

## Rとクラスター分析 (1)

同志社大学文化情報学部教授

金 明哲 (Jin Mingzhe)

■中国生まれ。総合研究大学院大学数物研究科統計科学専攻博士後期課程修了。博士(学術)。1995年札幌学院大学社会情報学部、助教授、教授を経て、2005年4月より現職。



### 1. クラスター分析とは

我々は、物事を整理整頓する際には、機能、形状などの側面から似ているものを同じところに集めて、片付ける。これと同じくデータについてもデータ構造の側面から似ている個体と同じグループに仕分けることが必要である場合がある。データサイエンスにおける分類のための方法は、学習(教師、訓練)データがある分類方法と学習データがない方法に大別される。

ここで言う学習データとは、どの個体がどのグループに属するかが既知であるデータである。グループの所属を示すデータは外的基準とも呼ばれている。学習データがある場合の分類方法は、どの個体がどのグループに属するかが既知であるデータから、分類に関するモデルを作成し、そのモデルに基づいて、グループの属性が未知であるデータを最も似ていると判断されるグループに割り当てる判別分析のような方法である。

学習データがない分類方法は、どの個体がどのグループに属するかに関する事前情報がないデータについて、グループ分けする方法で、外

的基準がない分類法である。通常この種の分類方法をクラスター分析と呼ぶ。このクラスター(cluster)とは、花やブドウなどの房の意味で、クラスター分析とは、データの構造が似ている個体と同じ房(グループ)にまとめて、そうでないものを異なる房に集めるデータの処理方法である。

広義では、主成分分析、因子分析、対応分析、多次元尺度もクラスター分析の方法と言える。これらの方法では、散布図を描くことで個体を2次元の平面、あるいは3次元空間上に配置し、クラスター分析を行う。

本稿では、グループの形成状態を房の形で示す樹形図を用いる階層的クラスター分析の方法について紹介する。

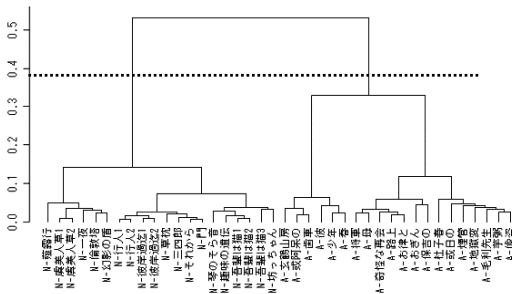
### 2. 階層的クラスター分析

階層的クラスター分析とは、個体間の類似度あるいは非類似度(距離)に基づいて、最も似ている個体から順次を集めてクラスターを作っていく方法である。クラスターが作られていく様子を図1に示すような樹形図で示すことがで

きる。樹形図はデンドログラム (dendrogram) とも呼ばれている。また、階層的クラスター分析はクラスタリング法、凝集型階層手法とも呼ばれている。

図1の樹形図は、芥川龍之介と夏目漱石のそれぞれの20作品における各々の助詞の使用率を用いた樹形図である。

図1 芥川と夏目の作品の樹形図



樹形図は文字どおり木の構造に似たグラフであり、ラベルが付いている部分を葉と言う。葉と葉との距離（葉から上に伸びている線が連結するまでの高さ）が短いほど個体が似ていると言える。樹形図では、幾つかの個体が階層的集まり1つのクラスター（房、枝）を形成し、複数のクラスターが最終的には1つのクラスター（木）となる様子が見て取れる。

また樹形図をある高さ（距離）のところで線を引いて切断することによって、クラスターの数が決まり個体の分類が決まる。例えば、図1において、点線前後の高い位置で樹形図を切断すると、樹形図は芥川氏の作品と夏目氏の作品が分類される。

### (1) 階層的クラスター分析プロセス

階層的クラスター分析には幾つかの方法があるが、いずれも次のようなステップを踏む。

①距離（あるいは類似度）を求める方法を選択

し、個体間の距離（類似度）を計算する。

- ②クラスター分析の方法（最近隣法、最遠隣法など）を選択する。
- ③選択された方法のコーフェン（Cophenetic）行列を求める。
- ④コーフェン行列に基づいて樹形図を作成する。
- ⑤結果について検討を行う。

①では、多次元尺度法を説明する際に示したような距離（あるいは類似度）をデータから求める。②では、次の節で述べる階層的クラスター分析方法を選択する。

### (2) クラスターの形成とコーフェン行列

階層的クラスター分析では、データから距離の行列を求め、距離の行列を用いて樹形図を描くためのコーフェン行列を求め、コーフェン行列に基づいて樹形図を描くというプロセスを経る。

データ行列 ⇒ 距離行列 ⇒ コーフェン行列 ⇒ 結果

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & & & \\ d_{21} & 0 & & \\ \vdots & \vdots & \ddots & \\ d_{m1} & d_{m2} & \cdots & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & & & \\ c_{21} & 0 & & \\ \vdots & \vdots & \ddots & \\ c_{m1} & c_{m2} & \cdots & 0 \end{bmatrix} \Rightarrow \begin{matrix} \text{樹形図} \\ \text{を描く} \end{matrix}$$

距離行列からコーフェン行列を生成する際の第1段階はすべての方法が同じで、最も距離が近い2つの個体間の距離をコーフェン距離とする。第1段階が終わった後、どのようにコーフェン距離を求めるかはクラスター分析の方法によって異なる。

ここでは具体的な例を用いて、そのイメージを説明する。データは、表1に示す7人の5教科の成績データを用いることにする。

表1 成績データ

	算数	理科	国語	英語	社会
田中	89	90	67	46	50
佐藤	57	70	80	85	90
鈴木	80	90	35	40	50
本田	40	60	50	45	55
川端	78	85	45	55	60
吉野	55	65	80	75	85
斉藤	90	85	88	92	95

```
>seiseki<-matrix(c(89,90,67,46,50,
57,70,80,85,90,80,90,35,40,50,
40,60,50,45,55,78,85,45,55,60,
55,65,80,75,85,90,85,88,92,95),
7,5,byrow = TRUE)
>colnames(seiseki)<-c(“算数”, “理科”, “国
語”, “英語”, “社会”)
>rownames(seiseki)<-c(“田中”, “佐藤”, “鈴
木”, “本田”, “川端”, “吉野”, “斉藤”)
```

このデータのユークリッド距離を次に示す。

```
>seiseki.d<-dist(seiseki)
>round(seiseki.d)

      田中  佐藤  鈴木  本田  川端  吉野
佐藤   69
鈴木   34   81
本田   60   64   53
川端   28   61   21   47
吉野   63   12   76   54   56
斉藤   68   38   88   92   68   46
```

このような距離データからコーフェン行列の計算過程の主なステップを説明する。

上記の距離行列では、最も距離が近いのは、吉野と佐藤の12である。そこでこの2人がまずクラスターc1{吉野,佐藤}を形成する。

次に距離の値が小さいのは川端と鈴木との距離である。川端と鈴木が1つのクラスター{川端,鈴木}を形成すべきか、それともc1に川端を加えたクラスター{c1,川端}, c1に鈴木を加えたクラスター {c1,鈴木} が新しいクラスターを形成すべきであるかは何らかの決まりに従って計算する必要がある。

説明の便利のため、この7人の距離関係を図2

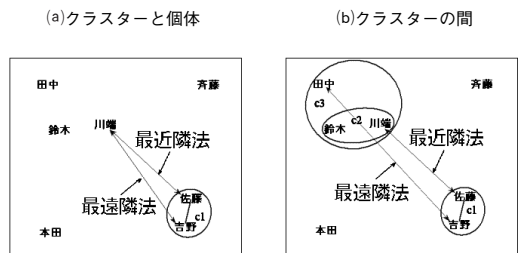
のように平面上で示すことができると仮定する。

川端と鈴木の距離は既に分かっているので求める必要がない。問題は、c1と川端の距離、c1と鈴木との距離をどのように求めるかである。c1と川端の距離を考える場合でも図2(a)に示すように幾つかの方法が考えられる。

この問題では、どのような方法を採用しても明らかに、川端と鈴木の距離が短いので新しいクラスターc2は {川端,鈴木} となる。このようにc2に田中が加えられ新たなクラスターc3 {c2, 田中} = {川端,鈴木,田中} が形成される。階層的クラスター法はこのように階層的クラスターを形成する。

クラスターを形成する際には、クラスターとクラスターとの間の距離を計算しなければならない問題に遭遇する。例えば、クラスターc1とクラスターc3との距離を考えてみよう。クラスター間の距離は図2(b)に示すように、クラスター間で最も近い個体同士の距離をクラスター間の距離にするか(最近隣法)、最も距離が遠い個体同士の距離をクラスター間の距離にするか(最遠隣法)など、クラスター間の距離の求め方は唯一ではない。このようなクラスター間の距離を求める方法の違いで、階層的クラスター分析の方法が異なる。このように何らかの方法で求めた距離行列がコーフェン行列である。階層的クラスター分析の樹形図はコーフェン行列に基づく。

図2 コーフェン距離



(3) 階層的クラスター分析の諸方法

階層的クラスター分析の方法はクラスター間の距離をどのように求めるかで、最近隣法、最遠隣法、群平均法、メディアン法、重心法、ウォード法などに分かれる。

①最近隣法

最近隣法 (nearest neighbor method) は、最短距離法、単連結 (single linkage) 法とも呼ばれる。最近隣法は、2つのクラスターのそれぞれの中から1個ずつ個体を選んで個体間の距離を求め、それらの中で、最も近い個体間の距離をこの2つのクラスター間の距離とする方法である。

②最遠隣法

最遠隣法 (furthest neighbor method) は、最遠距離法、完全連結 (complete linkage) 法とも呼ばれる。最遠隣法は、最近隣法とは逆に、2つのクラスターの中のそれぞれの中から1個ずつ個体を選んで個体間の距離を求め、それらの中で、最も遠い個体間の距離をこの2つのクラスター間の距離とする方法である。

③群平均法

群平均法 (group average method) は、最近隣法と最遠隣法を折衷した方法で、2つのクラスターのそれぞれの中から1個ずつ個体を選んで個体間の距離を求め、それらの距離の平均値を2つのクラスター間の距離とする。

④重心法

重心法 (centroid method) は、クラスターのそれぞれの重心 (例えば、平均ベクトル) を求め、その重心間の距離をクラスター間の距離とする。重心を求める際には、クラスターに含

まれる個体数が反映されるように、個体数を重みとして用いる。

⑤メディアン法

メディアン (median method) 法は、重心法の変形で、2つのクラスターの重心の間の重み付きの距離を求めるとき、重みを等しくして求めた距離の値を、2つのクラスター間の距離とする。

⑥ウォード法

ウォード法 (Ward's method) は、2つのクラスターを融合した際に、群内の分散と群間の分散の比を最大化する基準でクラスターを形成していく方法である。ウォード法は最小分散法 (minimum variance method) とも呼ばれる。

これらのコーフェン行列作成過程で、新しいクラスターを形成する際のクラスター間の距離は次の式で求める。式の中の  $d_{ij}$ ,  $d_{(ij)k}$ ,  $d_{ik}$ ,  $d_{jk}$  は図3に示すクラスター間の距離で、 $\alpha_i$ ,  $\alpha_j$ ,  $\beta$ ,  $\gamma$  はパラメータ (係数) である。パラメータと上述の方法との対応関係を表2に示す。

$$d_{(ij)k} = \alpha_i d_{ik} + \alpha_j d_{jk} + \beta d_{ij} + \gamma |d_{ik} - d_{jk}|$$

図3 クラスター間の距離

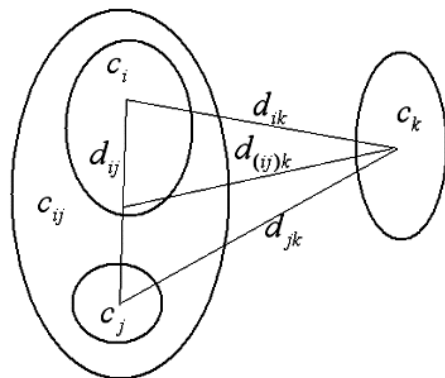


表2 方法とパラメータとの対応表  
( $n_i$  クラスタ  $c_i$  の個体数)

方法の名称	$\alpha_i$	$\alpha_j$	$\beta$	$\gamma$
最近隣法	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
最遠隣法	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
群平均法	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
重心法	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$-\alpha_i \alpha_j$	0
メディアン法	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0
ウォード法	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{n_j+n_k}{n_i+n_j+n_k}$	$\frac{n_k}{n_i+n_j+n_k}$	0

これら以外にも、可変 (flexible) 法、McQuitty法、重みつき群平均法などがある。

### 3. 階層的クラスタ分析のR関数

#### (1) 関数hclust

Rのパッケージstatsには階層的クラスタ分析の関数**hclust**がある。その書き式を次に示す。

`hclust(d, method = "complete", ...)`

引数dは、距離構造のデータで、引数methodにはクラスタ分析の方法を指定する。デフォルトにはcomplete法が指定されている。関数hclustで扱う方法とその引数を表3に示す。

表3 方法の名称と引数の対応

引数の文字列	方法の名称
single	最近隣法
complete	最遠隣法
average	群平均法
centroid	重心法
median	メディアン法
ward	ウォード法
mcquitty	McQuitty法

関数hclustを用いてクラスタ分析を行う際に必要となる幾つかの関数を表4に示す。

表4 関連関数と解析結果

名称	機能
summary	結果のオブジェクトのリストを返す
plot	樹形図を作成する
plclust	樹形図を作成する
cutree	クラスタ (房) の数を指定し、グループ分けする
cophenetic	コーフェン行列を返す

#### (2) 関数hclustの使用例

##### ①関数hclustの結果

まずユークリッド距離を用いた関数hclustの使用例を示す。データは、表1のデータ (seiseki) を用いる。

```
>seiseki.d<-dist(seiseki)
>(seiseki.hc<-hclust(seiseki.d))
```

```
Call:
hclust(d = seiseki.d)
Cluster method : complete
Distance      : euclidean
Number of objects: 7
```

返された結果から、用いた方法は“complete” (最遠隣法)、距離は“euclidean” (ユークリッド) であることが分かる。また、用いたデータの個体数に関する情報も返される。

関数summaryで、結果のオブジェクトに格納されたリストを確認することができる。

```
>summary(seiseki.hc)
      Length Class Mode
merge      12 -none- numeric
height      6 -none- numeric
order       7 -none- numeric
labels      7 -none- character
method      1 -none- character
call        2 -none- call
dist.method 1 -none- character
```

mergeは、クラスター形成の履歴のマトリクスである。

```
>seiseki.hc$merge
  [,1] [,2]
[1,] -2 -6
[2,] -3 -5
[3,] -1  2
[4,] -7  1
[5,] -4  3
[6,]  4  5
```

このデータは、階層的クラスターを形成する階層の情報である。マイナス符号が付いているのが個体の番号、マイナス符号が付いていないのがクラスターの番号である。行番号がクラスター形成の順番である。

用いた最遠隣法 (complete) では、ステップ 1 (第 1 行) は、個体 2 (佐藤) と個体 6 (吉野) がクラスター 1 を形成する。ステップ 2 (第 2 行) は、個体 3 (鈴木) と個体 6 (川端) がクラスター 2 を形成する。ステップ 3 (第 3 行) は、個体 1 (田中) とクラスター 2 が新しいクラスター 3 を形成する。

height はクラスターを形成する際の、樹形図の枝の長さ (高さ) を返す。

```
>seiseki.hc$height
[1] 12.40967 21.30728 33.77869 45.58509
60.13319 91.53142
```

この値は、mergeの結果と対応する。例えば、個体 2 (佐藤) と個体 6 (吉野) の距離 (枝の長さ) は12.40967である。

orderは樹形図の左から右方向の個体の番号を返し、labels は個体のラベル、methodは用い

た方法、callは用いた関数hclustの書き式、dist.methodは用いた距離の方法の名称を返す。

```
>seiseki.hc$order
[1] 7 2 6 4 1 3 5
```

## ② 樹形図

関数plotあるいはplclustを用いてhclustの結果の樹形図を作成することができる。次に関数plotを用いた4つの樹形図を示す。引数hang=-1はラベルの高さを揃える。

```
>par(mfrow=c(2,2))
>plot(sei.hc,main="最遠隣法")
>plot(sei.hc,hang=-1,main="最遠隣法")
>s.hc2<-hclust(sei.d,method="centroid")
>plot(s.hc2,hang=-1,main="重心法")
>s.hc3<-hclust(sei.d,method="ward")
>plot(s.hc3,hang=-1,main="ウォード法")
```

図 4 ユークリッド距離の樹形図

