアルタイムでのフィードバックを行っていく予定である。

参考文献

- [1] 松本慎平, 加島智子, 山岸秀一, “プログラミング学習前に行われたプログラミングゲームの理解度とその学習後の到達度との関係分析”, 情報教育, Vol.2, pp.31-40 (2020)
- [2] Jerry Chih-Yuan Sun, Kelly Yi-Chuan Hsu, “A smart eye-tracking feedback scaffolding approach to improving student’ learning self-efficacy and performance in a C programming course”, Computers in Human Behavior, Vol.95, pp.66-72, (2019).
- [3] 酒井 泰地, 浦野 昌一, “視線情報の傾向分析による特徴抽出”, 人工知能学会全国大会, 4G1-GS-2j-04(2021).