自己組織化マップを用いた C 言語演習状況の視覚化

円田智博[†1]　　堂薗浩[†1]

プログラミング講義において講義者は受講者の行動や学習状況の把握が困難である．本研究では，学生の行動情報をキーボードやマウスから取得し，その行動情報を SOM に入力をし，クラスタリングを行い視覚化する．本研究で提案するシステムは，学生の簡易的に学習状況を把握することができる．今後は講義中にリアルタイムで使用することができるシステムの開発を行う．

Visualization of Learning Situation of C Language Using Self-Organizing Maps

TOMOHIRO ENDA[†1]　HIROSHI DOZONO[†1]

The lecturer is difficult to identify the student's behavior and learning situation in programming lecture. In this research, acquiring behavior information of students from the keyboard and mouse, and the visualization is performed by clustering using SOM. The proposed system can identify the learning situation of the students. In the future, the development of systems that can be used in real time during the lecture is expected.

1. はじめに

プログラミングなどの，PC を扱う講義において，講義者側は，受講者側の行動や画面などがわかりづらく，学習進歩状況の判断が困難である．講義者側から受講者を監視するソフトもあるが，それでは，一人ひとりの画面が小さく受講者の学習状況が判断しづらい．また，一人の画面を大きくして監視をすると数人の行動しか把握できない．先行研究においては，キーボード入力やマウス操作などの行動情報をあらかじめ取得，解析，表示する実験を行い，受講者の簡単な学習進捗状況の推測できている[1]．そこで，本研究は，より重要的に高度な学習進捗状況を推測できるアプリケーション開発を行うことを目的とする．本論文では，C 言語を用いて行動情報を入力データとして，自己組織化マップに入力，受講生の行動の類似性を視覚化する実験を行い，受講生の学習状況の推測，評価した経緯を示す．

2. 自己組織化マップについて

2.1 自己組織化マップとは

自己組織化マップ(Self-Organizing Maps : SOM)とは，高次元のデータの類似性を距離に置き換え，2 次元平面マップ上に写像することで，データ間の関係を容易に視覚化できるニューラルネットワークの 1 つである[2]．本研究では，受講生のキーボードやマウスの行動情報を入力ベクトルとして，一般的な SOM アルゴリズムを用いてクラスタリングを行った．

2.2 SOM の例

ここでは, SOM の例として動物のクラスタリングを行ったマップを示す．

表.2-1 に動物の特徴のデータを示す．これは 10 匹のデータで，13 個の特徴をデータで表している．それぞれの動物の特徴を表すセルに 1 を入れ，そうでないセルには 0 を入れている．この 0 と 1 のデータで，1 つ 13 次元のデータを入力ベクトルとする．この動物の入力ベクトルを SOM へ入力してクラスタリングを行った．

Table 2-1　Animal of feature data

	小	中	大	2足	4足	毛	蹄	鼠	羽	狩猟	走	飛	泳
ハト	1	0	0	1	0	0	0	0	1	0	0	1	0
アヒル	1	0	0	1	0	0	0	0	1	0	0	1	1
タカ	1	0	0	1	0	0	0	0	1	1	0	1	0
ワシ	0	1	0	1	0	0	0	0	1	1	0	0	0
ネコ	1	0	0	0	1	1	0	0	0	1	0	0	0
イヌ	0	1	0	0	1	1	0	0	0	0	1	0	0
ライオン	0	0	1	0	1	1	0	1	0	1	1	0	0
トラ	0	0	1	0	1	1	0	0	0	1	1	0	0
ウマ	0	0	1	0	1	1	1	1	0	0	1	0	0
ウシ	0	0	1	0	1	1	1	0	0	0	0	0	0

図.2-1 に動物マップを示す．動物の名前の位置がそれぞれの動物の関係を示している．中央から右上では 4 足動物，左下には 2 足動物に分かれている．上には肉食動物，右下には草食動物，左下には鳥類と分かれている．大きさは左が小，中央が中，右が大となる．

†1 佐賀大学大学院工学系研究科先端融合工学専攻
Department of Advanced Technology Fusion, Graduate School of Science and Engineering Saga University

Fig 2-1 Animal map

このように動物の関係を距離と位置によって類似性を視覚化できている．

SOM の利点は，入力データの類似度をマップ上で表現，高次元のデータを視覚的に理解できる．そして，データがあれば予備知識無しでクラスタリングできるという点があげられる．

3. 受講者の行動データについて

3.1 データ取得方法

データの取得は，HSP(Hot Soup Processor)という言語で取得されている．表.3-1 に対応するキーコードを示す． getkey 命令を使用して表.3-1 で示している対応キーコードをもとに，データの作成を行っている[1].

Table 3-1 Corresponding key-code

キーコード	実際のキー	キーコード	実際のキー
1,2	マウス左,右	48~57	0~9
8	BS	65~90	A~Z
13	ENTER	96~105	テンキー
46	DEL	112~121	F1~F10

3.2 取得データ

実際に取得したデータ例を下記に示す.

20140527165430,258,14,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

始めに書いてある 12 桁の数字は時間を表しており, 2014 年 5 月 27 日 16 時 54 分 30 秒からの 5 秒間のキーコードを示している. 次の数字はキーコードの 1 番目を示しており，順番にキーコードの 121 番目までが書かれている． 実際に解析を行う際は 5 秒のデータを加算して，30 秒のデータを作成して解析している．

キーコードの 1 番目がマウスの左のキーで，2 番目がマウスの右のキーである．データ例の 1 番目のキーコードは 258 と示してある．これは，5 秒間でマウスの左のキーを 2.58 秒間押されているという意味である．5 秒間で同じキーを複数回押された場合は時間が加算されている．実際には，121 番目以降も対応するキーコードはあるが，使われていないため省いている．

3.3 SOM への入力ベクトル

今回時間に関する情報は用いないため，下記に示す数値を入力ベクトルとする．

1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

時間を示す数字以降には，数値が入っているところには 1，数値が入っていないところには 0 のままとし，時間以外の部分をデータとして，121 次元のデータを 1 つの入力ベクトルとしている．

4. 実験環境

4.1 開発環境

データ取得を HSP, データのクラスタリング部分を C 言語，表示に HTML を使用している．

4.2 授業内容

今回実験を行った授業内容の演習問題例のプログラムを下記に示す.

```
#include<stdio.h>
int main (void){
        int no;
        printf(“整数を入力”);
        scanf(“%d”,&no);
        if(no % 2)puts(“奇数”);
        else puts(“偶数”);
        return 0;
}
```

この講義は，C 言語の内容で，受講者が printf や scanf などの入出力や，if の条件分岐などの，C 言語の基本的となる部分である. ここに示しているものは例に過ぎないので，実際には，演習問題として別のプログラムも書いている．

5. 解析結果

5.1 121 次元入力ベクトルを用いた場合

1 つ 121 次元のデータを 30 秒毎に入力ベクトルとしてクラスタリングを行った.

図.5-1 に演習開始前のマップを示す．マップの数字は，受講生の学籍番号を示している．図.1 では，マップの中心に数字が集まっている．これは，講義者が講義を行っているため PC にロックをかけており，自己組織化マップは類似性の高い数字が近くなるので，受講生全員が入力を行なっていない事が推測できる.

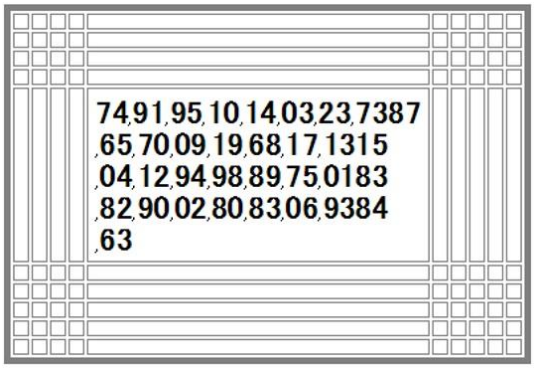

Figure 5-1　Before exercise

図.5-2 に演習開始直後のマップを示す. 数字の色は, C 言語のよく使われる単語やマウス操作に色をつけている.

表.5-1 に対応している色を示す．演習開始直後のマップは，数字が広がり，青い数字が多い．演習開始直後のため受講生が一斉に PC を扱う．そのためクリック操作をしている受講生が多く，速い人では，main を書き込み，ほとんどの受講生が演習を開始していることが推測でき，数字が黒の人は演習を始めていない可能性がある.

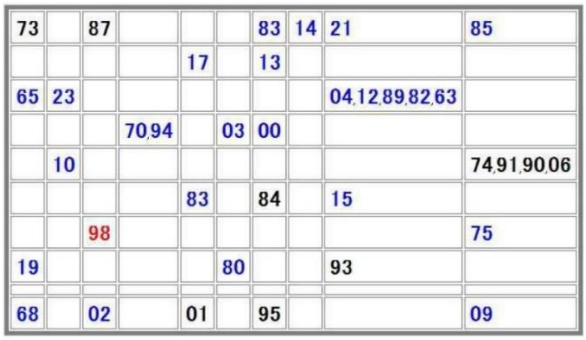

Figure 5-2 Just after a practice start

Table 5-1 Corresponding color

色	対応する操作や単語
青	クリック
赤	main
黄色	printf
ピンク	return
灰色	scanf

図.5-3 に演習途中のマップを示す. 右上の方では, 灰色, 黄色などが多くあり，この周辺の受講生は演習をしっかりと行えていると推測する．しかし，演習を行っている受講生は，確認できるが，右上以外の黒い数字の受講生は，演習がわからないのか，終わっているのか，何もしていないのかという判断ができない.

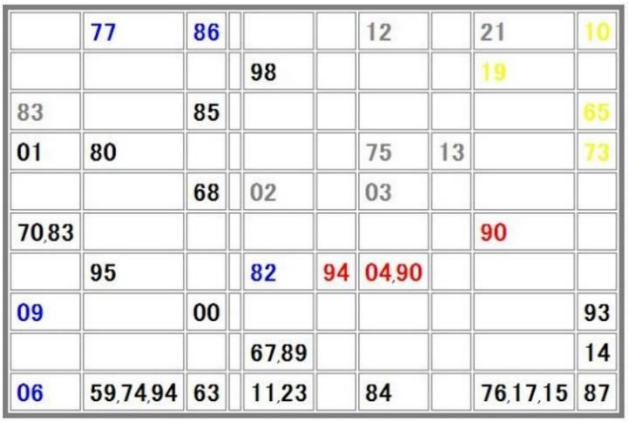

Figure 5-3 During exercise

図 5-4 に演習終了後のマップを示す．多くの数字は集まっているが，数人は，離れている．演習が終わり，講義者が解説を行っているため多くの受講生は PC を扱っておらず，離れている数人は演習をやりつづけていると推測できる. しかし, 離れている受講生が正確に何をしているかは, 判断できない.

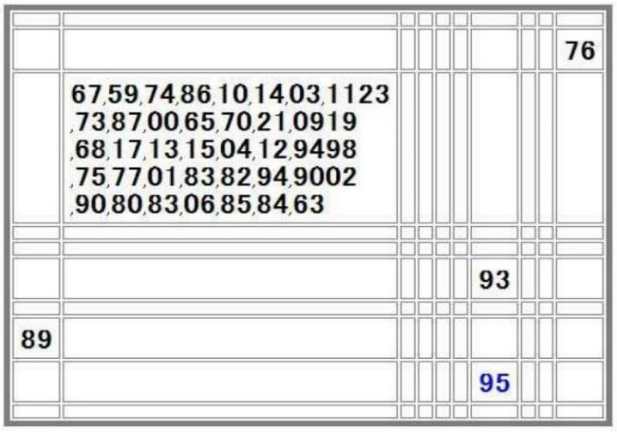

Figure 5-4 After exercise

5.2 入力ベクトルの拡張と関連性の表示

121 次元での解析だけでは，推測できない部分もあるため入力ベクトルの拡張と関連性の付加の手法を取り入れた.

入力ベクトルの拡張では，今までの 121 次元のデータを拡張し，より演習ができている受講生の区別がつくように次元数を拡張した。次元数の拡張は，今までの 121 次元のデータの後に，別のデータを付け加えている．122 次元目

からのデータは，main を書けたら 1 を 122 次元目にいれ，順に printf，return，scanf として，125 次元のデータを作成する．取得されているデータは，30 秒毎に更新されるが，この 122 次元以降のデータは更新されずに次のデータに引き継がれる様にしている．

関連性の付け方ついては，解析を行った後のマップで，それぞれの番号を基準に入力ベクトルの距離が 1 番目に近い番号を関連させる．30 秒毎の解析の度に行うので，後半に行くに連れて関連度が増していく．

図.5-5 に演習前のマップを示す．121 次元の解析と同様の結果となっている．ここでは，受講者が入力を行っていないので付け加えたデータも同じになり，変わらない結果となった．数字の右下に書いてある数字は，関連性の高い受講者の数字を示しているが，誰も入力を行っていないので関連性はすべての受講者が同じ関連性を持っている．

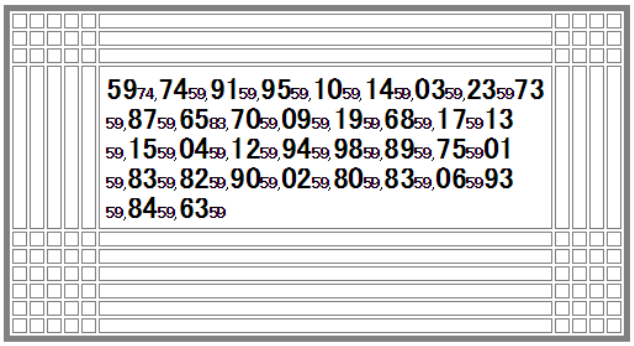

Figure 5-5 Before exercise

図.5-6 に演習途中のマップを示す．右上に黄色，中央左に赤が表示されている．受講者が演習を行っており，右上の受講者は printf を扱い，中央左の受講者は main を扱っているため，きちんと演習できているのではないかと考えられる．関連度の高い受講者が離れているのは，30 秒毎のデータは更新されているので 30 秒毎の関係がマップに表示されるためである．この関連度により，演習ができている受講生と関連のある受講生は，演習ができているか，その受講生の演習を写している可能性が高いと推測できる．

12 67		59 67			67 59		19 10		74 59,70 86,90 94
		84 59							
01 59				11 67			09 65		63 06
		83 74				13 67			
80 86		98 14,89 00	95 67	90 75				77 19	
							00 89		06 90
75 67		15 65		93 86			04 95		
		14 95,03 14,68 67,85 03					94 23		23 94,73 87
87 14		65 15	86 74		94 75,02 70				
	17 82		21 67,83 86	76 67		10 19		82 17	

Figure 5-6 During exercise

図.5-7 に演習後のマップを示す．多くの受講者は集まっているが，数人離れている．この離れている受講者は，演習ができていない可能性もあるが，関連度の受講者を見ると集まっている受講者の番号を示すので，演習はできているが PC を扱っていることが推測できる．しかし，演習はできているかもしれないが，PC を使っているので，演習以外のことを行っている可能性が出てくる．次元数を増やすことで，最後に演習ができていない受講生が離れると予測したが，集まっているところ以外は点在しているため判断が難しい．

			95 67						
								93 67	
		76 67		67 59,59 67,74 67,86 74,10 67,14 03,03 14,11 00 23 67,73 67,87 14,00 11,65 67,70 86,21 67,09 67 19 67,68 67,17 67,13 67,15 65,04 67,12 67,94 23 98 21,75 67,77 67,01 82,83 67,82 67,94 67,90 74 80 67,83 67,06 90,85 03,84 67,63 06				
								02 67	
			89 00						
							90 00		

Figure 5-7 After exercise

6. まとめ

今回の研究では，受講者の行動を，SOM を用いて視覚化することができ，学習状況の簡単な推測ができた．しかし，簡単な推測しか行えないため，実際の講義では使用は困難と思われた．次元数を増やすことでは，増やした次元数が少ないため，効果はあまりなかった．関連度の付加では，PC にロックをかけている状態でも関連度を増やしているので，遅刻してきた受講生との関連度は少なくなり，実際に演習だけの時間では関連度が大きくならない問題がある．30 秒毎の関連度であるため，考えている受講者と何もしていない受講者の時間が重なって，関連のないところでも関連度が大きくなることもある．今後は，次元数と関連度の見直し，高度で正確な推測が行えるアルゴリズムの考案や，リアルタイムで表示することができる UI の開発を行っていく．

参考文献

1) 川野義広，堂薗浩，中國真教(2013) 「自己組織化マップを用いた C 言語学習状況の視覚化」，第 15 回 日本知能情報ファジィ学会 九州支部学術講演会
2) T.Kohonen(2005)『自己組織化マップ--Self-Organizing maps--』，倉田耕治・中塚大輔・内野英治・山川烈・和久屋寛・伊藤則夫・加藤聡訳，pp111，シュプリンガー・フェアラーク東京株式会社