

自己組織化マップ (SOM)

井手陸斗

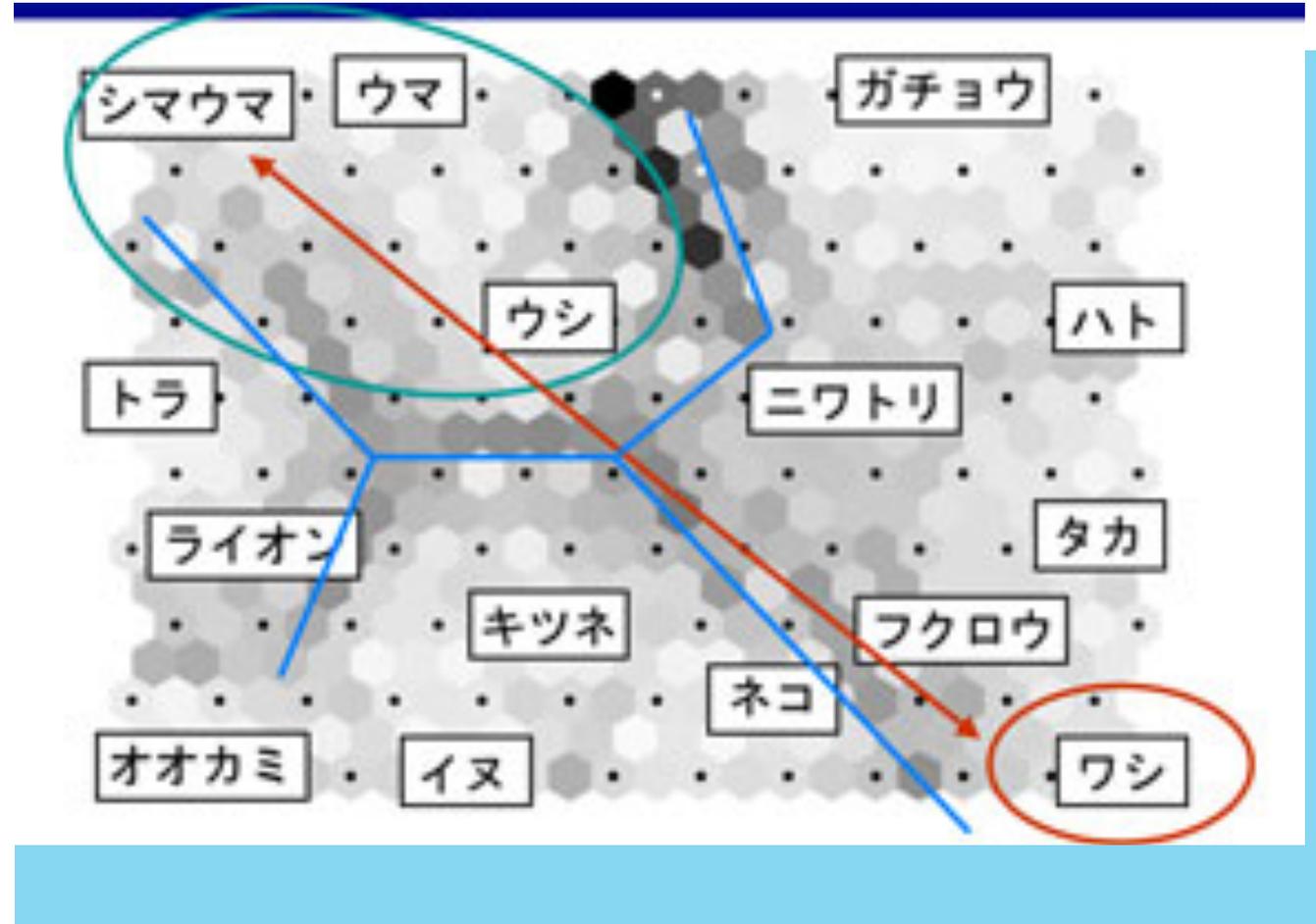
自己組織化マップとは？



自己組織化マップとはTeuvo Kohonenにより提案されたニューラルネットワークアルゴリズム（人間の脳の神経回路の仕組みを模したモデル化したアルゴリズム）であり、多次元データを2次元データ表示できるようにしたもの。

SOMの例 ～動物を自己組織化マップ化～

自分に近い性質を持つものを近くに配置した図で、色が表しているのは隔たりをあらわす。色が濃いとところが大きな隔たりとなる。



自己組織化マップのアルゴリズム

1. 入力 x_j と出力層の全てのユニットを比較し、最も類似しているユニット m_c を探し出し、そのユニットを勝者とする。

$$\|x_j - m_c\| = \min_i \{\|x_j - m_i\|\}$$

2. 探し出したユニットの更新

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x_j(t) - m_i(t)] \quad (i \in N_c)$$

$$m_i(t+1) = m_i(t) \quad (i \notin N_c)$$

ここで、 $h_{ci}(t)$ は近傍関数。

3. 全ての入力ベクトル x_j に対して1と2のステップを反復実行する

SOMはこのアルゴリズムを使って多次元空間上の分類対象を2次元平面に射影する。

パッケージKOHONENのケーススタディ

まず、パッケージkohonenをインストールする。

以下の赤字を入力し、 `library(kohonen)` と入力。

```
> options(repos="http://cran.md.tsukuba.ac.jp")
> install.packages("kohonen")
URL 'http://cran.md.tsukuba.ac.jp/bin/windows/contrib/3.0/kohonen_2.0.14.zi$
Content type 'application/zip' length 778026 bytes (759 Kb)
開かれた URL
downloaded 759 Kb
```

パッケージ 'kohonen' は無事に展開され、MD5 サムもチェックされました

ダウンロードされたパッケージは、以下にあります

```
C:\Users\1256507c\AppData\Local\Temp\RtmpOAAP0y\downloaded_packages
```

関数SOMGRID

次に使う関数somgridの書式は

「`somgrid(xdim=8,ydim=6,topo=c("rectangular","hexagonal"))`」である。

`xdim`=>出力層のサイズとなる列数

`ydim`=>出力層のサイズとなる行数

`topo`=>ユニットの配列の指定。`rectangular`はユニットを格子状に、`hexagonal`はユニットを蜂の巣上に並べる。

IRIS(あやめのガク弁や花弁のデータ)のクラス情報を除いて用いたデータ解析を試みる

以下の赤字を入力。

```
> set.seed(10)
> gr<-somgrid(topo="hexagonal",xdim=10,ydim=7)
> iris.som<-som(as.matrix(iris[,1:4]),gr,rlen=200)
> |
```

ここで、関数somgridの書式は

「somgrid(xdim=○,ydim=○,topo=c("rectangular","hexagonal"))」であるから
ここでは10列7行のサイズでhexagonal配置法(蜂の巣状に結果がplotされる)を用いている。

結果をプロットする。

コードマップと個体マップをプロットし、対照することでパターンの特徴を分析していく。

さて、まずはコードマップから

`plot(iris.com, type="codes")` と入力し、コードマップを出力する。右のような図がプロットされるはず。

右図は各ユニットに配属される個体の特徴を示した星グラフであり、

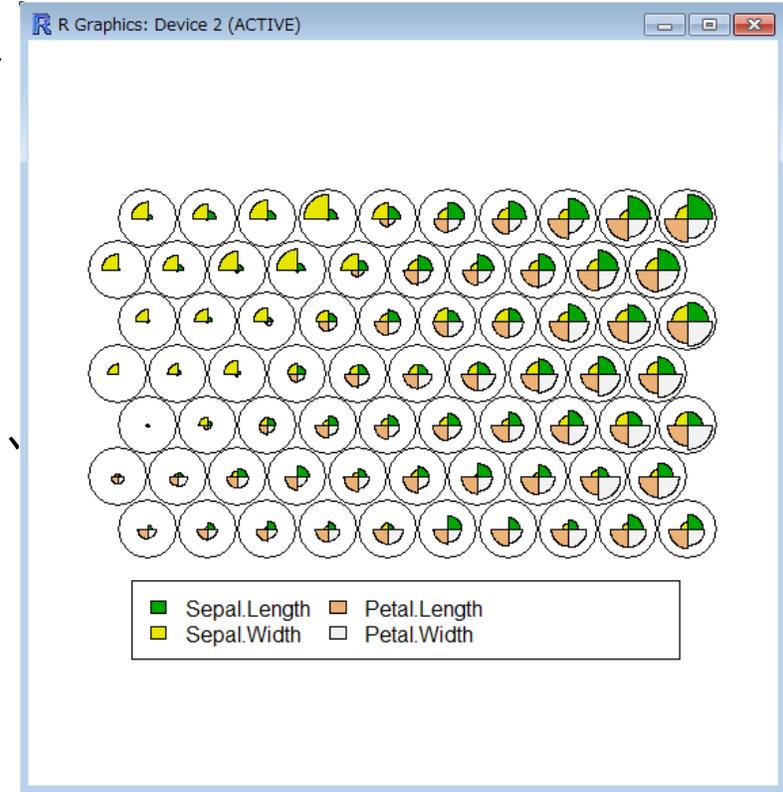
Sepal length=>ガク弁の長さ

Sepal.width=>ガク弁の幅

Petal.length=>花弁の長さ

Petal.width=>花弁の幅

にそれぞれ対応する。



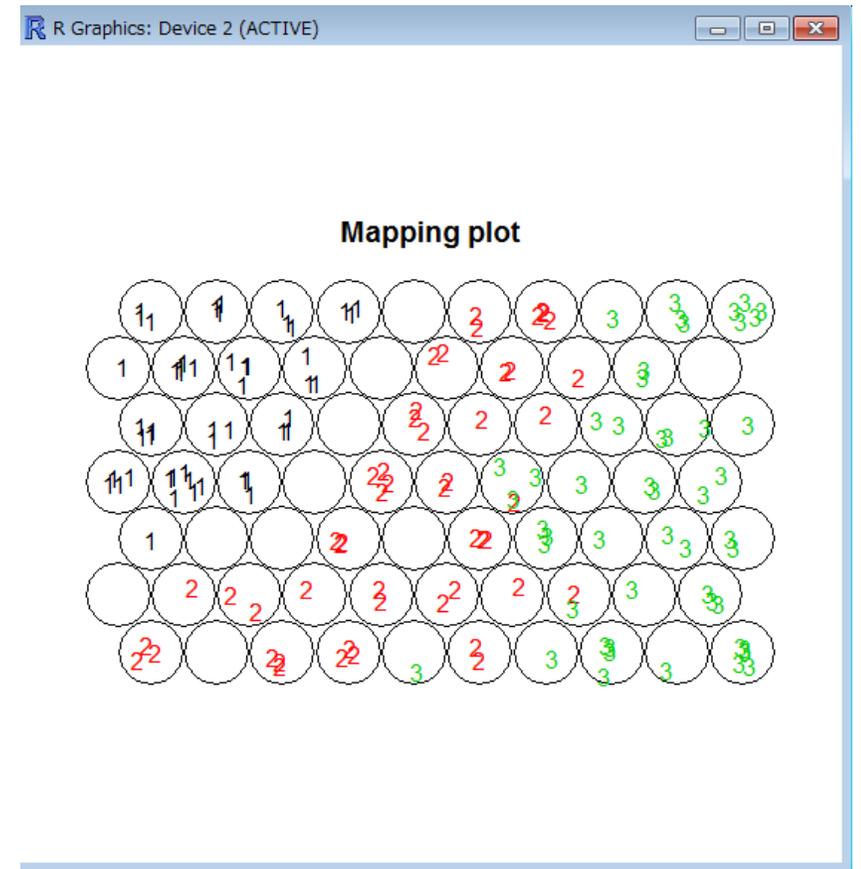
似ているもの同士を同じユニットあるいは、その近辺のユニットに配置してある。

次に、個体マップをプロットする。

```
lab.cod<- as.numeric(iris[,5])
```

```
plot(iris.som, type="mapping",labels=lab.cod,col=lab.cod)
```

と入力しプロットする。



パッケージSOMのケーススタディ

先ほどのパッケージkohonenと同様。

```
options(repos="http://cran.md.tsukuba.ac.jp")
```

```
install.packages("som")
```

と入力しsomのパッケージをインストールした後

```
library(som) する。
```

赤字を入力していく、プロットすると右下の図のようになる

エラーバー付き折れ線グラフ

```
R Console

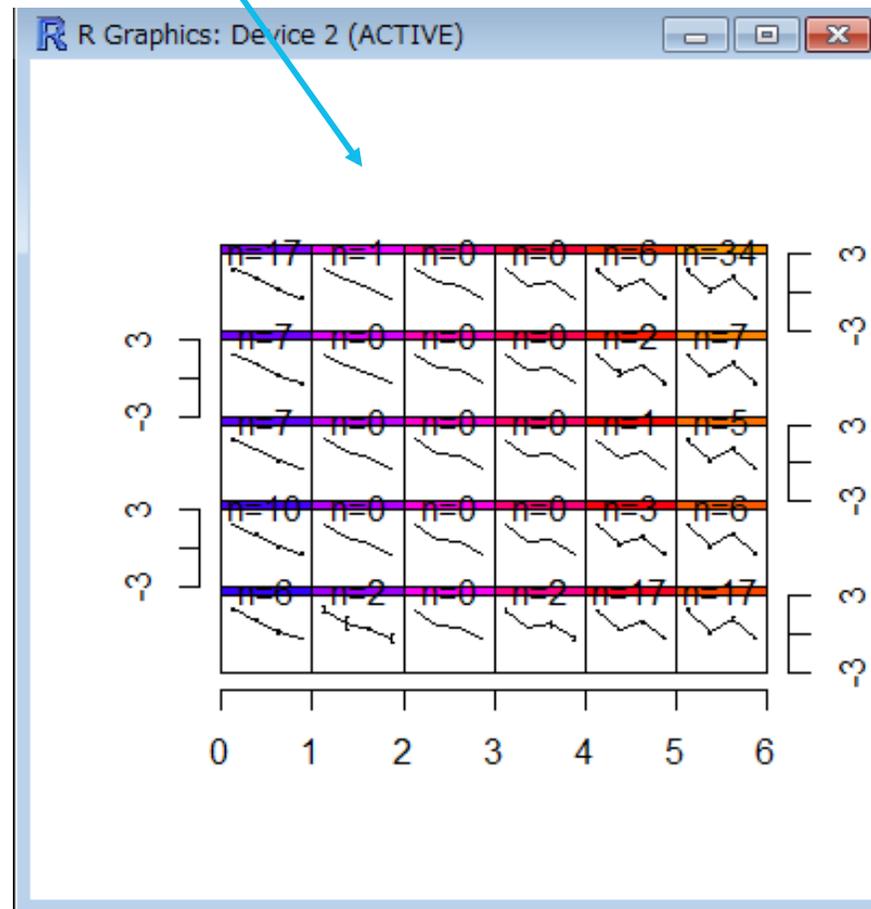
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘルプがみられます。
'q()' と入力すれば R を終了します。

[以前にセーブされたワークスペースを復帰します]

> options(repos="http://cran.md.tsukuba.ac.jp")
> install.packages("som")
URL 'http://cran.md.tsukuba.ac.jp/bin/windows/contrib/3.0/som_0.3-5.zip' を$
Content type 'application/zip' length 651464 bytes (636 Kb)
開かれた URL
downloaded 636 Kb

パッケージ 'som' は無事に展開され、MD5 サムもチェックされました

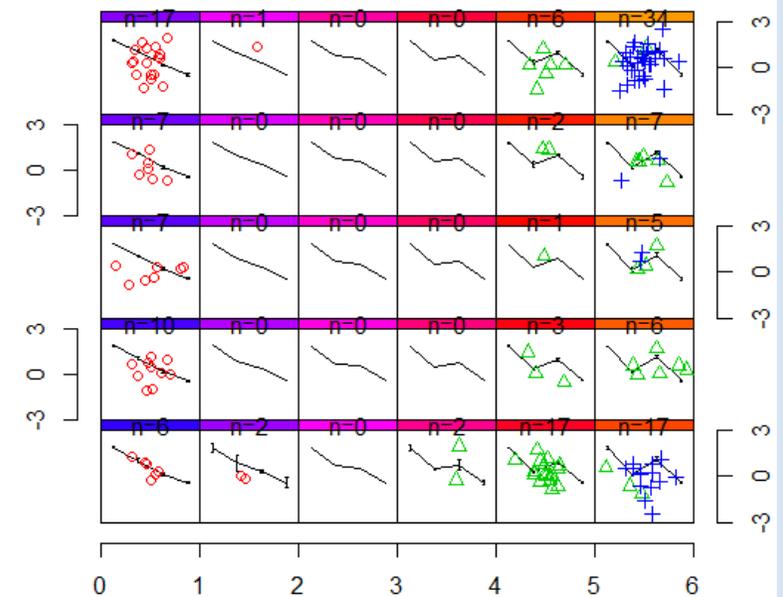
ダウンロードされたパッケージは、以下にあります
C:\Users\1256507c\AppData\Local\Temp\Rtmp0SszXab\downloaded_packages
> library(som)
> set.seed(80)
> iris1<-normalize(iris[,1:4])
> iris.som2<-som(iris1,xdim=6,ydim=5)
> plot(iris.som2)
> |
```



以下の呪文を打ち込んで、前のスライドの折れ線に点を追加する

```
> ransu<-cbind(rnorm(nrow(iris),0,0.13),rnorm(nrow(iris),0,0.13))
> out.new<-iris.som2$visual[,1:2]+0.5+ransu
> points(out.new[,1:2],col=c(2:4)[unclass(iris[,5])],
+ pch=c(1:3)[unclass(iris[,5])])
```

pch=c(1:3)なので、○、△、+の3種類の記号でプロットされる。

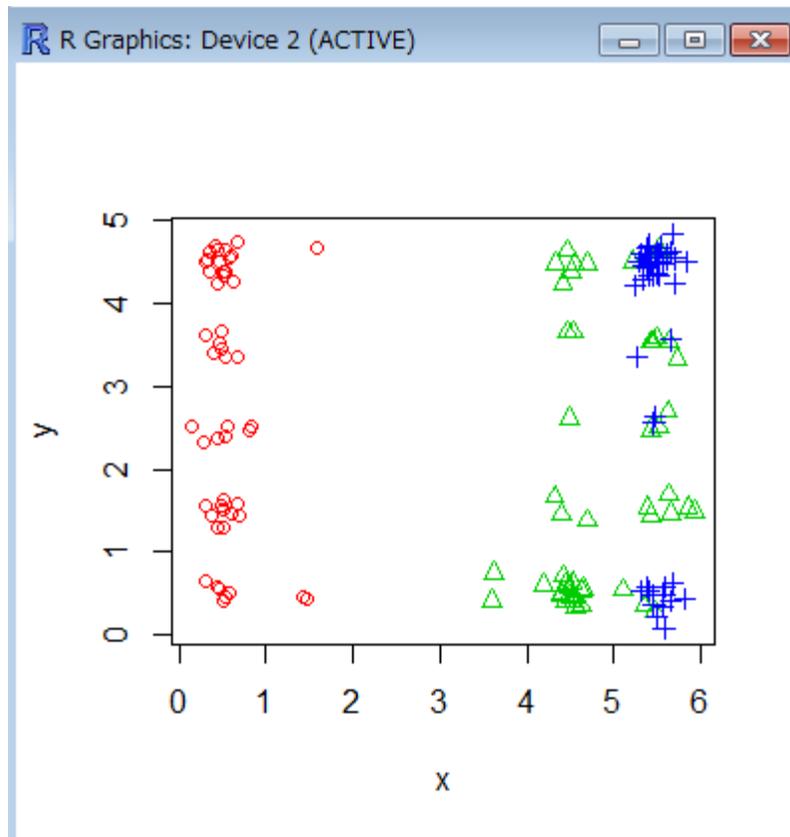


関数SOMの散布図をプロットする

```
plot(out.new[,1:2],col=c(2:4)[unclass(iris[,5])], pch=c(1:3)[unclass(iris[,5])])
```

で散布図（さっきの点）をプロットできる。

以下の図がプロットされる



自己組織化マップのメイン関数はsomであり、

```
som(data,xdim,ydim,...)
```

のように用いる。dataには解析対象のデータセットを
xdim,ydimはそれぞれ出力画面のユニットの列数と行数
を入れる。

今回はxdim=6,ydim=5としているので、左図の通り5列
6行のサイズで出力されている。

○は左端に

△は真ん中あたりに

+は右端に

という風に各個体がだいたい3種類ごとに分かっている。

参考にしたサイト

<http://mj.in.doshisha.ac.jp/R/>

<http://www.okada.jp.org/RWiki/>

http://www.gaya.jp/spiking_neuron/som.htm

→自己組織化についてのわかり易い解説