数理モデル

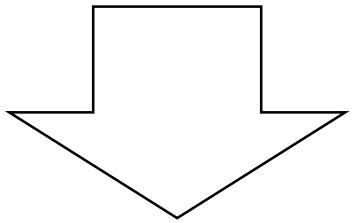
高橋　俊也

Chapter.4 自己組織化マップ

Chapter.4 自己組織化マップ

マーケティングの分野におけるポジショニング

興味の対象となるブランドが競合ブランドと様々な情報により比較されたときの位置づけ

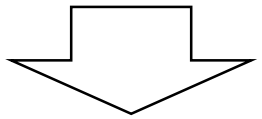


ポジショニングマップを描くためのデータマイニング手法
「コホーネンネット」

4.1 コホーネンネット

SOM（self-organizing map:自己組織化マップ）

→教師なし学習のアルゴリズムを用いるニューラルネットモデル



1984年コホーネン著『自己組織と連想記憶』によって有効性が認められ、「コホーネンネット」などと呼ばれている

4.1.1　多次元データの可視化

SOMでは多次元データを圧縮して低次元（多くの場合2次元）のマップを描く、すなわち多次元データを可視化することが手法の目的である。

第１層に提示された情報は第2層のすべてのユニットに伝えられ、入力情報と接続重みとが、互いにどれだけ似ているかを第2層のユニット間で競争する。
⇒競合学習

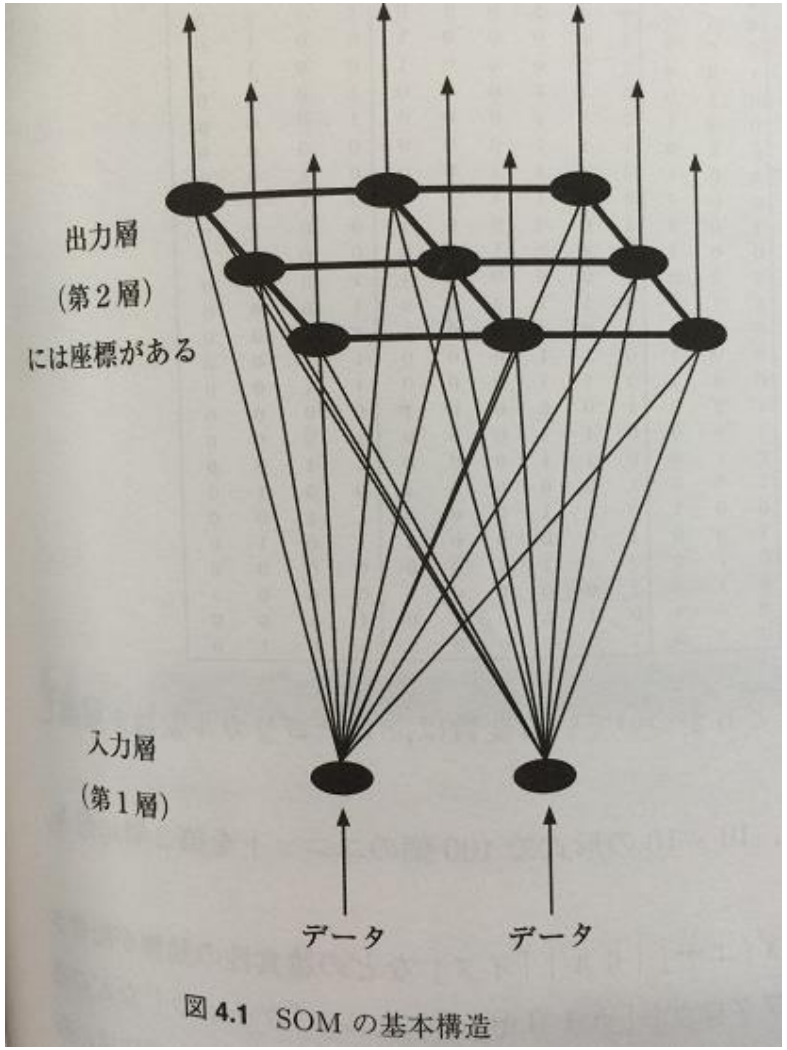


図 4.1　SOM の基本構造

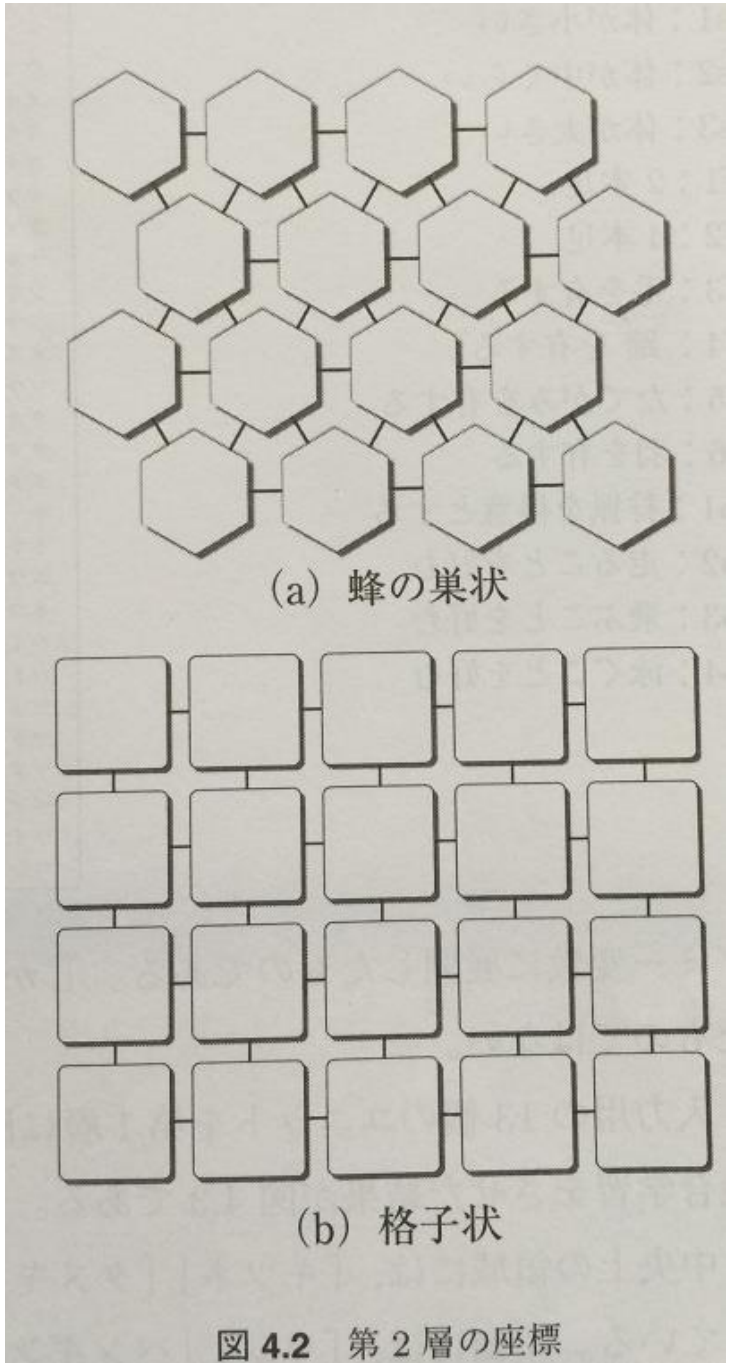


図 4.2　第２層の座標

4.1.1　多次元データの可視化

◎競合学習

SOMの第2層のユニットは1つひとつ明確な位置（座標）を持ち、空間的な拡がりを表現している。

競争の結果一番似ていたユニットは勝者と呼ばれ、そのユニットは重みが調節されて入力情報に近づけられる。さらにその近辺の重みも勝者との近さに応じて同じように調節される。このため隣り合ったユニットは似たような重みをもち第2層のユニットが張る空間に穏やかな距離（位相）が生じる。

4.1.2　動物地図

表4．1は30種の動物(オブザベーション)の特徴をそれぞれ13個のダミー変数で表現した多変量データ。入力用の13個のユニットを第1層に置き10＊10の形式で100個のユニットを第2層に置き、競合学習させた結果が図4．3である。

表 4.1　動物の特徴のダミー変数データ

名前	s1	s2	s3	f1	f2	f3	f4	f5	f6	b1	b2	b3	b4
アヒル	1	0	0	1	0	0	0	0	1	0	0	1	1
イヌ	0	1	0	0	1	1	0	0	0	0	1	0	0
イノシシ	0	1	0	0	1	1	1	0	0	0	1	0	0
ウシ	0	0	1	0	1	1	1	0	0	0	0	0	0
ウマ	0	0	1	0	1	1	1	1	0	0	1	0	0
オオカミ	0	1	0	0	1	1	0	1	0	1	1	0	0
カナリヤ	1	0	0	1	0	0	0	0	1	0	0	1	0
ガチョウ	1	0	0	1	0	0	0	0	1	0	0	1	1
キツネ	0	1	0	0	1	1	0	0	0	1	0	0	0
クマ	0	0	1	0	1	1	0	0	0	1	0	0	0
サル	0	1	0	1	1	1	0	0	0	0	0	0	0
シカ	0	0	1	0	1	1	1	0	0	0	1	0	0

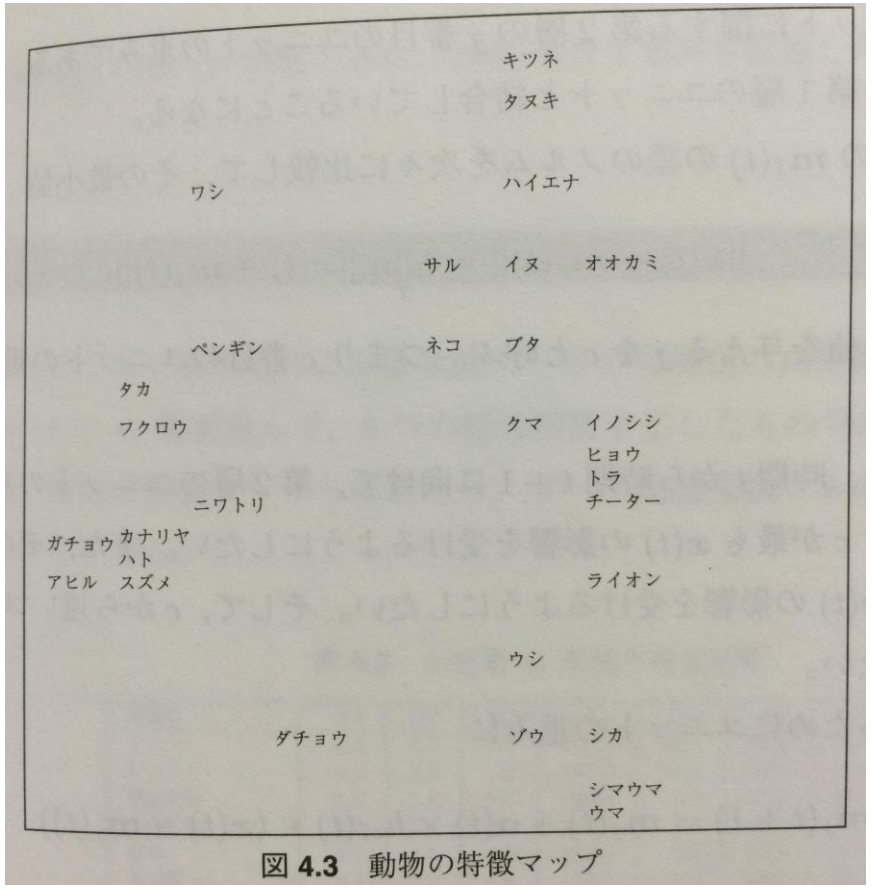


図 4.3　動物の特徴マップ

詳しくは演習にて

4.2　数理モデル

I次元の多変量データがN個あるとする。そこからt番目に取り出した観測変数ベクトルを

$$x(t) = (x1(t)\ x2(t)\ \cdots xi(t)\ \cdots\ xI(t))'$$

とする。
i : 観測変数の添え字、第１層のユニットの添え字（1≦i≦I）
I : 観測変数の数、第１層のユニット数
t : オブザベーションの添え字、時間の添え字

時間tにおける第２層のj番目のユニットの状態は

$$mj(t) = (m1j(t)\ m2j(t)\ \cdots\ mij(t)\ \cdots\ mIj(t))'$$

と表現される。
j : 第２層のユニットの添え字（ユニット数J個）
mij(t) : 第１層のi番目のユニットに関する第２層のj番目のユニットの重み

4.2　数理モデル

ここでx(t)とJ個のmj(t)の差のノルムを次々に比較し、最小値を見つける。

$$\|x(t)-mc(t)\| = \min_{j} \|x(t)-mj(t)\|$$

c：最小値をとるときの j

この場合第2層のc番目のユニットの重みがx(t)に一番似ているということになる。
よって時期t から時期t+1に向けて、第2層の重みを変更する
この時、隣り合ったユニットは似たような重みをもつようにするためにユニットに次のような変更を施す。

$$mj(t+1) = mj(t) + \alpha(t) \times hcj(t) \times (x(t) - mj(t))$$

4.2　数理モデル

$$mj(t+1) = mj(t) + \alpha(t) \times hcj(t) \times (x(t) - mj(t))$$

α(t) , hcj(t)は0以上１以下の重みであり、 α(t) = hcj(t) = １のとき

mj(t+1) = x(t) となりmj(t+1)がx(t)の影響を完全に受けることになる。

→ α(t) , hcj(t)はmj(t+1)に対するx(t)の影響の強さを表現した係数

・ hcj(t) : ユニットcとユニットjの近さを表した関数である。

$$hcj(t) = e^{(-\| rc - rj \|^2 / 2\sigma^2(t))}$$

・rc , ri : ユニットc,jの位置ベクトル
・σ^2(t) : 時間とともに減少する関数

α(t) もσ^2(t)と同様に時間とともに減少する関数である。例えば
α(t) = Max(1 − t/T , 0) が使われ、1000回重みを変更したい場合はT = １000とする

4.6.1 動物マップ（自己組織化マップの描き方）

演習
「animal1-2.csv」を使って図４．３「動物の特徴マップ」のようなマップ
（出力層１０×１０）を作ってみよう

4.6.1 動物マップ（自己組織化マップの描き方）

演習ー1

◎パッケージとデータの読み込み

```
> library(som)
> animalData <- read.csv(“animal1-2.csv”,header=T)
```

・som : 今回のように出力層のユニット数がオブザベーション数を超える場合に使うパッケージ

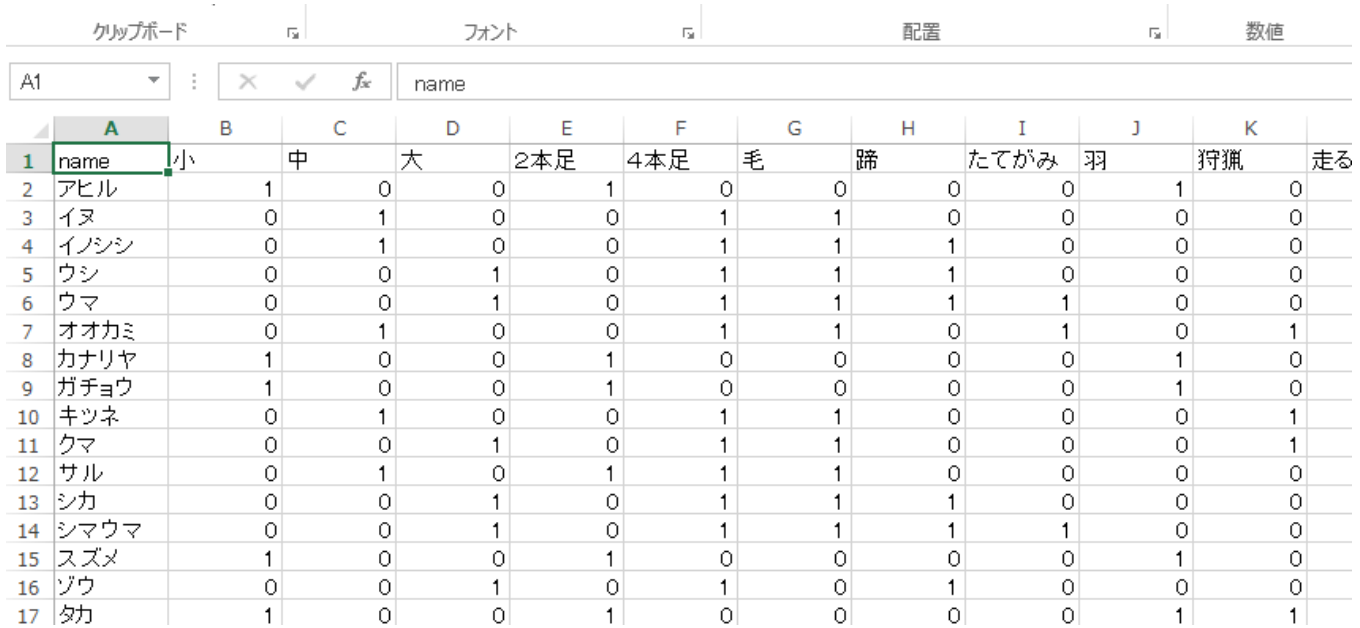
4.6.1 動物マップ（自己組織化マップの描き方）

演習ー2

◎データの標準化

```
> standardData <- normalize(animalData[,2:14],byrow=F)
```

・somに用意されている関数「normalize()」で　byrow=F　とすると（）内のデータを列ごとに平均0、分散1になるように標準化してくれる。



クリップボード　フォント　配置　数値

A1　name

	A	B	C	D	E	F	G	H	I	J	K	
1	name	小	中	大	2本足	4本足	毛	蹄	たてがみ	羽	狩猟	走る
2	アヒル	1	0	0	1	0	0	0	0	1	0	
3	イヌ	0	1	0	0	1	1	0	0	0	0	
4	イノシシ	0	1	0	0	1	1	1	0	0	0	
5	ウシ	0	0	1	0	1	1	1	0	0	0	
6	ウマ	0	0	1	0	1	1	1	1	0	0	
7	オオカミ	0	1	0	0	1	1	0	1	0	1	
8	カナリヤ	1	0	0	1	0	0	0	0	1	0	
9	ガチョウ	1	0	0	1	0	0	0	0	1	0	
10	キツネ	0	1	0	0	1	1	0	0	0	1	
11	クマ	0	0	1	0	1	1	0	0	0	1	
12	サル	0	1	0	1	1	1	0	0	0	0	
13	シカ	0	0	1	0	1	1	1	0	0	0	
14	シマウマ	0	0	1	0	1	1	1	1	0	0	
15	スズメ	1	0	0	1	0	0	0	0	1	0	
16	ゾウ	0	0	1	0	1	0	1	0	0	0	
17	タカ	1	0	0	1	0	0	0	0	1	1	

4.6.1 動物マップ（自己組織化マップの描き方）

演習ー3

◎多次元データをsomで分析しコード情報を表示する

```
> animalSOM <- som(standardData,xdim=10,ydim=10,topol="rect")
```

・xdimとydimで出力層のサイズ、topolで出力層のモデルを選択（”rect”→格子状　“hexa”→蜂の巣状）

⇒コード情報の表示ではそれぞれのユニットにどんな動物が配置されているかわからないのでポジショニングマップを描いてデータを可視化！！

4.6.1 動物マップ（自己組織化マップの描き方）

演習ー4

◎somにより分析したコード情報をポジショニングマップで可視化する

> head(animalSOM$visual)
または
> animalSOM$visual

・各オブザベーションがどのユニットに配置されたかは、animalSOMの中のリスト「visual」に格納されている。
⇒同じユニットに配置されているオブザベーションが複数あるので、plotすると重なってしまって見にくくなる！！

4.6.1 動物マップ（自己組織化マップの描き方）

演習ー5

◎somにより分析したコード情報をポジショニングマップで可視化する

・プロットした時にマップを見やすくするために「乱数」を利用する

```
> rundomNumber <-
cbind(rnorm(nrow(animalData),0,0.15),rnorm(nrow(animalData),0,0.15))
> animalMap <- animalSOM$visual[,1:2]+rundomNumber+0.5
```

・rnorm(n , mean , sd)は任意の平均meanと標準偏差sdを持つ正規分布に従う乱数をn個生成する関数
・nrow()は()内のデータの行数を返してくれる関数
⇒動物の数だけ平均0、標準偏差0．１５の正規分布を2回発生させてcbind()で横にして結合させ、さらに分析結果のx座標y座標に乱数と0.5を足したものを「animalMap」とおいている。

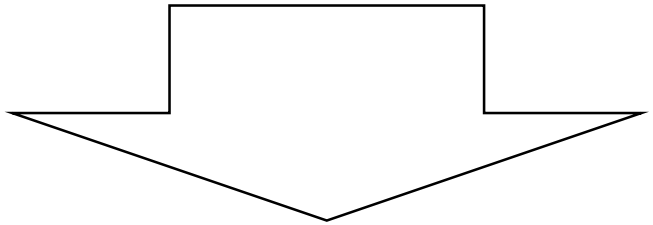
4.6.1 動物マップ（自己組織化マップの描き方）

演習ー6

◎somにより分析したコード情報をポジショニングマップで可視化する

```
> plot(animalMap,xlim=c(0,10),ylim=c(0,10))
> text(animalMap[,1],animalMap[,2],animalData$name)
```

・plotの「xlim」「ylim」でどこまでの範囲を表示するかを決定する（今回は10×10のユニットなのでx,yともに0から10）
・textで動物の名前を重ね書きする。

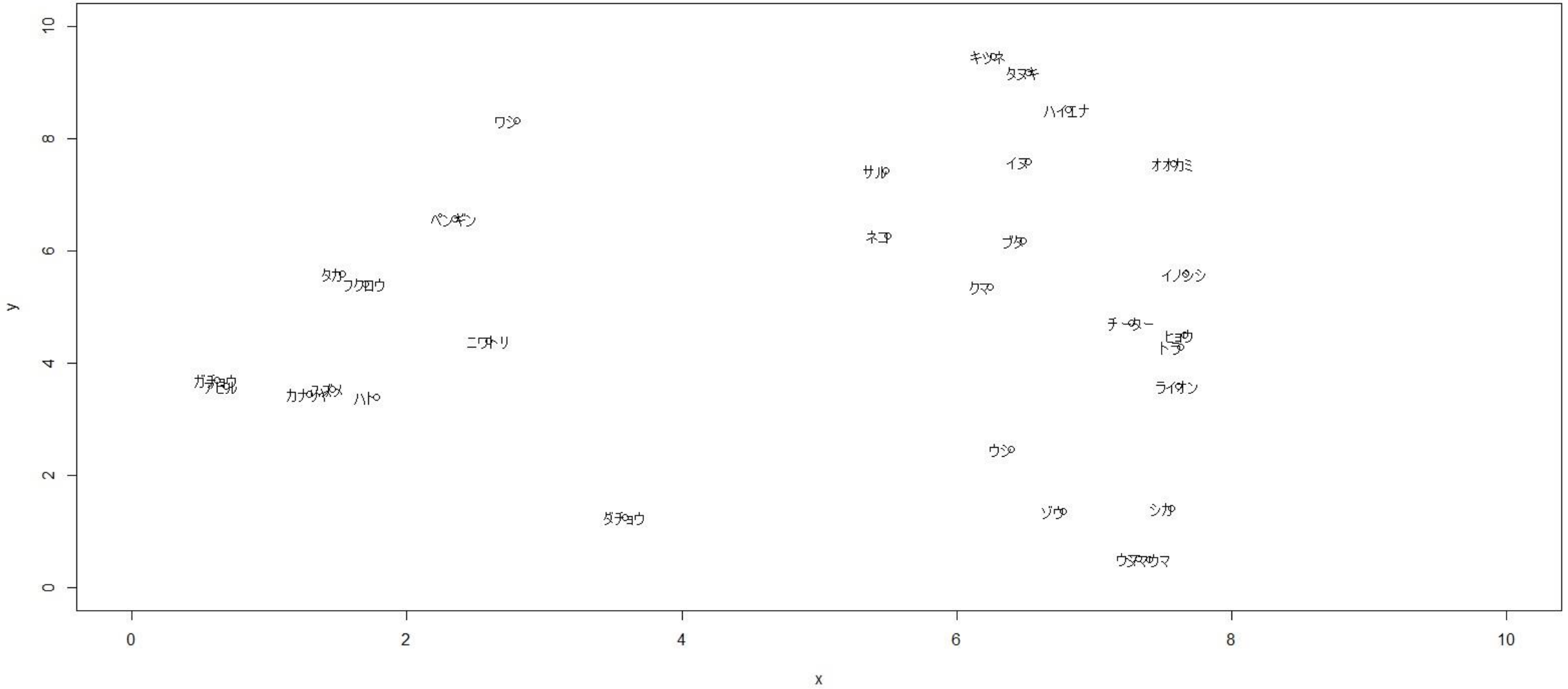


動物マップ完成

RGui (64-bit) - [R Graphics: Device 2 (ACTIVE)]

ファイル　履歴　サイズ変更　ウインドウ

乱数を使用した場合

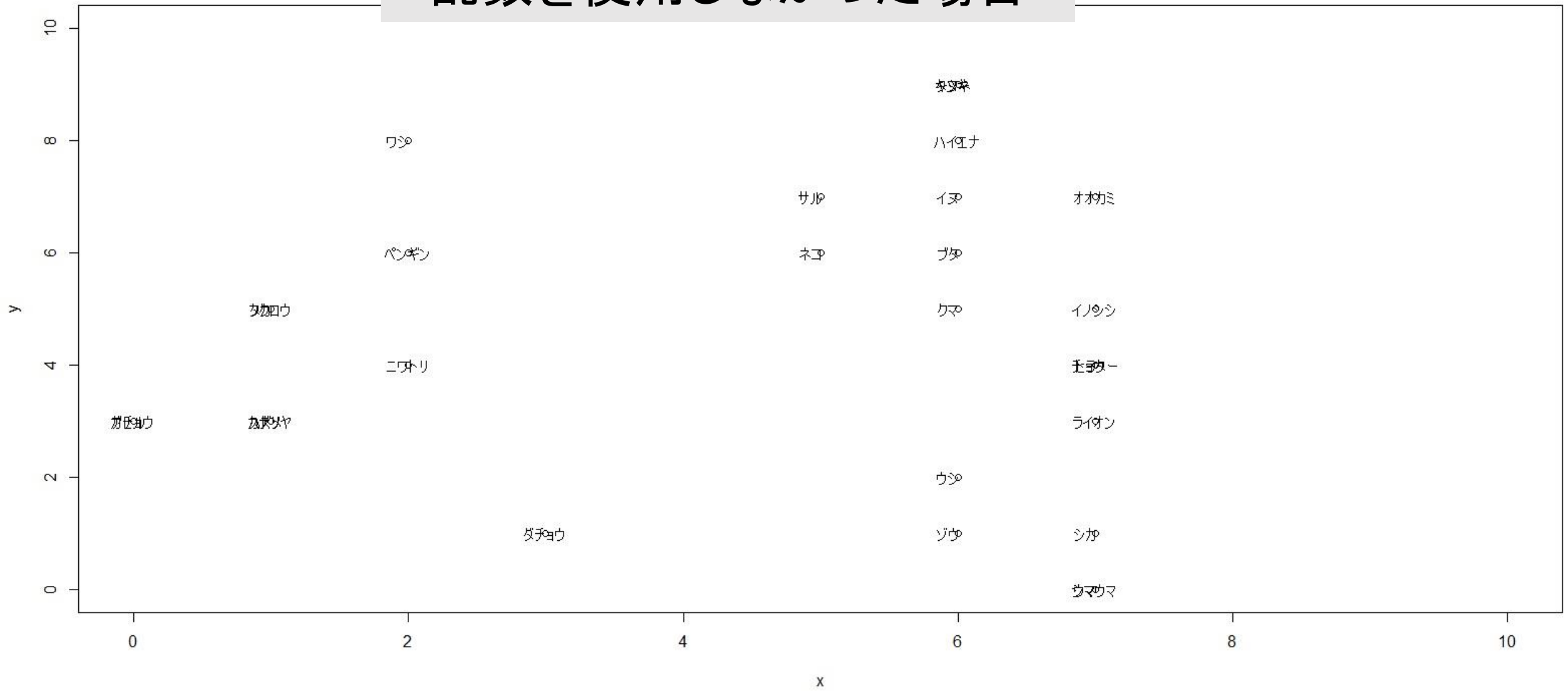


14:31
2014/06/03

RGui (64-bit) - [R Graphics: Device 2 (ACTIVE)]

ファイル　履歴　サイズ変更　ウインドウ

乱数を使用しなかった場合



14:36
2014/06/03

まとめ・感想

・多次元データの情報を圧縮して低次元にして可視化できるSOMはとても便利だということがわかった。

・数理モデルがすごく複雑に思えたがしっかりと順を追って考えれば理解に近づくことがわかった。

宿題

chap4に入ってる「takahashi.csv」を使って特徴マップ（出力層５×５）を描いてみよう。