

取捨選択操作の時間的な共起分析による プログラミング・プロセスでの迷いの検出

山口 琢¹ 松澤 芳昭² 新美 礼彦³ 大場 みち子³

概要：プログラミングのプロセスの学習分析では、学生が「何かに迷っている」ことを検出できる研究が増えてきたが、具体的に「何に迷っているか」を機械的に検出するのが困難である。われわれは、取捨選択・並べ替え型のプログラミング・パズルで、取捨選択操作の時間的な共起分析によって、学生の迷いを検出する手法を提案する。提案手法は、学生がどちらを選ぶか迷っている可能性が高い、ピースのペアを具体的に提示する。問題作成者および教師による判断を正解として、提案手法で検出した迷いを評価したところ、提案手法の適合率が44%、再現率が92%となり、提案手法による提示が問題作成者および教師による判断をほぼ含む結果となった。このとき、ピースの全ての組み合わせに対して、実際に迷われた組み合わせの割合は2%であった。取捨選択操作の時間的な共起は、問題作成者および教師に対して、学習者の具体的な迷いを提示する手法として有効であることを示した。

キーワード：迷いの検出、取捨選択、並べ替えプログラミング・パズル、短冊形、学習分析

Detecting Hesitation in a Programming Process by Analyzing Temporal Co-occurrence of Selection and Discard Operations

TAKU YAMAGUCHI¹ YOSHIAKI MATSUZAWA² AYAHIKO NIIMI³ MICHIKO OBA³

1. はじめに

論理的思考力への関心の高まりやプログラミング学習の広がりに伴って、学習分析 (Learning Analytics) 研究では近年、プログラミングのプロセスを測定・分析する研究が増えている。

プログラミングのプロセスの学習分析では、学生が「何かに迷っている」ことを検出できる研究が増えてきたが、具体的に「何に迷っているか」を機械的に検出するのが困難である。

2. 先行研究

森永らは、レーベンシュタイン距離を修正した距離を考案し、その距離を使って問題を解くプロセスを分析すること提案した。その中で「…(解答者が) 仮説的な考えを試しながら『建設的な試行錯誤』でアルゴリズムを設計せずパターンで回答しようとしていた過程が時系列変化に現れる」可能性を指摘している [2][3]。

Maharjan らはプログラム行を取捨選択・並べ替えて完成させる Parsons Puzzles について、「正解との編集距離は、解答の正しさの度合いを示す」として、レーベンシュタイン距離を修正した編集距離に基づく分析手法を提案した。そして「正解との編集距離の列が非単調で頻繁に上下に振れるのは、試行錯誤を示唆している」と述べている [4]。

¹ フリー

Independent Researcher

² 青山学院大学 社会情報学部 社会情報学科

School of Social Informatics, Aoyama Gakuin University

³ 公立はこだて未来大学システム情報科学部

Faculty of Systems Information Science, Future University
Hakodate

オレンジで囲まれた 選択肢群から選んで、青で囲まれた 部分に問題の解答を作成する。

問題

Aに水の量Bに食塩の量を入力する事で食塩水の濃度はoo%ですと出力されるプログラムを作成してください。
なお濃度を出す際は、数字を切り捨ててにして表示されるようにしてください。

```

answer= Math.ceil(answer);
var answer= B/(A+B)*100;
var answer=(A+B)/A/100;
answer =Math.floor(answer);
var A,B=input('水の量は何リットル?');input('食塩の量は何グラム?');
answer= Math.ceil(answer);
ここにドロップして選択をキャンセル

var A=input('水の量は何リットル?');
var B=input('食塩の量は何グラム?');
var answer=(A+B)/B*100;
answer =Math.floor(answer);
println('食塩水の濃度は'+answer+'%です');
    
```

完成!

図 1 取捨選択・並べ替えプログラミング・パズルのジグソー・コード
パズル問題「濃度判定プログラム」

3. これまでの取り組み

3.1 ジグソー・コードと操作の測定

ジグソー・コードは並べ替えプログラミングのアプリケーションである [1]。図 1 は、取捨選択型のジグソー・コード 2 の画面である。左側のオレンジで囲まれた枠内の最初の並び順は、解答を始めるたびにシャッフルされている。解答者は、左側の枠から、適切なピースを選んで、右側の青で囲まれた枠にドラッグ&ドロップで移動する。後で不要と分かったピースは、右から左にドラッグ&ドロップで戻す。左右どちらの枠内でも、ピースをドラッグ&ドロップで並べ替えることができる。これらの操作を繰り返して、ピースを右側の枠内に適切な順序で並べてプログラムを完成させる。短冊形と呼ばれるアプリケーションと似ており、ピースは短冊と呼ばれるものに対応する。

ジグソー・コードは、ユーザの並べ替え操作を測定している。左の枠から右の枠へピースを移動して選ぶ、右の枠から左の枠へピースを移動して捨てる、それぞれの枠内でピースを移動して並べ替える、これらすべての操作について、操作対象のピース ID や操作時刻などを記録している。

3.2 操作の時間的な共起分析

ジグソー・コードが測定した操作ログの分析では、操作過程のパターンを見つけるために、操作対象となったピースの時間的な共起行列を使うことが多い。この共起行列は、あるピースを動かした次にどのピースを動かしたかの回数を集計する [5]。続けて動かされる回数が多いとき、それら 2 つのピース間には何か特別な関係があると考えられるが時間的な共起分析の動機である。この発想は、テキスト分析においてテキストの空間的な位置の共起を分析する、

n \ n+1	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12
s3	16	91	11	45	31	7	1	8	14	9
s4	11	24	58	36	7	15	4	11	9	27
s5	6	11	32	71	6	12	1	7	3	4
s6	9	17	28	32	15	12	3	14	5	26
s7	5	27	21	8	9	1	1	7	10	12
s8	2	6	7	15	1	15	2	0	2	6
s9	2	1	4	5	2	1	5	0	0	3
s10	44	12	3	10	8	2	2	32	3	2
s11	3	9	2	6	13	4	4	1	12	4
s12	5	6	29	27	8	2	3	5	1	10

図 2 並べ替え操作の共起行列

従来の共起分析と同様である。図 2 は、図 1 のパズル (問題) を 223 人の学生が解いたときの、操作ログを集計した時間的な共起行列である。行や列の見出しにある s3 や s4 は「濃度判定プログラム」に含まれるピースの内部的な ID である。図では、s3 行の次に s4 行を動かした回数が全部で 91 回あったことが分かる。

セルの値が平均+標準偏差より大きいセルを黄色 (薄い背景色)、平均+標準偏差*2 より大きいセルを赤 (濃い背景色に白い文字) で強調している。図 2 では、s3 の次に s4 を動かした回数の「91」が比較的多いことが分かる。図 3 は図 1 の正解と、各ピースの ID を示している。

3.3 最後に操作したピースに迷いを推定

中村らはジグソー・コードの操作ログを、最後に動かされた回数が多いピースに着目して分析した。それらが最後の動かされることが多いのは、それらについて迷いが生じ

- s1 Aに水の量Bに食塩の量を入力する事で食塩水の濃度はoo%ですと出力されるプログラムを作成してください。
 s2 なお濃度を出す際は、数字を切り捨ててして表示されるようにしてください。

s7	var answer=(A+B)/B*100;	(なし)	
s8	answer= Math.ceil(answer);	s3	var A=input('水の量は何リットル?'); var B=input('食塩の量は何グラム?');
s9	answer= Math.ceil(answer);	s4	var answer= B/(A+B)*100;
s10	var A,B=input('水の量は何リットル?');input('食塩の量は何グラム?');	s5	answer =Math.floor(answer);
s11	var answer=(A+B)/A*100;	s6	println('食塩水の濃度は'+answer+'%です');
s12	answer =Math.floor(answer);	(なし)	

図 3 パズル「濃度判定プログラム」のピース ID と正解

n \ n	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12
s3	9	34	5	24	20	3	3	25	10	5
s4	5	39	7	17	24	11	4	9	5	13
s5	1	3	29	19	3	9	0	2	1	0
s6	5	5	8	15	8	9	2	7	4	15
s7	4	15	13	8	4	1	2	6	3	8
s8	1	3	2	12	1	8	1	0	2	3
s9	2	1	2	3	2	1	3	1	0	2
s10	20	8	1	6	6	2	0	14	3	2
s11	3	2	3	3	8	2	3	1	7	4
s12	1	3	7	19	7	0	1	3	1	5

図 4 迷いに対応するセルの背景を赤くした共起行列
 s4 と s7 の交点のセルの背景が赤くなっている

たためと推定した [1]。

3.4 高い共起頻度を迷いと判定

藤井らはジグソー・コードを使ったプログラミング問題を使って、正解者・不正解者らの思考過程の特徴を分析して、2つのピースの取捨選択という具体的な迷いを検出した [6]。この研究では、分離融合系学部の1年生約230名を対象に、JavaScript やタートルを使ってプログラミングの初歩を学ぶ講義で、ジグソー・コードを使った小テストを行った。1度に2~3問、合計13問を、新たに開発して出題した。

得られたデータの分析では時間的な共起行列 (3.2 節) を使ったが、独自の工夫として、「解答者が類似するピースに迷って、続けて動かしたための共起」と藤井らが判断したセルは、背景を赤く着色した。この手法では、例えば図1の問題を解いた不正解者の共起行列は図4のようになる。

図4では、s4の次にs7を動かした共起頻度が24と高

く、これは迷いのためと判断されて赤い背景となっている。具体的には: 「不正解者は、

```
var answer= B/(A+B)*100;
```

と

```
var answer=(A+B)/B*100;
```

とで迷っている」といった判断に対応している。図1のパズルを見れば、これらのどちらを選ぶか迷ったという推定は妥当と考えられる。

3.5 課題

前節3.4の研究では、高い共起頻度を示すピースの組み合わせから、人の判断で、解答者の迷いを読み取った。この判断を機械的にできれば、より早く解答者の迷いの様子を具体的に把握できる。

4. 目的

本研究の目的は、取捨選択・並べ替え型のプログラミング・パズルで、取捨選択の操作に限って時間的な共起分析を行い、解答者の「取捨選択の迷い」を機械的に検出できるか検証することである。この迷い検出結果の使い途は、学生が迷った可能性のある組み合わせとして、いち早く教師などに提示するためのものとする。

実際に迷った組み合わせの数 (割合) は少ないと予想して、提案方式による迷い検出の適合率 (precision, 精度) よりも再現率 (recall) を重視する。すなわち、網羅性を偽陽性 (False Positive) よりも重視する。実際に迷った組み合わせの割合を、本稿では「迷い率」と呼ぶことにする。本研究では、迷い率が実際に小さいことも確認する。

本稿では、すべての解答が済んでパズル操作が完了した段階で、全操作データを対象に分析する。将来的には、正解者・不正解者という結果が出る前の、テストの途中で迷いを推定し教師などに通知することを目標とする。そのときには、途中段階までのデータや、それらの時間変化が分析の対象となる。

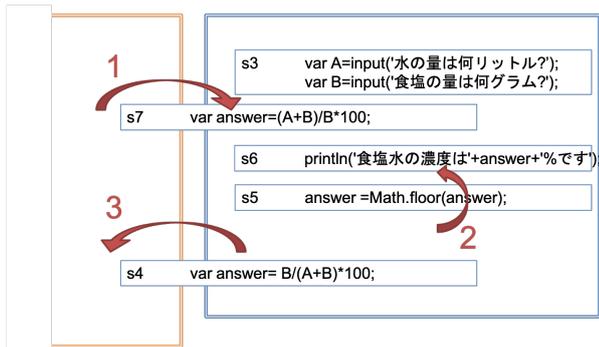


図 5 取捨選択操作の集計

	s4		s6	s7	
s4					
s6	+1				
s7			+1		

	s4		s6	s7	
s4				+1	
s6					
s7					

図 6 左が従来の共起行列、右が提案方式の共起行列

5. アプローチ

提案手法は、学生がどちらを選ぶか迷っている可能性が高い、ピースのペアを具体的に提示する。

5.1 提案手法: 取捨選択操作の共起分析

提案手法では、3.2 節の共起分析と異なり、左の枠から右の枠へピースを選ぶ操作と、右の枠から左の枠へピースを捨てる操作とのみ、すなわち取捨選択操作のみについて共起を集計する。2つの取捨選択操作の間に、右の枠内での並べ替え操作が入る場合があるが、そのような操作があったとしても2つの操作を共起としてカウントする。また、図2と異なり、2つの操作の順序を問わずに集計する。

図5で、まず、s7を選択し、続いてs6を枠内で上に移動し、s4を捨てたとする。3.2節の共起分析ではs7とs6、s6とs4が共起するが、s7とs4は共起とみなさない。提案手法では、s7とs4が共起したとみなす。図5のように操作された場合、共起行列は、従来方式では図6の左側のようにs6行s4列とs7行s6列のセルに1を加える。提案方式では図の右側のように、s4行s7列のセルに1を加える。

図7は図2と同じ操作ログについて集計した、取捨選択の共起行列である。どちらを先に選んだか順序を問わないので、共起行列の右上に集計されている。

対角線上のセルは、ある同じ1つのピースを選択した後に捨てた、あるいは捨てた後に選択し直した回数である。これは、「ある1つのピースを選ぶか選ばないか迷っている」などと解釈されることになるだろう。

n \ n+1	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12
s3	0	2	1	2	2	4	0	35	0	2
s4	-	3	1	0	16	2	0	1	8	3
s5	-	-	2	2	2	7	3	3	0	20
s6	-	-	-	0	1	0	1	2	1	1
s7	-	-	-	-	3	0	0	2	5	2
s8	-	-	-	-	-	3	0	0	1	2
s9	-	-	-	-	-	-	0	0	0	0
s10	-	-	-	-	-	-	-	3	0	1
s11	-	-	-	-	-	-	-	-	2	0
s12	-	-	-	-	-	-	-	-	-	1

図 7 取捨選択の時間的な共起共起

5.2 検証方法

提案手法の提示する迷いの組み合わせと、問題作成者および教師が推定した迷いの組み合わせとを照合して、提案手法の適合率と再現率を計算する。また、迷い率が小さいことを確認する。

問題作成者および教師による推定については、全ての問題に対して3.4のような推定が卒業論文にまとめられている。本研究では、提案手法の提示と卒業論文の記述とを照合する。

取捨選択の共起分析にあたって、正解者・不正解者を分けず全体を対象とする。これは、将来的には、テストの途中で学生の迷いを推定することを目指すからである。テストの途中では正解者・不正解者は確定しない。

一方、問題作成者および教師による推定は、正解者と不正解者で別々に共起行列を作成し、共起頻度の高い組み合わせについて、それが迷いによるものか推定している。

6. 結果

問題作成者および教師による判断に対して提案手法の適合率が44%、再現率が92%となり、提案手法による提示が問題作成者および教師による判断をほぼ含む結果となった。

表1は、提案方式による検出と正解、すなわち問題作成者および教師による判断を比較した表である。「s4-s9」などは迷いの組み合わせを示し、「s4にするかs9にするか迷った」ことを表す。

表2で、TPはTrue Positiveで、提案方式による検出が正解だった数である。FPはFalse Positive、FNはFalse Negative、TNはTrue Negativeである。ピース数は動かせるピースの数である。迷い候補数は、2つのピースの間でどちらを選ぶか迷う場合の数に、ピース自身を選ぶかどうか迷う場合の数を加えた数である。すなわち、動かせるピースから2つを選ぶ組み合わせ数に、ピース数を加えた数である:

表 1 結果

問題番号	卒論の問題名	提案方式による検出	正解
1-1	砂時計の容器を作るプログラム	s4-s9, s5-s9	s4-s9
1-2	濃度判定プログラム	s3-s10, s5-s12, s4-s7	s3-s10, s4-s7
1-3	整数入れ替えプログラム	s3-s9, s7-s12, s5-s11, s8-s13	s3-s9, s7-s12
2-1	PCR 検査陽性率	s4-s13, s8-s14	s8-s14
2-2	ひまわりを描くプログラム	s5-s8, s3-s10, s3-s11	なし
3-2	二乗を計算するプログラム	s4-s8, s2-s9	s4-s8
3-3	九九出力プログラム	s2-s5, s4-s8	s4-s8
4-1	ドーナツを描くプログラム	s4-s9, s5-s11	s4-s9
4-2	閏年判定プログラム	s3-s7	s3-s7
5-1	GO TO travel の料金判定プログラム	s5-s8, s4-s7, s6-s8	s4-s7
5-2	GO TO travel の料金判定プログラム 2	s10-s12, s4-s11, s4-s14	s4-s11, s9-s13

表 2 結果 2

問題番号	卒論の問題名	TP	FP	FN	TN	ピース数	迷い候補数	迷い率
1-1	砂時計の容器を作るプログラム	1	1	0	64	11	66	0.02
1-2	濃度判定プログラム	2	1	0	52	10	55	0.04
1-3	整数入れ替えプログラム	2	2	0	74	12	78	0.03
2-1	PCR 検査陽性率	1	1	0	89	13	91	0.01
2-2	ひまわりを描くプログラム	0	3	0	42	9	45	0.00
3-2	二乗を計算するプログラム	1	1	0	53	10	55	0.02
3-3	九九出力プログラム	1	1	0	26	7	28	0.04
4-1	ドーナツを描くプログラム	1	1	0	43	9	45	0.02
4-2	閏年判定プログラム	1	0	0	27	7	28	0.04
5-1	GO TO travel の料金判定プログラム	1	2	0	25	7	28	0.04
5-2	GO TO travel の料金判定プログラム 2	1	2	1	75	12	78	0.01

表 3 適合率と再現率

適合率	44%
再現率	92%
迷い率	2%

迷い候補数 = $nC_2 + n$, n : 動かせるピース数

TN は迷い候補数から TP、FP、FN の合計を引いたものである:

$$TN = \text{迷い候補数} - TP - FP - FN$$

迷い率は TP を迷い候補数で割ったもので、迷う可能性のあった全組み合わせから、実際に迷った組の割合である。

表 2 から適合率と再現率を計算すると表 3 となる。

7. 考察

再現率が 92% となり、実際、表 1 では、問題番号「5-2」以外の問題で、提案方式による検出が正解を含んでいる。

提案方式による検出のうち、False Positive にあたるものでも、パズルを具体的に検討してみると、外れとも言い切れないものであった。例えば、問題番号「1-2」の「濃度判定プログラム」は図 1 および図 3 のことである。提案方式の False Positive にあたる「s5-s12」は、

[s5] answer = Math.floor(answer);

と

[s12] answer = Math.floor(answer):

との間の迷いに該当する。s5 と s12 との違いは行末のセミコロン「;」とコロン「:」の違いで、両者はよく似ている。この両者の間で取捨選択の時間的な共起頻度が高かったならば、両者のどちらを採用するか解答者が迷ったという解釈は妥当であろう。

「正解とした迷い組み合わせは従来の時間的な共起行列を出発点に推定したものであり、提案手法も時間的な共起関係に基づくならば、この結果は自明なのではないか？」との疑問が湧く。しかし、5.1 節と図 5 および図 6 で示したように、2 つの共起行列はカウントしている組み合わせが異なる。

以上から、取捨選択操作の時間的な共起は、問題作成者および教師に対して、学習者の具体的な迷いを提示する手法として有効である。

8. おわりに

8.1 まとめ

取捨選択操作に対する時間的な共起分析によって、取捨選択・並べ替え方式のプログラミング・パズルにおける、解答者の迷いを具体的に検出できた。提案方式の再現率は 92%、問題群の迷い率は 2% であった。提案方式は、解答者の迷いを具体的に検出する方式として有効である。

n \ n+1	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12
s3	0	1	0	1	1	3	0	20	0	2
s4	-	2	1	0	9	1	0	0	6	3
s5	-	-	0	2	2	5	2	2	0	17
s6	-	-	-	0	1	0	1	1	1	1
s7	-	-	-	-	1	0	0	0	3	0
s8	-	-	-	-	-	1	0	0	1	1
s9	-	-	-	-	-	-	0	0	0	0
s10	-	-	-	-	-	-	-	2	0	1
s11	-	-	-	-	-	-	-	-	2	0
s12	-	-	-	-	-	-	-	-	-	1

図 8 正解者の取捨選択の共起行列

n \ n+1	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12
s3	0	1	1	1	1	1	0	15	0	0
s4	-	1	0	0	7	1	0	1	2	0
s5	-	-	2	0	0	2	1	1	0	3
s6	-	-	-	0	0	0	0	1	0	0
s7	-	-	-	-	2	0	0	2	2	2
s8	-	-	-	-	-	2	0	0	0	1
s9	-	-	-	-	-	-	0	0	0	0
s10	-	-	-	-	-	-	-	1	0	0
s11	-	-	-	-	-	-	-	-	0	0
s12	-	-	-	-	-	-	-	-	-	0

図 9 不正解者の取捨選択の共起行列

8.2 残された課題

本稿では事後的に迷いを検出したが、解答者たちが小テストを解いている最中に、一定の時間が経過したところで「続々と迷いつつある」ことを検出できるだろう。最後まで待たなくても、問題作成者の狙い通りに解答者たちが迷ったのか、迷わなかったのかが判れば、小テスト後のコメントを変える時間的な余裕が生まれる。このような分析は今後の課題としたい。

誤り率 2%は小さい。問題作成者によれば、これらは意図してパズルに仕込んだ迷い組み合わせである。そして、意図したとおりに、たった 2%の迷い候補に学生が迷った。このことから、問題作成者と解答者との間に、一定の共通理解があったと言えるのではないか。また、問題作成者の狙いがよく当たった、良い問題だったと言えるのではないか。このような観点からの、提案方式による問題(パズル)そのものの評価は、今後の課題としたい。

本稿は学生の迷いについて中立である: 迷うことが良いとも悪いとも前提としていない。本稿では、迷いを正解/不正解の観点から分析しなかった。図 8 は、図 1 の正解者についての提案方式による共起行列、図 9 は不正解者についてのものである。これらを見ると、正解者も不正解者も、どちらも同じ組み合わせについて迷っていたことが分かる。それと同時に、解答者の人数が約 230 人であったことを考えると、これらの組み合わせで迷わなかった解答者も、正解者/不正解のそれぞれにいたことも分かる。前の段落で指摘した「一定の共通理解」を踏まえると、このことは、正解/不正解とは別の評価指標が解答プロセスに存在する可能性を示唆している。

謝辞 藤井沙苗氏には、ジグソー・コードのパズル問題、実験データおよび卒業論文を提供していただいた。本研究は JSPS 科研費 20H01728 の助成を受けたものである。

参考文献

- [1] 中村 陽太, 大場 みち子, 山口 琢, 伊藤 恵, 学習進度に対応するパズルを利用したプログラミング思考過程の分析, 情報処理学会研究報告コンピュータと教育 (CE), 2019-CE-151(1), 2019-09-28
<http://id.nii.ac.jp/1001/00199559/>
- [2] 森永 笑子, 松本 慎平, 村上 瑠香, 林 雄介, 平嶋 宗, カード操作方式によるプログラミング学習支援システムでの学習過程の可視化方法の提案, 人工知能学会 先進的学習科学と工学研究会 85, 92-97, 2019-03-07
<http://id.nii.ac.jp/1004/00009654/>
- [3] 森永 笑子, 松本 慎平, 林 雄介, 平嶋 宗, カード操作方式によるプログラミング学習支援システムにおける学習者の活動に基づく学習課題の特徴分析, 電気学会 C 部門 情報システム研究会 (CIS), 2019-11-10
<http://id.nii.ac.jp/1031/00126864/>
- [4] Salil Maharjan and Amruth Kumar, Using Edit Distance Trails to Analyze Path Solutions of Parsons Puzzles, Educational Data Mining 2020 (EDM 2020), 2020-07-10
https://educationaldatamining.org/files/conferences/EDM2020/papers/paper_163.pdf
- [5] 山口 琢, 大場 みち子, 編集操作の時間的共起分析の提案, 情報処理学会 研究報告コンピュータと教育 (CE), 2019-CE-151(9), 2019
<http://id.nii.ac.jp/1001/00199567/>
- [6] 藤井沙苗, 松澤芳昭, パズル型問題を利用したプログラミング初学者の理解度と思考過程の分析, 情報処理学会 第 83 回 全国大会, 2021