

環境認識ライログからの 行動パターン解析による 類似性・イベント検出

富山県立大学 福嶋瑞希

目次

- ・はじめに
- ・ライフログ取得アプリケーション
- ・自己組織化マップ
- ・今後の展望

はじめに

GoogleのVisionAPIを用いたAR教育コンテンツの作成と
学習評価に関する考察



卒業論文として認められない危険性

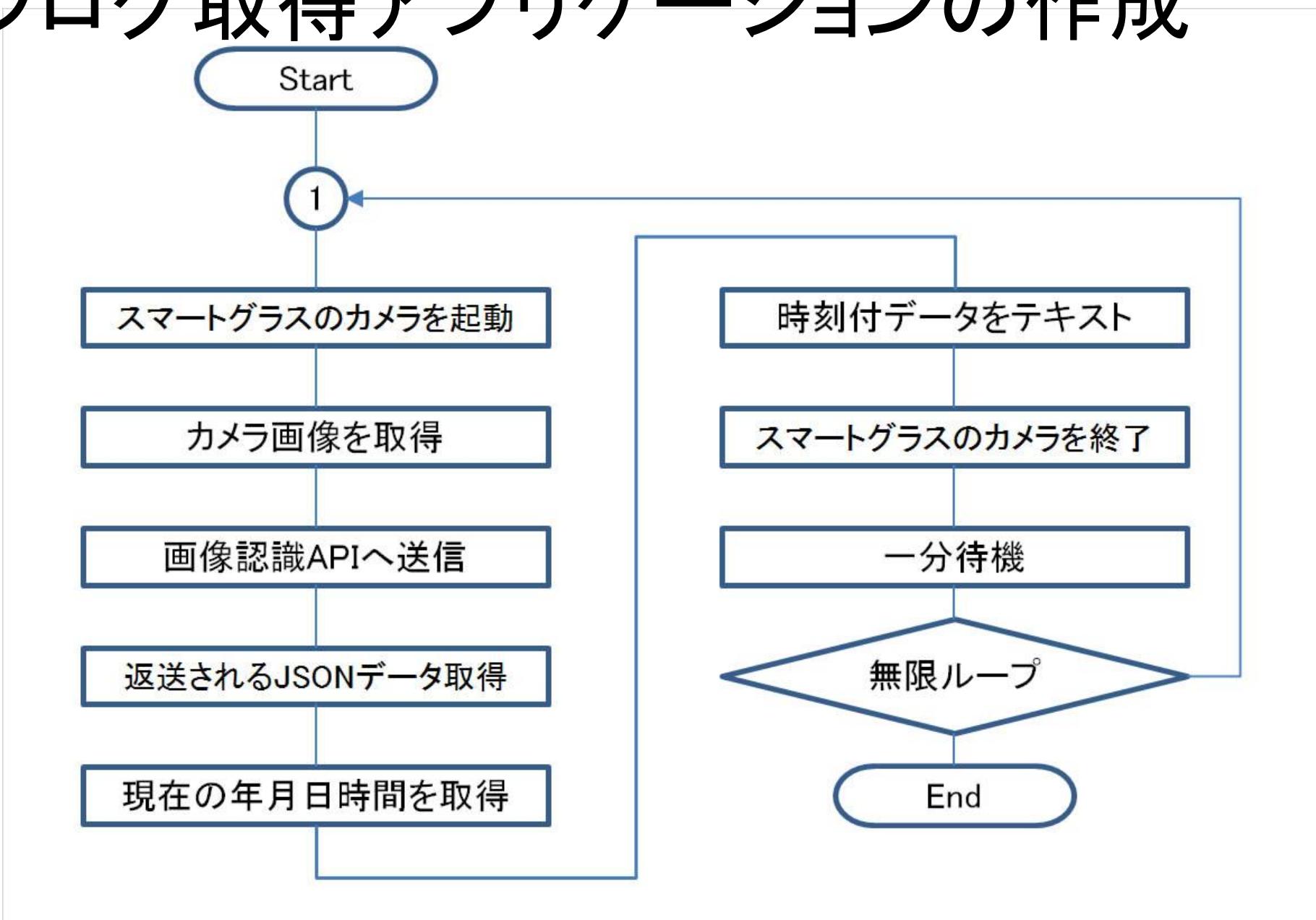


環境認識ログからの行動パターン解析による類似性・イベント検出

ライフログ

- ・環境認識ライフログからの行動パターン解析
→ ライフログ(lifelog)=人間の活動(life)+記録(log)
- ・プライバシーの侵害
- ・誰でも負担のないライフログの収集

ライフログ取得アプリケーションの作成



取得したデータ

[2018-01-06 09:46:04]

```
{"description":{"tags":["indoor","table","sitting","laptop","computer","top","water","open","small","plastic","desk","green","counter","white","playing","laying","phone"],"captions":[{"text": "an open laptop computer sitting on a table", "confidence":0.87985064456962381}],"requestId":"89ebe0a8-b490-482d-b213-730775945669","metadata":{"width":1280,"height":720,"format":"Jpeg"}}
```

[2018-01-06 09:47:16]

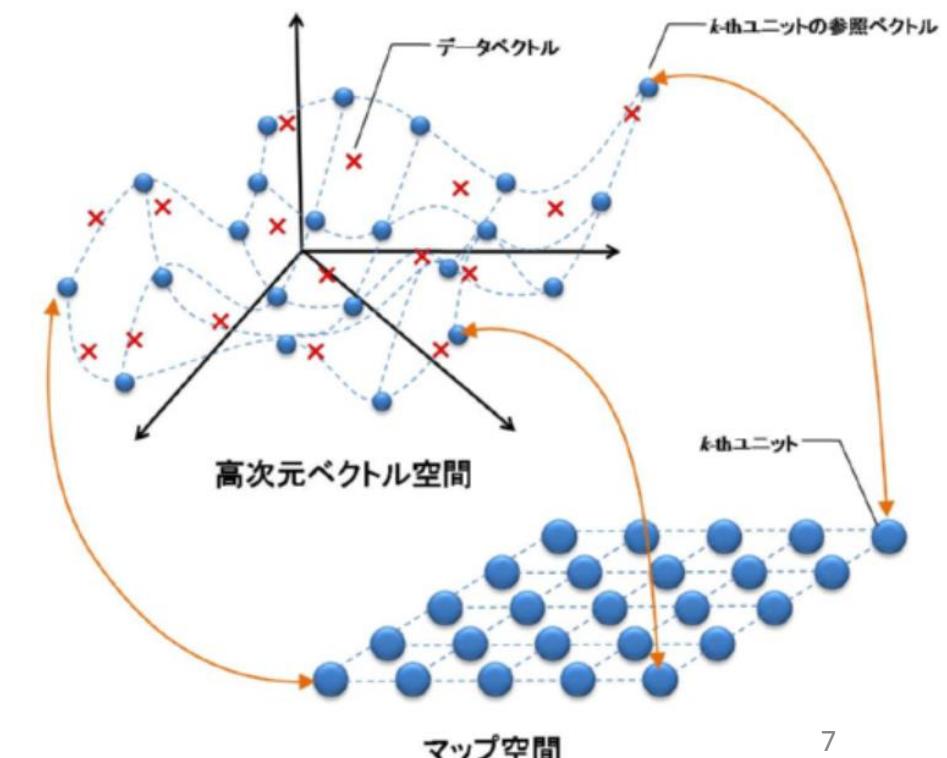
```
{"description":{"tags":["laptop","computer","indoor","electronics","sitting","table","open","top","keyboard","plastic","white","display","playing"],"captions":[{"text": "an open laptop computer sitting on a table", "confidence":0.9236841532655462}],"requestId":"2c2f551d-78db-4951-a99a-8fe6e61a954e","metadata":{"width":1280,"height":720,"format":"Jpeg"}}
```

[2018-01-06 09:48:30]

```
{"description":{"tags":["indoor","table","computer","refrigerator","items","sitting","desk","counter","kitchen","top","small","cluttered","laptop","filled","man","sink","laying","white","different","room","bunch","covered","keyboard","bed","holding","many","mirror","group","water","woman","standing","people","bedroom"],"captions":[{"text": "a bunch of items that are sitting on a counter", "confidence":0.69510813380373893}],"requestId":"74b0a6b0-7c29-43cd-9fa6-4d3694d665e9","metadata":{"width":1280,"height":720,"format":"Jpeg"}}
```

自己組織化マップ(Self Organizing Map:SOM)

- 教師なし学習を行なうニューラルネットワークの代表例
- 何種類かのデータを繰返し入力し続ける:似たようなデータを近くに配置していく
- 一種のデータ圧縮
- 類似した性質を持ったものどうしが近接するように低次元化を行う



SOMのアルゴリズム: 概要

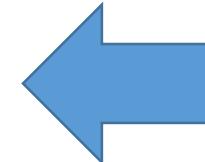
- 1, 入力データを用意する
- 2, 出力結果(平面の結果)を初期化する
- 3, 綺麗に並べ替える(学習をする)

1.入力データ

	A	B
1	時間	キャプション
2	9:46:04	an open laptop computer sitting on a table
3	9:47:16	an open laptop computer sitting on a table
4	9:48:30	a bunch of items that are sitting on a counter
5	9:49:42	an open laptop computer sitting on a table
6	9:50:54	an open laptop computer sitting on a table
7	9:52:06	a laptop computer sitting on a table
8	9:53:18	person sitting at a table with a laptop
9	9:54:30	a laptop computer sitting on a table
10	9:55:42	person sitting at a table with a laptop computer
11	9:56:54	an open laptop computer sitting on a table
12	9:58:06	person sitting at a table using a laptop computer
13	9:59:18	an open laptop computer sitting on a table



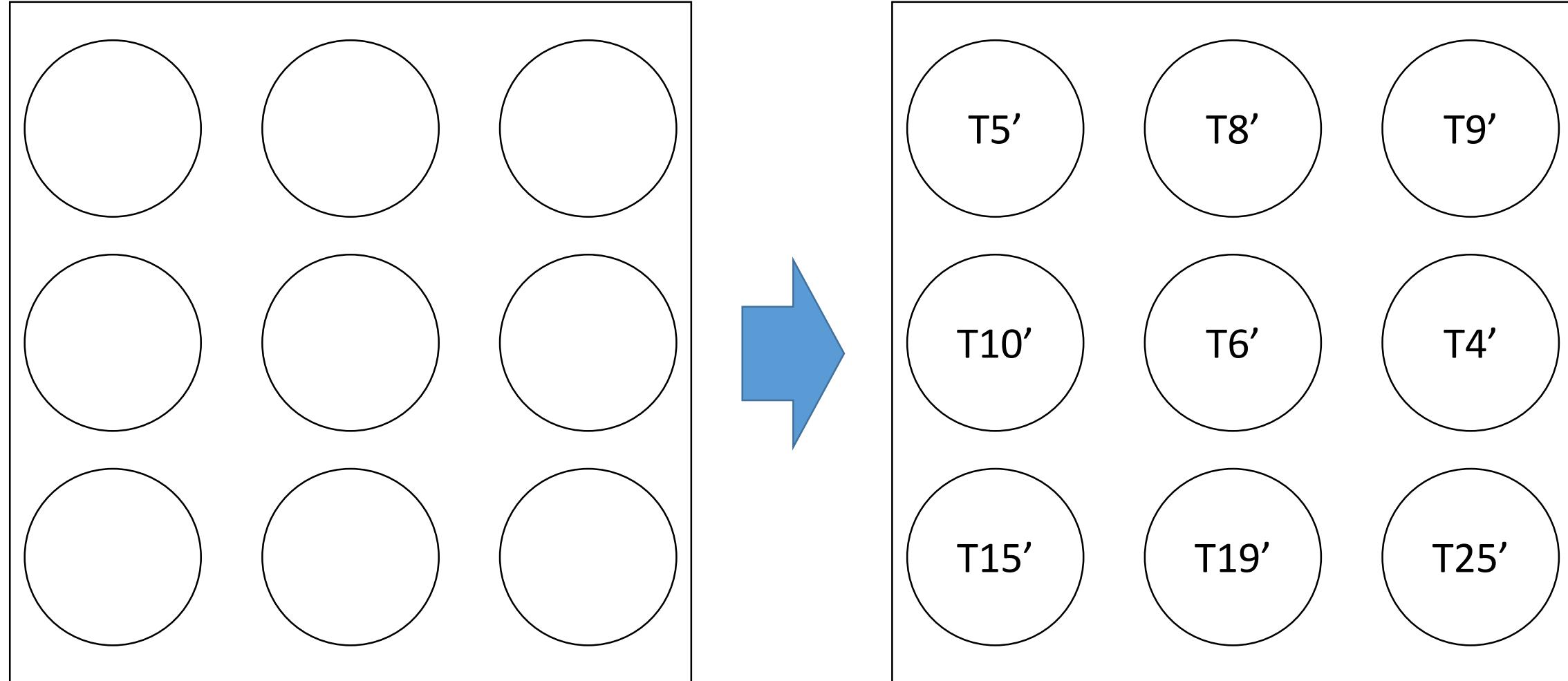
	A	B	C	D	E	F	G	H	I	J
1	time	computer	laptop	table	desk	screen	shot	person	room	standing
2	t1		1	1	1	0	0	0	0	0
3	t2		1	1	1	0	0	0	0	0
4	t3		0	0	0	0	0	0	0	0
5	t4		1	1	1	0	0	0	0	0
6	t5		1	1	1	0	0	0	0	0
7	t6		1	0	1	0	0	0	0	0
8	t7		0	1	1	0	0	0	1	0
9	t8		1	0	1	0	0	0	0	0
10	t9		1	0	1	0	0	0	1	0
11	t10		1	1	1	0	0	0	0	0
12	t11		1	0	1	0	0	0	1	0
13	t12		1	1	1	0	0	0	0	0
14	t13		1	1	1	0	0	0	0	0
15	t14		1	1	1	0	0	0	0	0
16	t15		1	1	1	0	0	0	0	0
17	t16		1	1	1	0	0	0	0	0



KHcoder

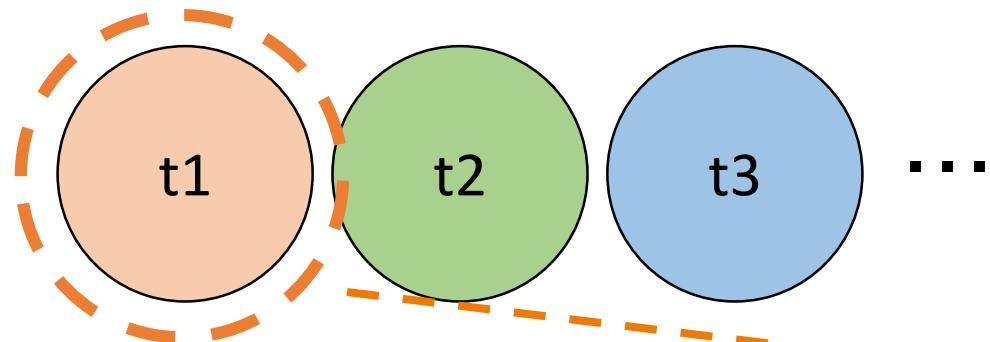
- 1,単語で切り取る
- 2,文の中に出現する単語の回数を表にまとめる
- 3,csvで出力

2. 出力結果(平面の結果)を初期化



3,綺麗に並べ替える(学習をする)

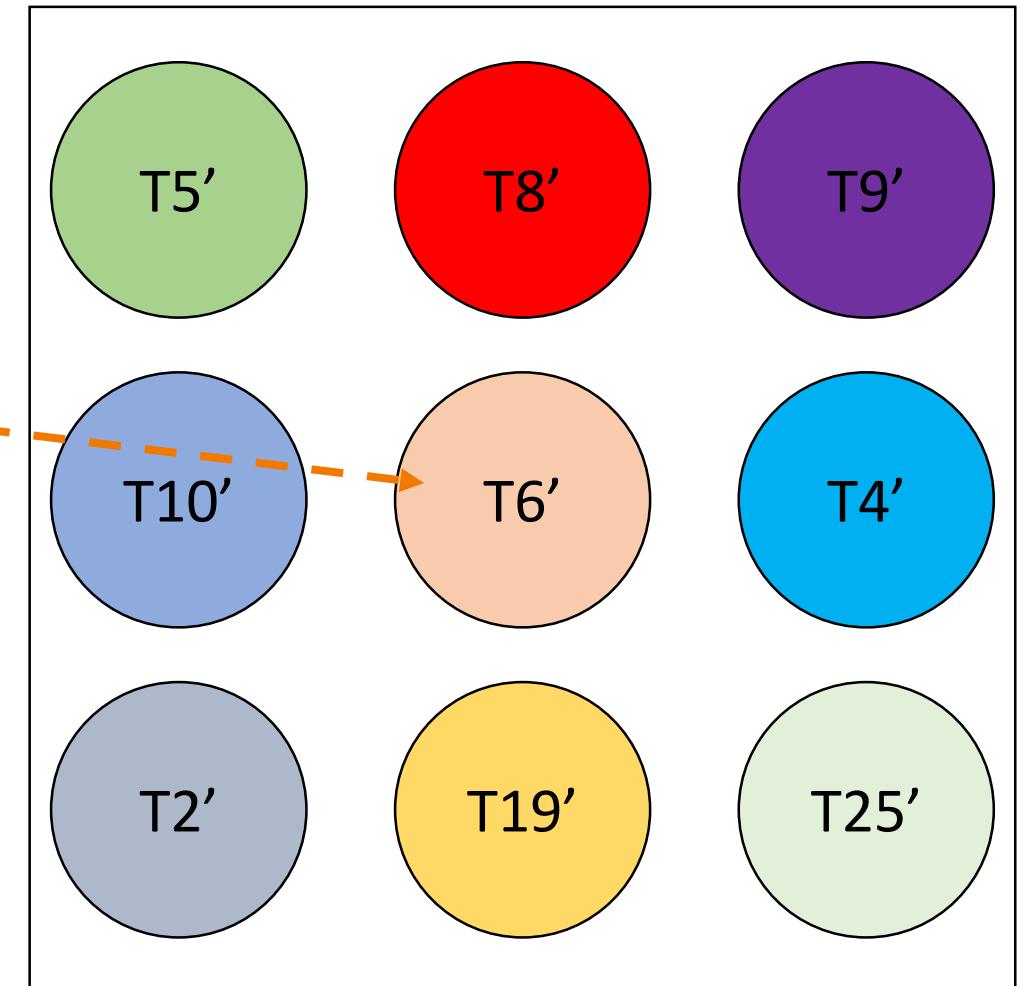
入力データ(入力層)



学習の方法

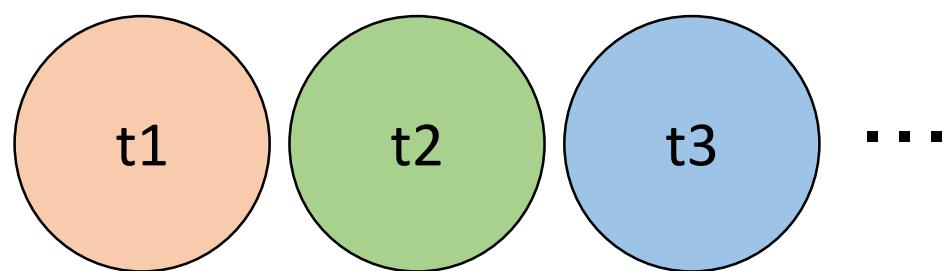
- 1,入力データ一つ選択
- 2,入力データに最も似たベクトルをもつニューロンを探す

結果(競合層)



3,綺麗に並べ替える(学習をする)

入力データ(入力層)

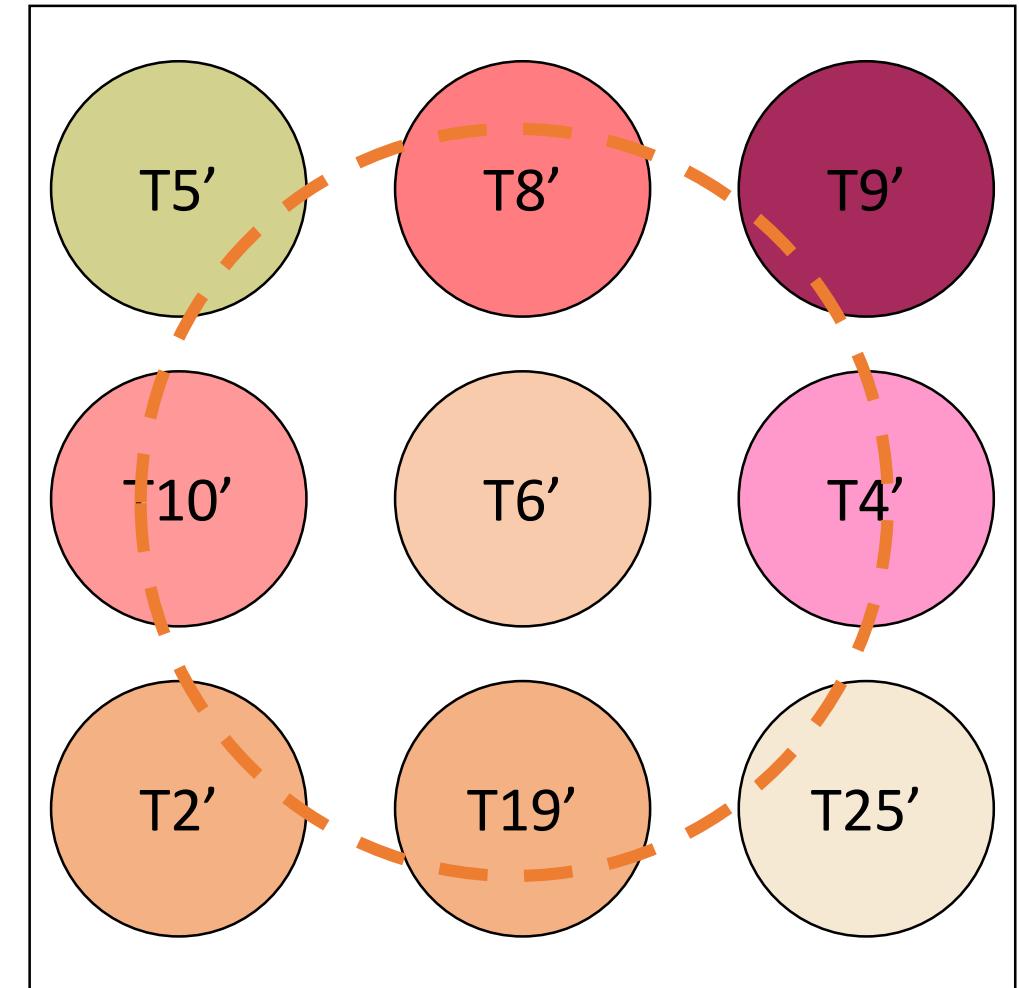


学習の方法

3,周囲にあるニューロンを入力データに近いベクトルにする
4,1~3を繰り返す

入力データ全て学習して:学習一回

結果(競合層)



SOMのアルゴリズム: 理解

1	time	computer	laptop	table	desk	screen	shot	person	room	standing
2	t1	1	1	1	0	0	0	0	0	0
3	t2	1	1	1	0	0	0	0	0	0
4	t3	0	0	0	0	0	0	0	0	0
5	t4	1	1	1	0	0	0	0	0	0
6	t5	1	1	1	0	0	0	0	0	0
7	t6	1	0	1	0	0	0	0	0	0
8	t7	0	1	1	0	0	0	1	0	0
9	t8	1	0	1	0	0	0	0	0	0
10	t9	1	0	1	0	0	0	0	1	0
11	t10	1	1	1	0	0	0	0	0	0
12	t11	1	0	1	0	0	0	1	0	0
13	t12	1	1	1	0	0	0	0	0	0
14	t13	1	1	1	0	0	0	0	0	0
15	t14	1	1	1	0	0	0	0	0	0
16	t15	1	1	1	0	0	0	0	0	0
17	t16	1	1	1	0	0	0	0	0	0

… N次元ある

似たベクトルを持つ

↓

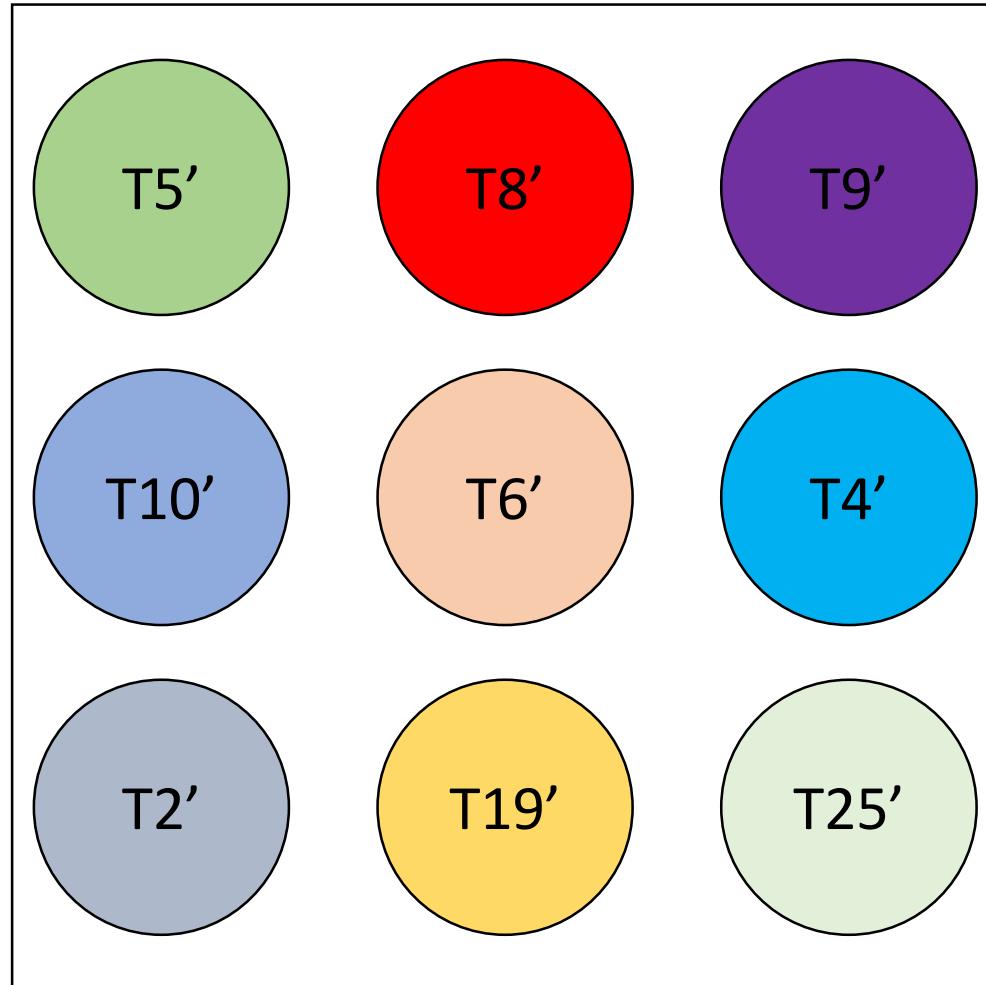
ベクトル同士の距離が近い

N次元

$$d = \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2 + (c_2 - c_1)^2 \dots (n_2 - n_1)^2}$$

(d:距離 n1:入力データのn次元の値 n2:競合層のn次元の値)

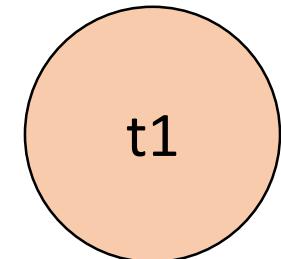
SOMのアルゴリズム: 理解



←この中からdが最も小さくなるものをさがす

↓みつけたら

周囲にあるニューロンを
入力データに近くする



SOMのアルゴリズム: 理解

$$R_{new} = R_1 + (R_2 - R_1) \times \alpha_n \times \beta_n$$

$$\alpha_n = \alpha_0 \times \left(1 - \frac{n}{N}\right) \quad \beta_n = \beta_0 \times \left(1 - \frac{n}{N}\right)$$

R_{new} : ニューロンの新しいベクトル

t8

R_1 : 入力データのベクトル

t1

R_2 : ニューロンのベクトル

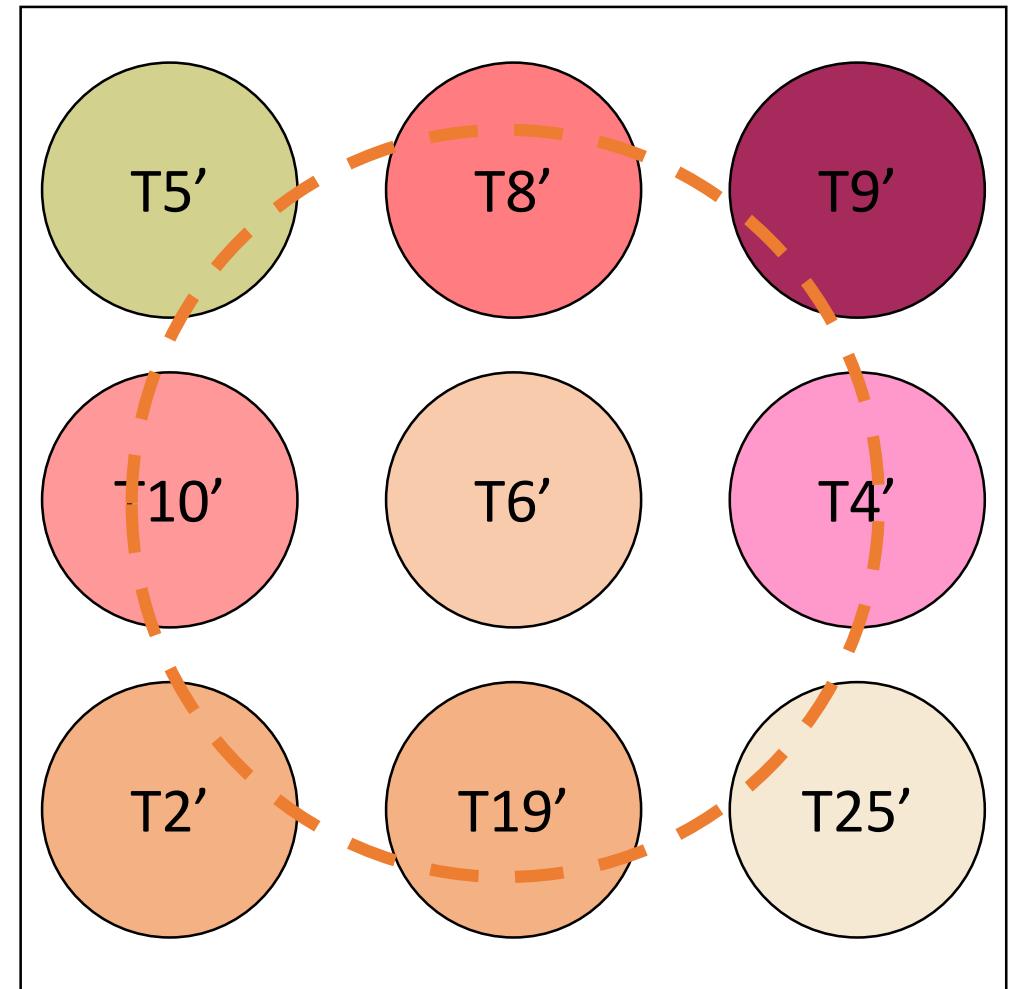
t8

a_0 : 一番最初の近づけ度合

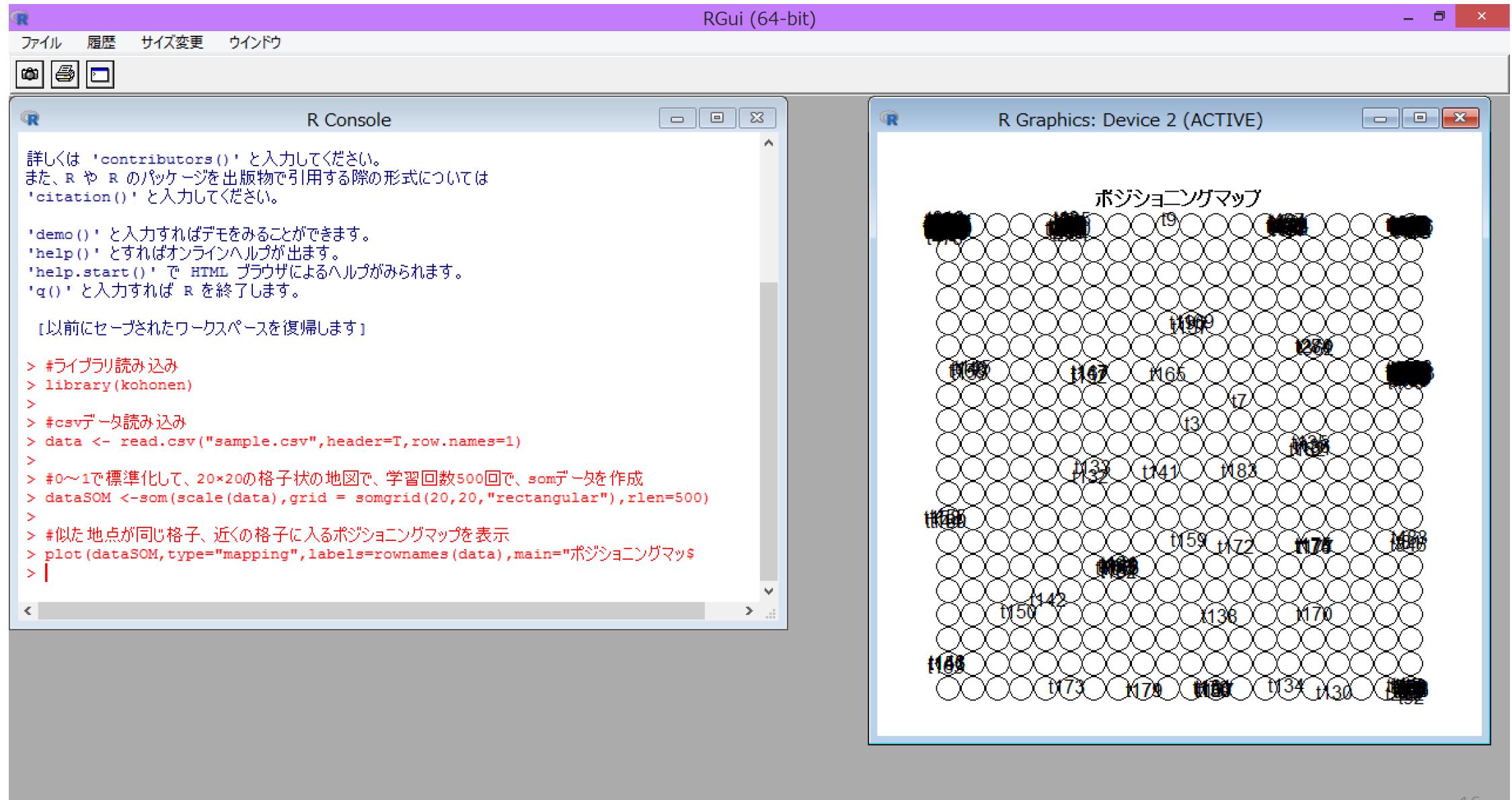
n : 現在の学習回数

N : 全学習回数

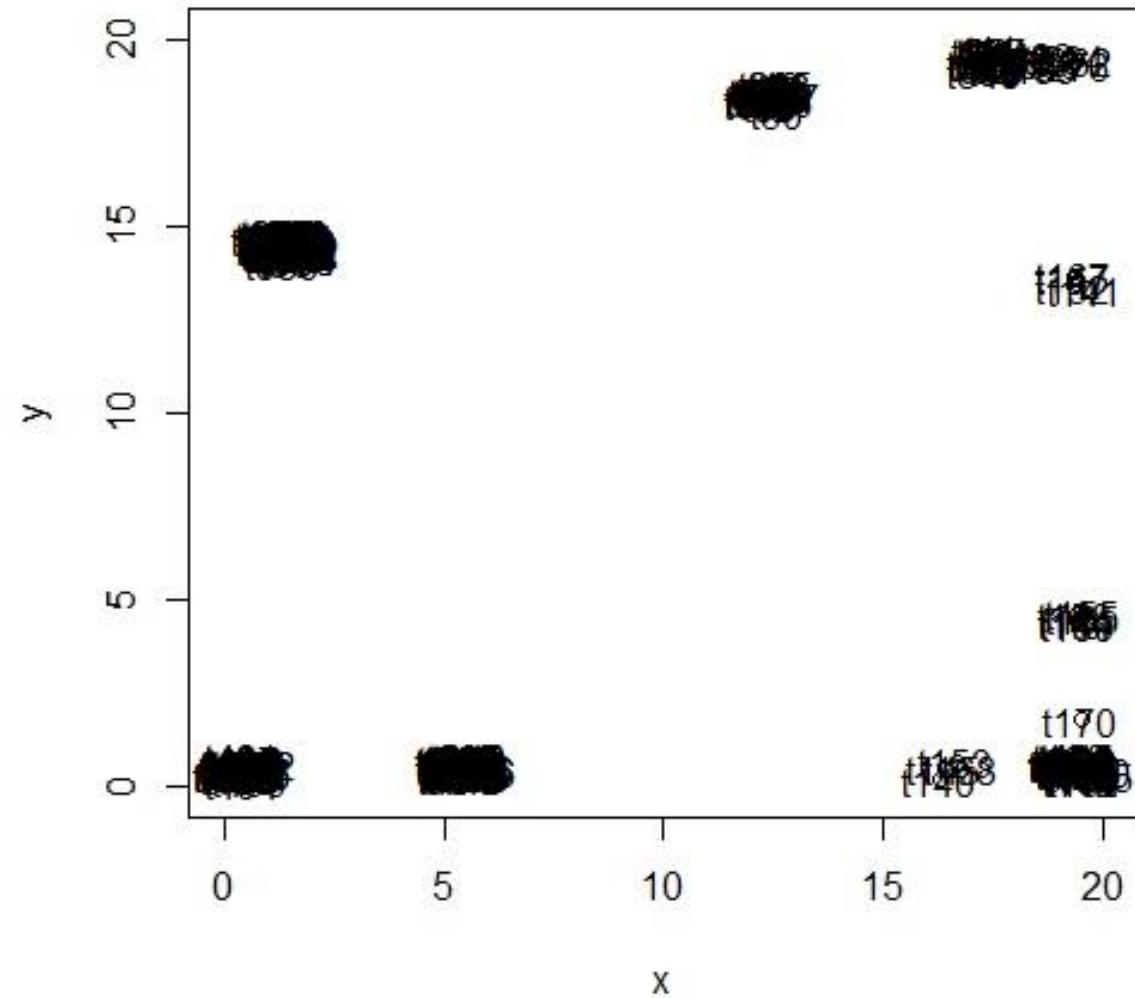
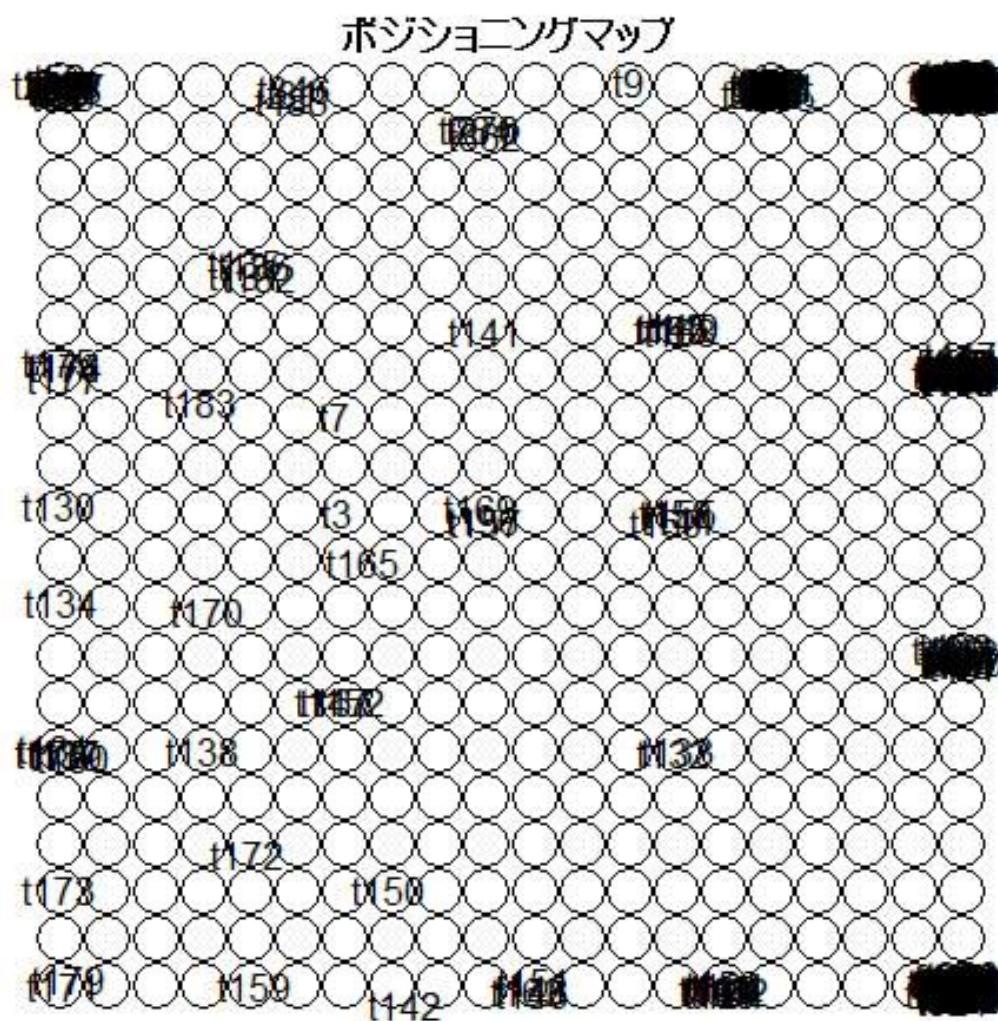
b_0 : 一番最初の周囲



解析結果(R使用)



解析結果(R使用)



今後の展開

- SOM理解→共起ネットワーク
- 解析比較:類似性・イベント検出
- 卒論執筆