

# Scratch プログラムの可視化による類似度推定

G13924 森下汐美

平成 29 年 1 月 31 日

## 1 はじめに

近年プログラミング教育の推進に伴い、義務教育化が進んでいる。その中で米国マサチューセッツ工科大学のメディアラボが開発した Scratch は無償で提供されているグラフィックプログラミング環境である。プログラミングを行う際の命令を本ツールではブロックを組み合わせて作り上げる。初心者にとっては使いやすい構造となっているため米国では利用が増えているものの、日本のユーザーは全体の 1% にも満たない。そこで実際に本ツールで公表をされているデータを利用してより教育に用いられるツールの解析を目指す。Scratch 公式サイトのデータでは全体図の把握が可能であるが 1 つのプログラムでブロックがどのように使われているか、引用していた場合引用元からどの程度変更させたかは不明である。従って本研究ではある元のプログラムと関連プログラムのブロックを主に解析し結果を基によりわかりやすいデータ発掘を目指す。

## 2 使用した技術

本研究では 2 つのプログラムと 1 つの辞書データを使用する。PPython 環境では json モジュールをインポートすることで JSON 形式ファイルを読み込み辞書型に変換されるため ython で解析プログラムを作成し、JSON 形式でインポートした Scratch プログラムを解析する。

ソースコード 2 はブロック、スプライトの数、cos 類似度を計算結果を出力する。ソースコード 1 は cos 類似度を実施に計算するプログラムとなっている。cos 類似度とはベクトル空間モデルにおいて、文書同士を比較する際に用いられる類似度計算方法である。

【数式】

$$\cos(A, B) = \frac{\vec{A} * \vec{B}}{|\vec{A}| |\vec{B}|} = \frac{\vec{A}}{|\vec{A}|} * \frac{\vec{B}}{|\vec{B}|} = \frac{\sqrt{\sum_{i=1}^{|V|}}}{\sqrt{\sum_{i=1}^{|V|} A_i^2 * \sum_{i=1}^{|V|} B_i^2}} \quad (1)$$

本研究では例”色”に当たる部分を、”ブロックの種類”にして計算を行った。Scratch では数多くのブロックが用意されており、どのブロックを使用するかによって、まったく異なるプログラムを作成することができる。それぞれのプログラムのブロック数を集計した後、ソートしベクターに直し計算式に適用する。

### 3 実験結果

出力されたデータを用いて散布図に表した。縦軸に  $\cos$  類似度の値、横軸に出力されたブロック数（スプライト数）を元のプログラム（"Pong Starter"）の数値で割った値の対数値でとる。数値では（1,1）が最も類似しているプロジェクトであるため似ているものから青、赤、緑、紫、水色で色を区別してプロットをする。

#### 3.1 ブロック数と $\cos$ 類似度のグラフ

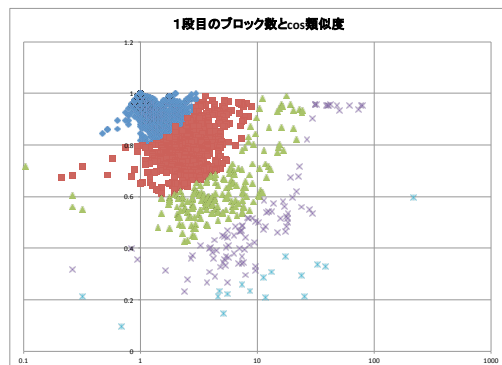


図1 1段目のグラフ

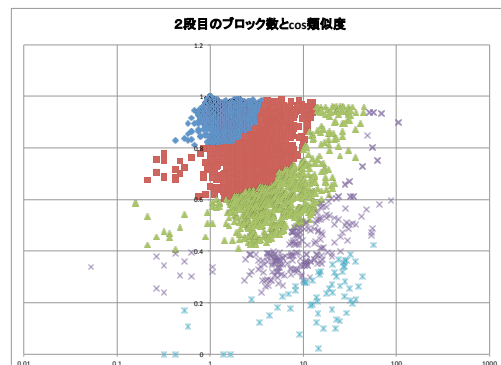


図2 2段目のグラフ

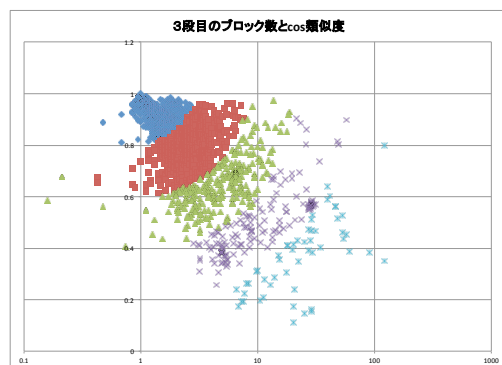


図3 3段目のグラフ

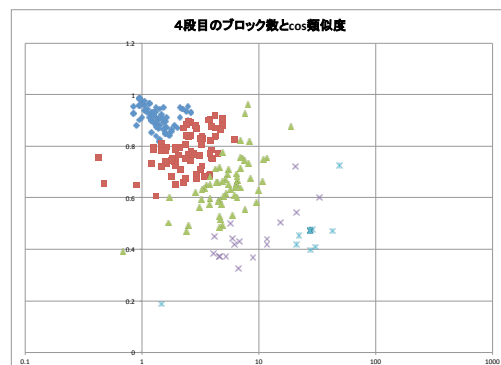


図4 4段目のグラフ

#### 3.2 スプライト数と $\cos$ 類似度のグラフ

### 4 評価

### 5 まとめ

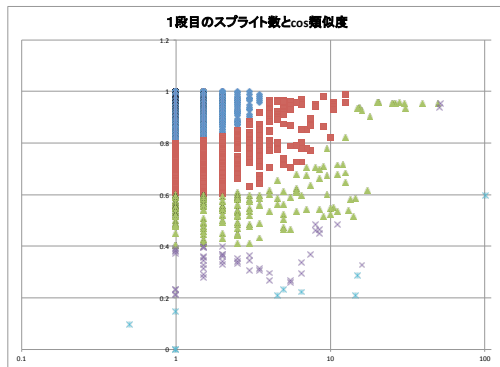


図5 1段目のグラフ

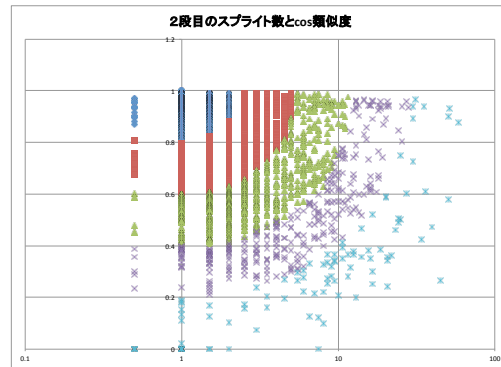


図6 2段目のグラフ

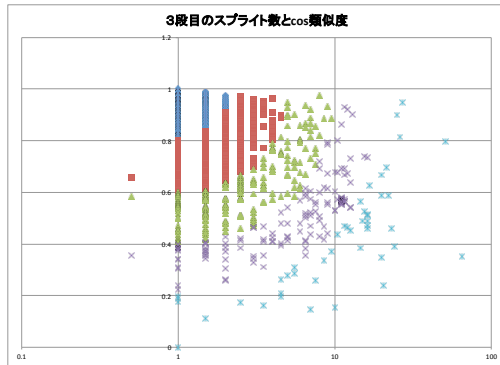


図7 3段目のグラフ

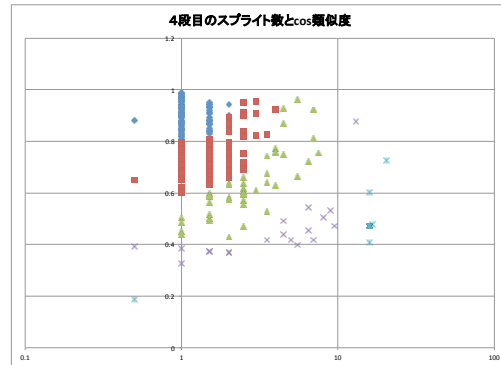


図8 4段目のグラフ

#### ソースコード 1 SimCalculator.py

```

1 # -*- coding: utf-8 -*-
2
3 #cos 類似度
4 import math
5
6
7 class SimCalculator():
8     def sim_cos(self, v1, v2):
9         numerator = 0
10        # v1 と v2 で共通する key があつたとき、その値の積を加算していく。2つのベクトルの内積になる。
11        i = 0
12        for n in v1:
13            if n in v2:
14                numerator += v1[i]*v2[i]
15                i=i+1
16
17        ss1 = []
18        ss2 = []
19        for x in v1:
20            ss1.append(x*x)
21        for x in v2:
22            ss2.append(x*x)
23
24        denominator = math.sqrt(sum(ss1))*math.sqrt(sum(ss2))
25
26        if denominator == 0:
27            return 0
28        return numerator / denominator

```

## ソースコード 2 scratchAnalysis.py

```

1  # -*- coding: utf-8 -*-
2
3  '''
4  各モジュールをインポートする
5  JSON モジュール
6  URLLIB2 モジュール
7  CSV モジュール
8  '''
9
10 import json
11 import urllib2
12 import csv
13
14 '''
15 SCRATCH_BLOCK.PY をインポートする
16 '''
17
18 import scratch_block
19
20 '''
21
22 COS 類似度を計算する SIMCALCULATOR.PY_のインポート
23 '''
24 from SimCalculator import SimCalculator
25
26 '''
27
28 事前に作成した分析対象プロジェクトをリスト化した
29 CSV ファイルの取得
30 '''
31 dataReader = csv.reader(open("PONG.CSV", "rb"))
32
33
34 '''
35 ブロックの種類とその個数をカウントし、
36 SCRATCH_BLOCK.PY に保存をする
37 '''
38
39 def getFirst(L, dict):
40     global count
41     if isinstance(L, list) and len(L) > 0 :
42         first = L[0]
43         if isinstance(first, unicode):
44             '''
45             例外ブロックの処理
46             '''
47             if "TURNRIGHT:" in first:
48                 first = first.rstrip(":")
49             if "TURNLEFT:" in first:
50                 first = first.rstrip(":")
51             if "WAIT:ELAPSED:FROM:" in first:
52                 first = first.rstrip(":")
53
54             if first in dict:
55                 dict[first] += 1
56                 count += 1
57         for e in L:
58             getFirst(e, dict)
59
60 '''
61 比較対象となる元のプロジェクトのデータを URL よりプロジェクト番号をして取得
62 JSON 形式のファイルを読み込み、"CHILDREN" の持つリストに絞る
63 その中のブロックが使用されている "SCRIPTS" リストにさらに絞る
64 '''
65
66 url1 = 'HTTP://PROJECTS.SCRATCH.MIT.EDU/INTERNALAPI/PROJECT/'+ '10000036' + '/GET/'
67 r1 = urllib2.urlopen(url1)
68 root1 = json.loads(r1.read())
69 y1 = root1["CHILDREN"]
70 a = "SCRIPTS"
71 s1 = []
72
73 '''
74 全体のブロック数をKEY の指定をしブロック数を数える FOR 文により SCRATCH_BLOCK.PY の値を更新
75 '''
76
77 for i in range(len(y1)):
78     if not a in y1[i].keys():
79         continue

```

```

80     s1 = s1 + y1[i][u'SCRIPTS']
81
82     '''
83     スプライト数がOBJNAMEで指定されているためその個数を調べるFOR文
84     '''
85
86     splite_count1 = 0
87     for i in range(len(y1)):
88         if 'OBJNAME' in y1[i]:
89             splite_count1 += 1
90
91
92     '''
93     元のプログラムの個数を求めていく
94     GETFIRSTのメソッドを動かす
95     '''
96
97     count = 0
98     x = scratch_block.block
99     getFirst(s1,x)
100
101     '''
102     GETFIRSTで集計された個数をブロックの名前順にソートをし、
103     ブロック名を覗くことでベクターを取得する
104     '''
105     xx = x.keys()
106     xx = sorted(xx)
107     xxx = []
108     for ww in xx:
109         xxx.append(x[ww])
110
111     '''
112
113     FOR文でCSVファイルのURLを順番に取得し、計算を行う
114     '''
115
116     for row in dataReader:
117
118         row = "".join(row)
119         row = row.replace('HTTPS://SCRATCH.MIT.EDU/PROJECTS/', '')
120         print row
121
122     '''
123
124     比較をする2つ目のプロジェクトを読み込む
125     以後元のプロジェクトに行った同様の処理を行う
126     '''
127
128     url2 = 'HTTP://PROJECTS.SCRATCH.MIT.EDU/INTERNALAPI/PROJECT/'+row+'GET/'
129     r2 = urllib2.urlopen(url2)
130     root2 = json.loads(r2.read())
131     y2 = root2["CHILDREN"]
132     s2 = []
133
134     for i in range(len(y2)):
135         if not a in y2[i].keys():
136             continue
137         s2 = s2 + y2[i][u'SCRIPTS']
138
139     count = 0
140     y = scratch_block.block
141
142     '''
143     1つ目のプロジェクトの個数がSCRATCH_BLOCK.PYに残っているため初期化をする
144     '''
145
146     for www in y:
147         y[www] = 0
148
149     getFirst(s2,y)
150
151     splite_count2 = 0
152     for i in range(len(y2)):
153         if 'OBJNAME' in y2[i]:
154             splite_count2 += 1
155
156     yy = y.keys()
157     yy = sorted(yy)
158     yyy = []
159     for ww in yy:
160         yyy.append(y[ww])

```

```

161     '''
162
163     1  2  3
164     1  2  3
165
166     if __name__ == '__main__':
167         sc = SimCalculator()
168         cos = sc.sim_cos(1,2,3)
169         print str(cos)
170
171     '''
172     1  2  3
173     1  2  3
174     1  2  3
175
176     f = open("RESULT_PONG.CSV", "w")
177     writecsv = csv.writer(f)
178     header = ['BLOCK', 'SPLITE', 'COS']
179     csvlist = []
180
181     '''
182     1  2  3
183     1  2  3
184     1  2  3
185
186     body = [count, splite_count2, cos]
187     csvlist.append(body)
188
189
190     writecsv.writerow(header)
191     writecsv.writerows(csvlist)
192
193     f.close()

```

### ソースコード 3 scratch\_block.py

```

1  # -*- coding: utf-8 -*-
2  #動き
3  block = {
4      'FORWARD':0,
5      'TURNRIGHT':0,
6      'TURNLEFT':0,
7      'HEADING':0,
8      'POINTTOWARDS':0,
9      'GOTOX:Y':0,
10     'GOTOSPRITEORMOUSE':0,
11     'GLIDSECS:TOX:Y:ELAPSED:FROM':0,
12     'CHANGEXposBY':0,
13     'XPOS':0,
14     'CHANGERYposBY':0,
15     'YPOS':0,
16     'BOUNCEOFFEDGE':0,
17     'SETROTATIONSTYLE':0,
18     'XPOS':0,
19     'YPOS':0,
20     'HEADING':0,
21  #見た目
22     'SAY:DURATION:ELAPSED:FROM':0,
23     'SAY':0,
24     'THINK:DURATION:ELAPSED:FROM':0,
25     'THINK':0,
26     'SHOW':0,
27     'HIDE':0,
28     'LOOKLIKE':0,
29     'NEXTCOSTUME':0,
30     'STARTSCENE':0,
31     'CHANGEGRAPHICEFFECT:BY':0,
32     'SETGRAPHICEFFECT:TO':0,
33     'FILTERRESET':0,
34     'CHANGESIZEBY':0,
35     'SETSIZETO':0,
36     'COMETOFRONT':0,
37     'GOBACKBYLAYERS':0,
38     #'costumeIndex':0,
39     #'sceneName':0,
40     #'scale':0
41  #音
42     'PLAYSOUND':0,
43     'DOPLAYSOUNDANDWAID':0,
44     'STOPALLSOUNDS':0,
45     'PLAYDRUM':0,
46     'REST:ELAPSED:FROM':0,
47     'NOTEON:DURATION;ELAPSED;FROM':0,
48     'INSTRUMENT':0,
49     'CHANGEVOLUMEBY':0,
50     'SETVOLUMETo':0,
51     #'volume':0,
52     #'changeTempo':0,
53     'SETTEMPOTo':0,
54     #'tempo':0
55  #ペン
56     'CLEARPENTRAILS':0,
57     'STAMPCOSTUME':0,
58     'PUTPENDOWN':0,
59     'PUTPENUP':0,
60     'PENCOLOR':0,
61     'CHANGEPENHUEBY':0,
62     'SETPENHUETo':0,
63     'CHANGEPENSHADEBY':0,
64     'SETPENSHADETo':0,
65     'CHANGEPENSizeBY':0,
66     'PENSize':0,
67  #データ
68     'READVARIABLE':0,
69     'SETVAR:TO':0,
70     'CHANGEVAR:BY':0,
71     'SHOWVARIABLE':0,
72     'HIDEVARIABLE':0,
73     'CONTENTSOFLIST':0,
74     'APPEND:TOList':0,
75     'DELETELINE:OFLIST':0,
76     'INSERT:AT:OFLIST':0,
77     'SETLINE:OFLIST:TO':0,
78     'GETLINE:OFLIST':0,
79     'LINECOUNTOFLIST':0,

```

```

80      'LIST:CONTAINS':0,
81      'SHOWLIST':0,
82      'HIDEList':0,
83  #イベント
84      'WHENGREENFLAG':0,
85      'WHENKEYPRESSED':0,
86      'WHENCLICKED':0,
87      'WHENSCENESTARTS':0,
88      'WHENSSENSORGREATERTHAN':0,
89      'WHENIRECEIVE':0,
90      'BROADCAST':0,
91      'DOBROADCASTANDWAIT':0,
92  #制御
93      'WAIT:ELAPSED:FROM':0,
94      'DOREPEAT':0,
95      'DOFOREVER':0,
96      'DOIF':0,
97      'DOIFELSE':0,
98      'DOWAITUNTIL':0,
99      'DOUNTIL':0,
100     'STOPSCRIPTS':0,
101     'WHENCLONED':0,
102     'CREATECLONEOF':0,
103     'DELETECLONE':0,
104  #調べる
105     'TOUCHING':0,
106     'TOUCHINGCOLOR':0,
107     'COLOR:SEES':0,
108     'DISTANCETO':0,
109     'DOASK':0,
110     'ANSWER':0,
111     'KEYPRESSED':0,
112     'MOUSEPRESSED':0,
113     'MOUSEX':0,
114     'MOUSEY':0,
115     'SOUNDLEVEL':0,
116     'SENSEVIDEOMOTION':0,
117     'SETVIDEOSTATE':0,
118     'SETVIDEOTRANSARENCY':0,
119     'TIMER':0,
120     'TIMERRESET':0,
121     'GETATTRIBUTE:OF':0,
122     'TIMEANDDATE':0,
123     'TIMESTAMP':0,
124     'GETUSERNAME':0,
125  #演算
126     '+':0,
127     '-':0,
128     '*':0,
129     '/':0,
130     'RANDOMFROM:TO':0,
131     '<':0,
132     '=':0,
133     '>':0,
134     '&':0,
135     '|':0,
136     'NOT':0,
137     'CONCATENATE:WITH':0,
138     'LETTER:OF':0,
139     'STRINGLENGTH':0,
140     '%':0,
141     'ROUNDED':0,
142     'COMPUTEFUNCTION:OF':0,
143  #その他
144     'PROCDEF':0
145  }
146
147  block2 = sorted(block)
148  # print block2

```