

プログラミング穴埋め問題における 穴抜きの難易度と学生の解答過程のクラスタ分析

中山陽平^{†1} 掛下哲郎^{†1}

概要: 我々は効果的なプログラミング教育のための支援ツールとして pgtracer を開発している。pgtracer は、プログラムとトレース表から構成される穴埋め問題を出題し、学生の答案を自動的に採点する。また、学生の解答データとその採点結果をログデータとして収集する。本研究では統計解析用言語 R を用いて、pgtracer で収集したログデータを分析する。問題と穴埋めの難易度を評価し、学生の理解度を把握することでラーニングアナリティクスを実現する。穴埋めにかかる解答所要時間と正解率の相関分析を行った結果、二つの変数の間にはある程度の相関はあるが単純に学生の能力や穴抜きの評価をする指標としては不十分であることを示し、解答所要時間と正解のトークン数との関係を分析すると、単純な穴埋めを除いて、トークン数が解答所要時間に必ずしも影響しないことを示した。ログデータを穴抜きと学生の2つの観点で分類した結果、解答所要時間で有意差がみられた。また、分類から具体的な穴抜きの考察をした結果、単純な要素で解答できる穴抜き、解答にアルゴリズムやトレース表といった複数要素の理解を必要とする穴抜きが確認できた。学生の具体的な解答過程を調べたところ、分類毎に問題の理解に時間を費やさない学生、理解の有無にかかわらず問題に積極的に取り組む学生、正解率が幅広く分布した一般的な学生が確認できた。

キーワード: 統計解析用言語 R、穴埋め問題、プログラミング教育、ラーニングアナリティクス、単回帰分析、階層的クラスタ分析、t 検定

Cluster Analysis of Difficulty of Blanks and Student's Learning Process in Fill-in-the-Blank Programming Questions

YOHEI NAKAYAMA^{†1} TETSURO KAKESHITA^{†1}

Abstract: We are developing a programming education support tool pgtracer utilizing fill-in-the-blank questions composed of a program and a trace table representing execution of the program. Pgtracer automatically evaluates student answers and collect learning log of the students. In this paper, we analyze the collected log using statistical language R. Relationships are analyzed between the right answer ratio and the required time to fill the blanks. As a result, we find that the required time and the right answer ratio are almost uncorrelated. We also find that the number of tokens within a blank does not always affect the required time except for a simple blank. By analyzing the log data in terms of blank and student, we find significant differences in the time required for answering. The blanks with short required time can be answered with a simple element, while other blanks requires understanding of multiple elements of problems such as algorithm and trace table. As a result of investigating concrete answer process of the students, we find three clusters of the students: students who do not spend much time to understand the problem, students actively and/or repeatedly working on the problem, general students whose right answer ratio is widely distributed.

Keywords: Statistical Programming Language R, Fill-in-the-Blank Problem, Programming Education, Learning Analytics, Single Regression Analysis, Hierarchical Clustering, t Test

1. はじめに

プログラミング教育は理工系の大学や高専において重要性が高いが、学生の学力低下に関する懸念や、プログラミング自習時に教員や TA 等だけでは十分な指導が行えない等の課題がある。そこで我々は穴埋め問題を用いたプログラミング教育支援ツール pgtracer[1-3]を開発している。

pgtracer には問題出題時にログを収集する機能がある。2016 年度の研究[4]では収集したログを分析する追加機能として、学生の利用状況一覧機能、問題の平均点・平均所要時間表示機能、問題分析機能、解答過程の分析機能を含む7つの機能を開発した。また、評価実験を通じて機能の有用性が示された。

ログデータの分析においては穴埋め問題を構成する穴抜きを分類し、解答所要時間を用いて穴抜きの難易度の評価が行われていた。また、学生全体の受験状況や正解率から苦手箇所の特長、穴抜きの種類毎の特長が行われていた。分析に用いられたツールは Microsoft Excel と Access であり、学習ログデータに対して、集計などの単純な操作のみで分析が行われていた。

本研究では pgtracer で穴埋めの難易度を評価し、学生の理解度を把握することで出題するプログラミングの穴埋め問題の作成時に役立つデータを取得することを目的としている。また、学習データの分析に統計解析用言語 R[5]を用いている。R では主成分分析やクラスタ分析、回帰分析な

^{†1} 佐賀大学
Saga University

ど 20 種類ほどの高度な分析方法が適応でき、Learning Analytics を実現できる。また、無料で利用できるデータ解析環境として大学だけでなく、企業にも普及しつつある。

分析対象のデータは pgtracer を用いて収集した佐賀大学知能情報システム学科の 2 年次専門必修科目「データ構造とアルゴリズム」2016 年度受講者 74 名分の学習データを用いている。分析では穴抜きごとに解答所要時間と正解率の相関分析、および解答所要時間の分布の可視化を行った。また、ログデータに対するクラスタリングを行うことで穴抜きおよび学生のカテゴリを行った。学生毎のカテゴリに対しては、クラスター毎の理解度や理解過程の違いを分析した。また、穴抜き毎のカテゴリに対しては、穴抜きの難易度（正解率および解答所要時間）の評価をクラスター毎に行っている。

本論文では、2 節でプログラミング教育支援ツール pgtracer について紹介し、穴埋め問題の学習ログデータの収集機能とログデータを記録するテーブルの説明を行う。3 節では学習ログデータの中で解答所要時間と正解率の相関関係を分析する。4 節では個別の穴抜きについて解答所要時間の分布状況を分析する。

5 節および 6 節ではそれぞれ、穴抜きと学生という観点からデータを分類し、得られた各クラスターの特徴について考察する。7 節では学生の解答プロセスをクラスター毎に比較する。

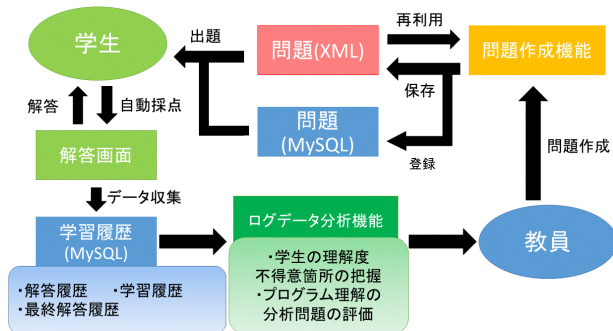


図 1: pgtracer の教育概念図

2. プログラミング教育支援ツール pgtracer

2.1 pgtracer によるプログラミング教育

プログラミング教育支援ツール pgtracer を用いた教育プロセスを図 1 に示す。pgtracer はプログラミング初学者を主な対象とした演習や補習を行うためのツールとして開発を行っている。そのため、プログラムを 1 から記述させるといった出題形式ではなく、穴埋め形式の問題を採用している。プログラムとトレース表に対する穴埋め問題を学生に出題し、学生が解答した際の点数や開始時刻、終了時刻、穴埋めする際の答案や所要時間、最終的な答案や正誤などをログデータとして収集する。教員は収集したログデータを分析し、個々の学生や全体の理解度・不得意箇所を把握し、穴埋め問題の難易度を評価する。また、分析結果をもとに、理解度が低い箇所や不得意箇所を重点的に教育でき

るように問題を再作成する。このサイクルを繰り返すことにより、プログラミング教育を継続的に改善できる。

2.2 問題出題機能

問題の登録・編集機能で登録された問題は、MySQL テーブルの情報をもとにテーマ、タイトル、難易度毎に一覧表示される。学生はテーマ、タイトル、難易度から自分の学力に合った、もしくは教員に指示された問題を選択することで解答画面を表示できる (図 2)。学生が解答画面で穴抜き個所に答案を入力したら、「解答終了」ボタンを押すことで pgtracer は答案を自動採点する。それと同時に pgtracer はログデータを自動収集する。

解答画面 試験モード

注意事項: ①プログラム内の「」は半角スペースを表しています。
 ②トレース表内の出力値において改行を表したい場合は\nを入力して下さい。
 ③画面サイズが足りず問題が見られる場合は、ページのズームを縮小して下さい

プログラム		ルーチン	step	main a	main b	main c	出力
1 2 3 4 5 6	# include <iostream >	main	1	?	?	?	
	using namespace std;	main	2	?	?	?	
	int main () {	main	3	2	3	?	
	int a, b, c;	main	4	2	3	5	
	a = 2 ;	main	5	2	3	5	5
	b = 3 ;	main	6	2	3	5	
c = b+a ;							
cout << c << endl;							
return 0;							
}							

解答終了

図 2: pgtracer の解答画面

2.3 データテーブルの構成

pgtracer が収集しているログデータを格納するテーブルの構成を以下に示す。

pgtracer_question: 出題する問題を登録しているテーブル

- 問題 ID
- 問題のテーマ
- 学習内容
- 難易度 (整数値により教員が適宜定義する)
- 配点

pgtracer_study_log: 学習履歴を保持するテーブル。問題への一回の答案が 1 レコードに対応する。

- 学習履歴 ID
- ユーザーID (Moodle のユーザーテーブルを参照)
- 問題 ID (pgtracer_question を参照)
- 学習開始時刻
- 学習終了時刻
- 採点結果

pgtracer_answer_log: 学生の解答履歴を保持するテーブル。穴抜きに対する個別の入力が 1 レコードに対応する。

- 解答履歴 ID
- 学習履歴 ID (pgtracer_study_log を参照)
- 穴抜きの場所
- 学生の答案

- 正解文字列
- 正誤判定結果
- 解答終了時刻

pgtracer_student_answer : 学生の最終的な答案を保持するテーブル。格納するデータは **pgtracer_answer_log** と同様である。

2.4 学習ログデータ収集機能

学生が問題に解答する際、ログデータの収集が行われる。学生が問題を選択すると、**pgtracer_study_log** に解答開始時刻や問題などのレコードが保持される。学生が1つの穴抜きに解答すると **pgtracer_answer_log** にレコードが追加される。また、穴抜きの終了時刻が、直前に追加したレコードに追記される。学生が解答終了ボタンを押すと **pgtracer_student_answer** に問題の終了時刻や正誤判定結果などの最終解答データが追加される。ログデータを収集するか否かは教員が設定できる。

2.5 登録されている問題

現在 **pgtracer** には出題できる問題として7つのプログラムが用意されている。数表1や数表2、時刻変換は、学生が **pgtracer** に慣れるための練習問題として登録されている。それを除く4つのプログラムが成績評価の対象となる。成績評価の対象となる問題のアルゴリズムは授業で説明されており、学生は **pgtracer** を用いた演習を通じて、アルゴリズムとプログラムの相互関係を理解する。

表1：登録されている問題

問題	初回平均点	初回平均所要時間
時刻変換	95.68 点	191.25 秒
数表1	92.70 点	136.59 秒
数表2	74.39 点	423.15 秒
二分探索	54.78 点	921.15 秒
ハッシュ関数	56.70 点	2702.77 秒
ヒープソート	42.69 点	2784.25 秒
括弧の対応検査	49.90 点	2324.98 秒

3. 解答所要時間と正解率の相関分析

3.1 分析の目的

教員は問題を作成する際、効率的に解答者にプログラミング教育を行う必要がある。また、問題には短い解答所要時間で正解率が高い問題や、解答所要時間はかかるが正解率が高いといったものがあり、出題者の意図した問題を作成する上で解答所要時間と正解率は重要な指標となる。そこで本研究では解答所要時間と正解率の相関関係を調べ、プログラミング教育を行う際に有効な穴埋め問題を作成することを目的としている。

また、前提条件として学習者は佐賀大学の知能情報システム学科に所属しているため、プログラミングに対する最低限の理解を有しているものとする。

対象データに対する評価の視点

- 穴抜き毎の正解率
- 穴抜き毎の解答所要時間

穴抜きの内訳：穴抜きの総数 91 個

- トークン（キーワードや演算子を表す文字列の最小単位）単体を正解とするもの：30 個
- 複数トークンを正解とするもの：28 個
- 文を正解とするもの：33 個

分析方法

本研究では問題の1つの穴抜きの正解率と所要時間との関係を調べるために2変数間の相関関係や予測に適した分析方法として単回帰分析を用いる。

また、正解の穴抜きの構造による違いにより、解答所要時間と正解の違いを見るために、正解の種類に応じて穴抜きを上記の3つに分類した。分類毎に解答所要時間と正解率が比例するように変化するのはないかとの推測のもとで分析を行っている。

3.2 分析結果

図3ではx軸を正解率、y軸を解答所要時間とし、分類毎に番号を付けたデータを散布図で示している。今回得られた結果ではトークン単体が図右下に多く表れ、複数トークンと文の点は図全体にばらばらに分布していた。

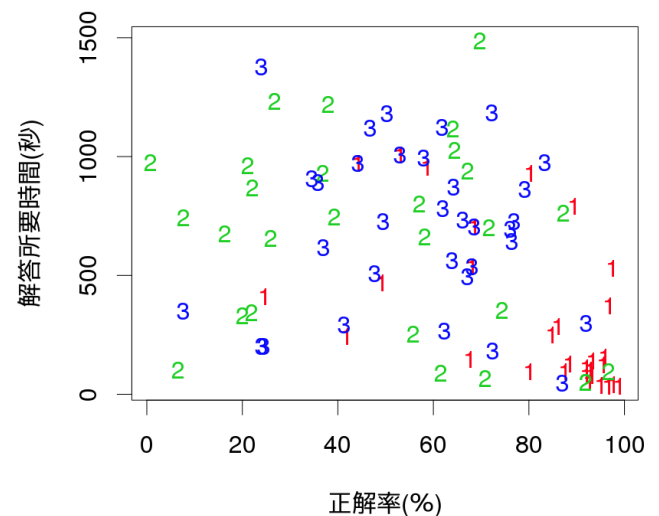


図3：穴抜き毎の回帰分析

表2：分類毎のt検定の結果

番号	分類	相関係数	p 値	決定係数
1	トークン単体	-0.547	0.002	0.299
2	複数トークン	-0.261	0.179	0.068
3	文	-0.050	0.783	-0.002
全体	なし	-0.407	6e-05	0.166

表2に示す統計データからもトークン単体を正解とする穴抜きが最も高い相関があり、決定係数も他の穴抜きと比べると大きくなる。また、複数トークン、文になるにつれ低い相関になった。しかし、どの分類でも決定係数がおおよ

そ 0.3 以下であり、統計的に値が小さいことがわかった。

3.3 相関分析のまとめ

今回の分類毎の単回帰分析では、いずれも決定係数が低く、回帰モデルで解答所要時間を推定できる精度はない。また、全ての穴抜きの構造に対して二変数間である程度の相関があるという結果から、条件を課す、または、解答所要時間、正解率のほかに回帰分析の変数を追加することで実用性のある回帰モデルが作成できる可能性がある。

4. 解答所要時間の分析

4.1 分析の目的

解答所要時間と正解率の相関関係はそれほど強くないことが分かった。そこで解答所要時間だけに注目し、どのような傾向があるかを分析する。分析を行うことで穴抜きと問題の解答所要時間を確認し、穴抜きについての所要時間の考察を行う。

4.2 で解答所要時間の分布が既存のどの分布に近いかを評価し、4.3 で穴抜きの模範解答をトークン数で分け、解答が他のトークン数に比べて異なるような穴抜きを調査している。

4.2 解答所要時間の分布

学生が個別の穴抜きに要する解答所要時間の分布を図 5 に示す。データ総数は 1001 である。x 軸は解答所要時間、y 軸は相対出現度を表している。また、所要時間の階級は 5 秒間隔で集計している。ヒストグラム上のオレンジの線はカーネル密度推定を用いて境界面の設定によらない分布を作図している。また、解答の出現頻度が最も高くなるのは 0~10 秒の間であった。

また、分布についてシャピロ-ウィルク検定、コルモゴロフ-スミルノフ検定を用いて正規性の検定を行った。その結果、どちらも正規分布に従わないという結果になった。解答所要時間の分布にトークン数で分布を分け正規性について調べたが、どのトークン数の場合でも正規分布にはならなかった。

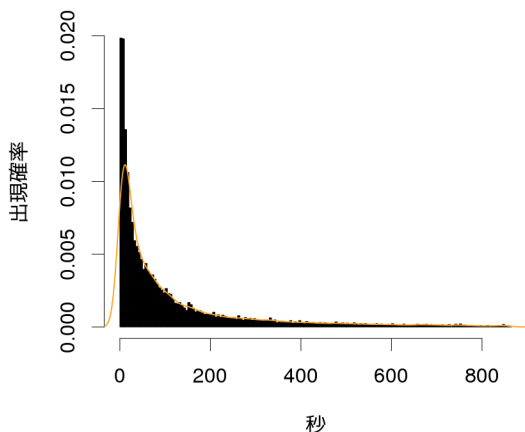


図 4：解答所要時間のヒストグラム

4.3 分布の評価

穴抜きの解答所要時間がどのような分布にしたがっているかを AIC (Akaike's Information Criteria) を用いて評価した。その結果、AIC 値は、正規分布の場合 -2851.648、二項分布の場合 6.009477、ポアソン分布の場合 5055.581 であった。従って、この中で最もモデルの当てはまりが良いものは二項分布である。

4.4 トークン数と解答所要時間の相関分析

穴抜き毎の解答所要時間を対象として分析する。

分類：最小データ数 210

- トークン毎：トークン数 1~12 (ただし、トークン数が 9 と 11 の問題は定義されていない)

分析方法

本研究ではトークン数毎の関係を定量的に検定するために t 検定を採用した。t 検定を用いて総当たりでトークン数毎の有意差を調べている。t 検定の前提条件となる正規性については分布の評価より、解答所要時間の分布は二項分布に近く正規分布に従わないことが分かっている。また、トークン毎に分けた解答所要時間の分布に対してシャピロウィルク検定を用いてすべてのトークン数で正規分布に従わないことを確認している。

トークン数毎の解答所要時間の分散については F 検定を用いてトークン数のすべての組み合わせで等分散であることを確認している。

表 3 ではトークン数毎の解答所要時間の有意差の有無を示す。○は p 値が 5%未満のため有意差が確認できた。一方、×のセルは p 値が 5%以上のため、有意差がないと判断した。

表 3：トークン数毎の解答所要時間の有意差検定

1	○	○	○	○	○	○	○	○	○	
2	×	×	×	○	×	×	×	×	○	
3		○	×	○	○	×	○	○	○	
4			×	○	×	○	×	○	○	
5				○	○	×	×	×	×	
6					×	○	×	○	○	
7						○	×	○	○	
8							○	×	○	
									○	
										○

有意差検定の結果からおおむねトークン数の差が広がると解答所要時間に差が出ることが分かった。しかし、トークン数が離れた場合でも有意差がないと判定された事例がある。これは正解トークン数が多い穴抜きでも、慣用表現 (イディオム) などの影響により、解答所要時間が短い場合があるためだと考えられる。

5. 穴抜きの分類

5.1 分析の目的

pgtracer では学生の学習ログデータを収集しているが、教員がすべての学習ログデータを閲覧し、分析するには手間がかかる。そこで各ユーザーが持つ学習ログデータの分類を行う。分類をすることで解答に特徴のある穴抜きを抽出することを目的とする。

対象データおよび評価尺度

- 穴抜き (78 個)
- 平均解答所要時間(初回時・最高得点時)
- 平均正解率(初回時・最高得点時)
- 平均解答回数(初回時・最高得点時)

分析方法

問題毎の分類でも学生の分類と同様に、ワード法を用いたクラスタ分析が最も均等にクラスタの形成が行われた。図5では4つのクラスタが形成される様子を樹形図で示す。

問題毎の分類分析結果

クラスタの作成を定量的に評価するためにコーフェン相関係数を調べた結果、距離行列とコーフェン相関係数の歪みが少ない方法はユークリッド距離を用いた重心法だったが、その方法を用いるとクラスタ毎の要素数が不均等だったが、そのため本研究では樹形図により、グループ化の状況を視覚的に判断し、各クラスタの要素数が均等に近く、最もクラスタの形成が上手くいっていたキャンベラ距離を用いたワード法を採用した。

枠線は各クラスタの分類を示している。また、クラスタ番号を1~4とすると、データ数はクラスタ1が16個、クラスタ2が20個、クラスタ3が28個、クラスタ4が20個だった。

樹形図で把握した各クラスタについて変数である解答所要時間、正解率の間にどのような違いがあるのかを箱ひげ図を用いて調べた(図6、図7)。クラスタ間の有意差検定にデータ数が少ないこと解答所要時間の分布が正規分布に従わないことがわかっているので、ウィルコクソン順位和検定を採用した。

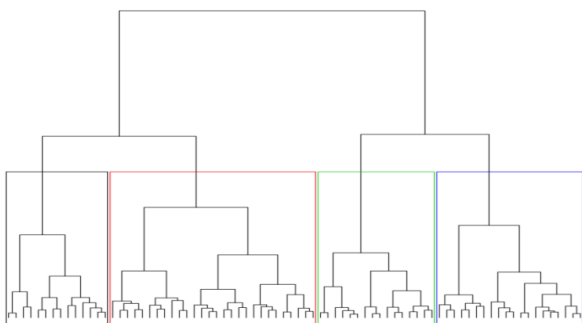


図5：ワード法でのクラスタ形成

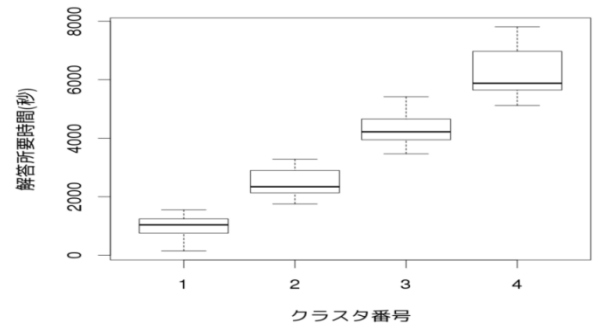


図6：クラスタ毎の解答所要時間分布

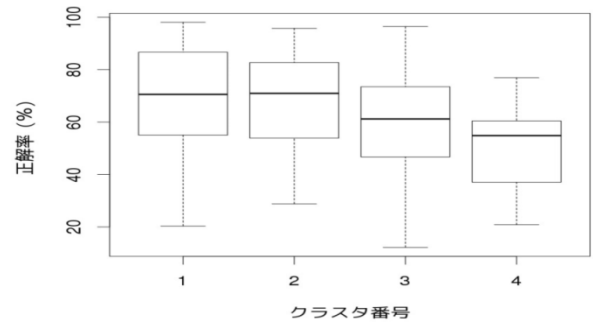


図7：クラスタ毎の正解率分布

クラスタ1では解答所要時間が短い穴抜きを分類できた。また、各クラスタの初回解答所要時間の違いについてウィルコクソンの順位和検定を有意水準5%で行った結果、すべてのクラスタの組み合わせで有意差がみられた。

正解率の箱ひげ図ではクラスタ2から3,4になるにつれて正解率が低下する傾向が見られた。また、各クラスタの正解率についても同様にウィルコクソンの順位和検定で検定した結果、クラスタ1と2の間には有意差がなく、クラスタ1と3、クラスタ2と3の間では有意差がみられた。また、初回解答回数についての有意差検定では正解率と同様のクラスタの間で有意差がみられた。

分析結果の考察

図5で得られた分類を元にそれぞれのクラスタについて穴抜きを観察し、クラスタの傾向を調べた。表4ではクラスタ毎に具体的な穴抜きを3つ挙げ、初回解答時の解答データを示し、プログラム中の穴抜きを表現するため、穴抜きの前後を抜き出し、対象の穴抜きを四角で括弧で表現している。また、具体例を挙げるにあたって初回解答所要時間が平均に近いものをサンプリングした。穴抜きの傾向を調べる方法としてサンプリングした穴抜きの最も重複する誤答を調べた。

クラスタ1

表7の穴番号1では穴抜きの正解が「right=n-1;」に対し、解答に「;」をつけ忘れていた間違いが最も多く、誤答全体で21回の内4回を記録していた。また、穴番号2,3についても正解に近いような解答をしていることが分かった。このことからクラスタ1では比較的難易度の低い穴抜きが所

クラス番号	穴番号	問題	プログラムの穴抜き
1	1	二分探索	//4.探索間を配列の全域とする left=0; right = n - 1;
	2	二分探索	if(x < array [mid]){ right = mid - 1; }
	3	二分探索	cout<<"探索位置"<< right <<endl;
2	4	ハッシュ関数	if(storeResult== -1){\
	5	ヒープソート	for(i==0;i<N;i++){ insert (array [i]);}
	6	括弧の対応検査	if(paren == rightParenTable [i]){\
3	7	ハッシュ関数	//質問データを検索し、結果を表示する searchResult = search (x);
	8	ヒープソート	//1.データ x をヒープの最後尾に追加する m=n; heap [n] = x ;
	9	括弧の対応検査	while(*p!= NULL &&isParen (*p ==0){
4	10	ハッシュ関数	while(hashTable [i] != 0){\
	11	ヒープソート	while(heap[(m - 1) / 2] <x){
	12	ヒープソート	heap[n-1]=0; n -- ; yPos=0;

表4：クラス毎の穴抜きの具体例

属していると考察する。

クラス2

穴番号4では穴抜きの正解が「-1」に対し、「0」と解答したもののがもっとも多く、誤答全体で16回の内、10回を記録した。また、穴抜き番号5では最も多かった誤答が「heap[i]=insert(array[i]);」となっており、問題は理解しているが文の前後関係などの見落としがあるなどの傾向がみられた。このことからクラス2ではクラス1ほど単純ではないが、比較的學生が理解できている穴抜きが所属していると考察する。

クラス3

穴番号7については最も多い誤答が「search(x);」であった。これは誤答全体で28回の内、8回あるため、関数の戻り値をsearchResultに格納する必要があることに気付いていない、複雑なトレース表を正確に追えていないなどが考えられる。

クラス4

穴抜き番号10では誤答の解答は多いものの重複する誤答が少ない傾向がみられる。また、穴番号11では誤答全体で38回の内、「m」と解答したものが11回見られ、重複する誤答の数も多くみられた。このことから、クラス4では學生が理解できなかった穴抜きや、間違いやすい傾向にある穴抜き、穴抜き自体は単純だが、トレース表で数値を辿るのに時間が掛かる穴抜きが所属していると考察する。

5.2 穴抜きの分類分析のまとめ

穴抜きの分類では各クラスでどのような違いがあるかを観察した。その結果、各クラス間の解答所要時間においてウィクソクソソ順位と検定より、全ての組み合わせ

で有意差が見られた。また、クラス1から4になるにつれ解答所要時間が大きくなっており、正解率は徐々に小さくなっていった。また、この分類は初回解答時と最高得点時の学習データを用いたが、初回時の学習データからもほぼ同様のクラスが形成されたため、最高得点時の学習データは穴抜きの分類にはあまり寄与していないと考えられる。

また、クラス毎に重複する誤答について調べ、各クラスの傾向を考察した。クラス1,2では比較的単純な穴抜きが分類され、クラス3,4でアルゴリズムやトレース表といった複数の要素の理解ができなければ解けない穴抜きが分類されていると考えられる。また、クラス4では學生が理解できなかった穴抜きや決まった誤答に誘導されている穴抜き、穴抜き自体は単純だがトレース表を辿るのに時間が掛かる穴抜きの3つが分類されていると考察した。

6. 學生の分類

6.1 分析の目的

学習ログデータを學生の解答毎に分類する。この分類を行うことで解答に有意差のある學生を分類する。また、分類を調べることにより、同じ分類に所属する學生の得意箇所、不得意箇所を把握することを目指している。

対象データ

- 學生 (72人)
- 問題にかけた平均所要時間
- 問題への解答回数
- 問題の平均点

分析方法

學生毎の学習ログデータにクラス分析を行った。クラ

スタ分析ではユークリッド距離を用いた最遠距離法、ユークリッド距離を用いた重心法、キャンベラ距離を用いたウォード法の3つの場合で分析を行った。

学生のカテゴリ分析結果

樹形図による視覚化

図8では3つの大きなクラスタが形成されている様子を示している。クラスタの作成を定量的に評価するためにコーフェン相関係数を調べた結果、距離行列とコーフェン相関係数の歪みが少ない方法はユークリッド距離を用いた重心法だったが、その方法を用いると、小さなクラスタがいくつかできてしまい、うまく分類できていなかった。そのため本研究では樹形図により視覚的に判断し、最もクラスタの形成が上手くいっていたキャンベラ距離を用いたウォード法を採用した。また、クラスタ番号を1~3とすると、それぞれのデータ数は24、25、23であった。

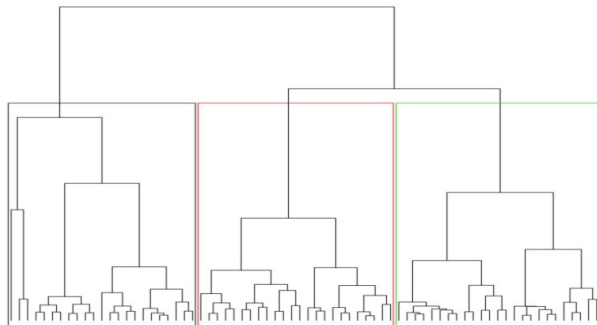


図8：ウォード法でのクラスタ形成

クラスタの詳細

図9では樹形図で示したクラスタの平均所要時間、解答回数、平均点の詳細を箱ひげ図を用いて視覚化を行った。この図では各クラスタ間の平均所要時間の差は少なくとも200秒以上あることが確認できる。

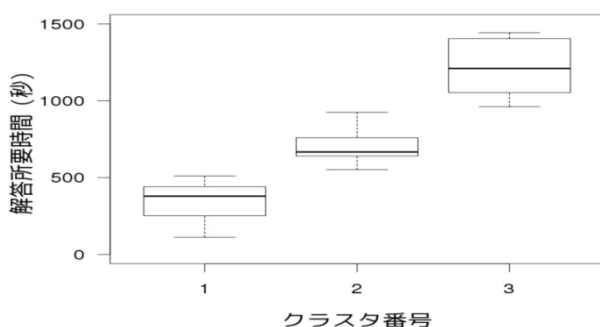


図9：クラスタ毎の解答所要時間の分布

クラスタ毎の解答所要時間、解答回数、平均点について違いを定量的に評価するためにウィルコクソンの順位和検定を行った。また、中央値はクラスタ1が397.5、クラスタ2が667、クラスタ3が1210.455であった。表5では各クラスタの解答所要時間についてウィルコクソンの順位和検定を用いて定量的に評価した結果を示す。検定結果は有意水準5%未満である。

表5：ウィルコクソンの順位和検定結果（解答所要時間）

クラスタ番号	2	3
1	○	○
2	-	○

解答所要時間のみがそれぞれのクラスタ間で有意差があるという結果になった。

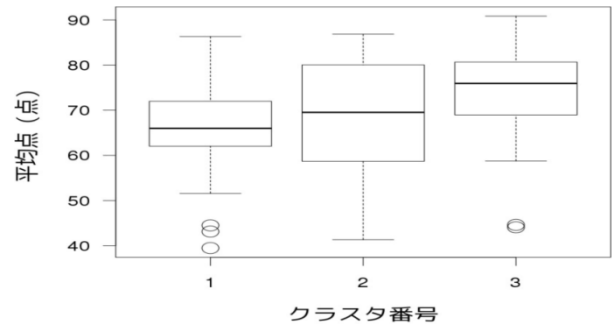


図10：クラスタ毎の平均点の分布

各クラスタの平均点の分布は図10に示すようになっており、中央値はクラスタ1で66.0、クラスタ2で69.5、クラスタ3で76.0であった。

表6：ウィルコクソンの順位和検定結果（平均点）

クラスタ番号	2	3
1	×	○
2	-	×

ウィルコクソンの順位和検定の結果（表6）ではクラスタ1とクラスタ3の間でのみ有意差がみられた。そのほか二つでは有意差は見られなかった。

各クラスタの解答回数の分布は図のようになっており、中央値は、クラスタ1で12.5、クラスタ2で15、クラスタ3で17であった。また、ウィルコクソンの順位和検定の結果、クラスタ間の差は見られなかった。

6.2 分析のまとめ

学生のカテゴリについて解答所要時間、平均点、解答回数の三要素でクラスタ分析を行った。その結果、各クラスタ間で有意差がみられる要素は解答所要時間のみであった。従って、今回の学習ログデータを学生という観点で分類する場合、解答所要時間に基づいて分類することが最も適当であると考えられる。

7. 学生の解答過程の分析

6節で分類した各クラスタに属する学生の解答過程について観察する。学生のカテゴリでは解答所要時間でクラスタ間の有意差がみられた。そこで解答所要時間を代表値とし、実際に異なる学生で同じ問題の解答過程を調べ、解答所要

クラスタ 番号	二分探索		ハッシュ関数		ヒープソート		括弧の対応検査	
	平均点	解答所要 時間 (秒)	平均点	解答所要 時間 (秒)	平均点	解答所要 時間 (秒)	平均点	解答所要 時間 (秒)
1	64.22	335.15	57.13	683.87	41.38	514.67	38.14	543.05
2	73.56	441.44	71.43	1283.57	49.57	1099.25	59.56	1248.89
3	75.98	630.31	75.02	2292.70	63.13	2232.42	65.70	1832.82

時間の差が生じる理由について考察する。

なお、pgtracer に登録されている7つの問題のうち、数表1、数表2、時刻変換の3つは学生の能力を測る意図はなく、pgtracer の操作修得のために作られた基本問題であるため、具体例としては挙げていない。

クラスタの具体的な考察

学生の解答過程を考察するにあたり、同じ問題について、クラスタにより異なる学生の振る舞いを分析する。

表7には各クラスタにおける問題別の平均点(100点満点)と解答所要時間の平均値を示す。クラスタ1では、4つの問題について、解答所要時間には大きな違いはない。一方、クラスタ2および3では問題毎の解答所要時間にはより大きな違いがある。特にクラスタ番号3の解答所要時間は、他のクラスタと比較して長い、平均点もより高いことが、どの問題についても言える。

我々は二分探索問題、ハッシュ関数、ヒープソートについてクラスタから代表的な学生を選び解答過程を調べた。その結果、どの問題についても同様の傾向が見られた。

クラスタ1の中でサンプリングした学生は、冒頭の穴抜きを埋めるために時間が掛かるが、他の穴抜きは比較的短時間で解いている学生、特定の穴抜きに時間をかけているが、それ以外では短時間で解答している学生が所属しており、基本的に順調に解答を行った学生が所属していた。しかし、表7を参照すると、どの問題に対しても他のクラスタよりも所要時間が短く、正解率も低くなっている。そのため、クラスタ1には解ける問題は解答するが、理解できない問題には考える時間を費やさない学生が多く含まれていると考えられる。また、クラスタ1では解答を行っていない(問題を見ただけの)学生、一部のみ解答を行っている学生など、特に解答所要時間が少ない学生が多く所属している。このクラスタに属する学生に対しては、重点的なケアが必要と考えられる。

クラスタ2の中でサンプリングした学生は、ある程度の時間をかけて解答しているが、正解率は幅広く分布しており、一般的な学生が多く含まれていると考えられる。

一方、クラスタ3の中でサンプリングした学生は問題初めの穴抜きを含め、特定の穴抜きに時間をかけている箇所が多かった。また、二分探索問題では探索の範囲指定部分で解答に時間をかけている学生が多くみられた。クラスタ

3では理解できない問題に対しても積極的に取り組む学生が所属していると考えられる。特にクラスタ3の学生には解答が正解になるまで複数回答していた事例が見受けられ、このことが解答所要時間と正解率が他のクラスタよりも高い原因だと考えられる。

8. おわりに

本論文ではプログラミング教育支援ツール pgtracer を用いて収集した穴抜き問題に対する学習ログデータを用い、Rを用いたクラスタ分析と、各クラスタに関する考察を行った。穴抜きの解答所要時間と正解率は、必ずしも相関関係にないことが分かった。また、解答所要時間はトークン数に依存する傾向があることが検定結果より分かった。

学習データの分類では穴抜きと学生の2種類のデータに対してクラスタ分析を行った。穴抜きの分類では全てのクラスタの解答所要時間と特定のクラスタの正解率について有意差のある穴抜きを分類できた。また、穴抜きの傾向を調べ、各クラスタに所属する穴抜きの特徴を考察した。

また、学生の分類では解答所要時間に有意差のあるものを検出し、分類した学生の解答過程を分析した。

今後の研究課題としては、本研究で抽出した各クラスタの学生に対して、学習態度やプログラミング能力を考慮した穴埋め問題を作成する方法に関する検討が挙げられる。

謝辞：本研究は JSPS 科研費 17K01036 の支援を受けています。

参考文献

- [1] 掛下哲郎, 柳田峻, 太田康介, “穴埋め問題を用いたプログラミング教育支援ツール pgtracer の開発と評価”, 情報処理学会論文誌: 教育とコンピュータ, Vol. 2, No. 2, pp. 20-36, 2016.
- [2] 柳田峻, “穴埋め問題を用いたプログラミング教育支援ツール pgtracer”, 佐賀大学工学系研究科知能情報システム学専攻修士論文, 2015年3月.
- [3] Tetsuro Kakeshita, Kosuke Ohta, Student Log Analysis Functions for Web-Based Programming Education Support Tool pgtracer, 17th International Conference on Information Integration and Web-based Applications & Services (iiWAS2015), pp. 120-128, December 2015.
- [4] 太田康介, “プログラミング教育支援ツール pgtracer におけるログデータ分析機能の開発と学生の理解度分析”, 佐賀大学大学院工学研究科知能情報システム学専攻修士論文, 2016年3月
- [5] 金明哲, “Rによるデータサイエンス(データ解析の基礎から最新の手法まで)”, 森北出版, 2017.