研究ノート

R の hclust 関数による 正しい ward 法分析の方法
～その２：Ward[2] を R で独自実装する～

木村　敦｜Kimura Atsushi

㈱統計センター 理事・CIO

■ NTT にて ICT 関連開発に長年携わり、㈱ NTT ファシリティーズ総合研究所 取締役情報技術本部長を経て、2019 年 4 月から現職。1988 年 3 月名古屋大学大学院理学研究科博士課程（前期）修了、修士（理学）、専門統計調査士。

1. はじめに

前回、「その 1」として ward 法クラスター分析を R の hclust 関数を用いて行う場合の正しい使い方について述べた。検証実験のために Ward [2] に厳密に則って独自に作成した ward 法プログラムによる結果との比較を行った [1]。今回は、その独自プログラムの設計から実装までを詳細に解説する。独自実装プログラムのソースコードは、誌面の都合上、エストレーラ Web[注1)] に全て掲載した。理論からプログラムへの繋がりの実例として参考になれば幸いである。なお本稿における見解は筆者個人のものであり、本文章の誤記や誤りなどはすべて筆者の責に帰する。

注 1）㈶統計情報研究開発センター HP（https://www.sinfonica.or.jp/）内の［刊行物］＞［エストレーラ］＞［参考］に掲載。

2. ward 法の実装設計詳細

⑴　ward 法の概要と計算式

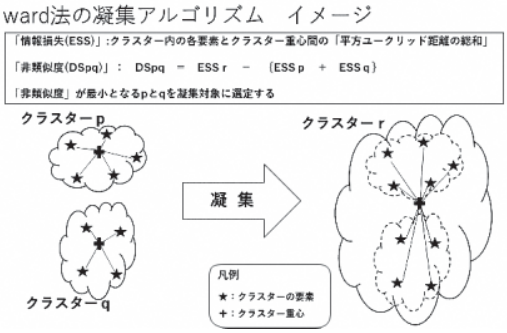

図 1　ward 法の概要

クラスター p に含まれる要素数を Np、クラスター p の重心ベクトルを $\boldsymbol{c}_p$ とする。クラスター p と q の ward 非類似度を $DSpq$ とすると、

$$DSpq = \{Np \cdot Nq/(Np + Nq)\} \times \|\boldsymbol{c}_p - \boldsymbol{c}_q\|^2 \quad (2\text{-}1)$$

初回のクラスター i（＝要素 i）と j（＝要素 j）の ward 非類似度（$DSij$）は、$Ni = Nj = 1$ であることから、

$$DSij = (1/2) \times \| \boldsymbol{c}_i - \boldsymbol{c}_j \|^2 \qquad (2\text{-}2)$$

また、クラスター p と q が凝集した新しいクラスター r と、それ以外のクラスター s の ward 非類似度（$DSrs$）は、計算済みの $DSps$, $DSqs$, $DSpq$ から

$$DSrs = \{1/(Np + Nq + Ns)\} \times \{(Np + Ns) \times DSps + (Nq + Ns) \times DSqs - Ns \times DSpq\} \qquad (2\text{-}3)$$

と算出できる。つまり、初回の ward 非類似度行列さえ出来上がれば、後は式 (2-3) の更新式を用いて各工程の非類似度行列を計算することができる。

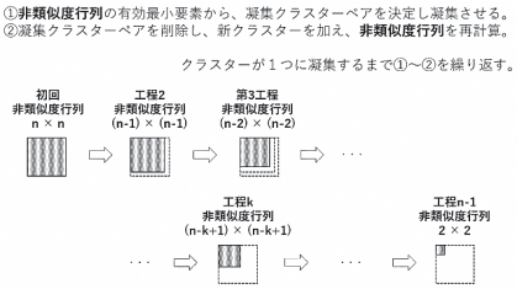

図２　ward 非類似度行列の更新

図２に示すように、凝集工程が進むにつれて非類似度行列のサイズは１つずつ小さくなる。

⑵　プログラムで扱う変数の種類と形式

せっかく独自に実装するのであるから、既存のツールの関数と差別化して、凝集工程の状態を全て保存して後から参照可能にしよう。毎工程変化してゆく非類似行列等を工程毎に後から参照できるようにしたい。このためには、保持する変数の種類と変数の構造について十分吟味して設計を行っておく必要がある。

各凝集工程において２つのクラスターが消滅し、新たなクラスターが１つ誕生することになる。また、式 (2-3) の計算では非類似度の他に各クラスターに含まれている要素数が必要となる。これらの動的に変化してゆく必要情報を整理して、途中の計算で利用できるようにする必要がある。

大まかな処理の流れを具体的に追いながら、実装において必要となる変数の種類と構造について洗い出してゆくことにしよう。全体の要素数が n 個で、各要素がそれぞれ m 種類の変量で表現されている多変量データ（n 行×m 列）を分析する場合で考える。この場合、最初に n 個あったクラスターが $n-1$ 回凝集して最終的に１つのクラスターにまとまる。初期のクラスターが n 個ある状態を工程１、最終的にクラスターが１つにまとまった状態を工程 n と呼ぶことにする。

①工程１（初回）におけるクラスターの状態保持

クラスターを識別する番号を「**クラスター識別番号**」と呼ぼう。工程１では要素１つ１つを１つのクラスターとみなすので、要素 i からなるクラスターを識別番号 i と定めれば良いだろう。

各要素がどの識別番号のクラスターに含まれているかを表す変数を「**クラスターベクトル**」と定義する。n 次元ベクトルである。全工程でのクラスターベクトルを保持できるように $\boldsymbol{CV[k,i]}$ なる２次元配列の「**クラスターベクトル行列**」を変数として用意する。$CV[k,i]$ は、工程 k において要素 i が含まれているクラスター識別番号を示す。工程１（$k = 1$）におけるクラスターベクトル行列は、

$$CV[1,i] = (1,2,3,\ldots,n-1,n)$$

である。

次に、各工程において存在（残存）しているクラスター識別番号を保持する変数として「**残存クラスター識別番号ベクトル**」を定義する。こちらも n 次元ベクトルとして設計しておこう。***CNV*[*k*,*i*]** なる 2 次元配列「**残存クラスター識別番号ベクトル行列**」を変数として用意する。クラスターの凝集が進むにつれて消滅するクラスターと新たに凝集して誕生するクラスターが出現することになるため、その状況を保持する変数である。残存するクラスターの識別番号の小さいものから順にベクトルの値として並べる。工程 1 では、

$$CNV[1,i] = (1,2,3,\ldots,n-1,n)$$

である。凝集工程が進むにつれて残存するクラスターの数は 1 つずつ減少するので、工程 k において、残存するクラスターの総数は $n-k+1$ 個である。i が n から $n-k+2$ の範囲においては、$CNV[k,i]$ に値 0 を入れることに決めておこう。

次に、残存クラスターに含まれる要素数を保存する「**残存クラスター要素数ベクトル**」を用意する。変数として ***CNNV*[*k*,*i*]** なる「**残存クラスター要素数ベクトル行列**」を定義する。

$$CNNV[1,i] = (1,1,1,\ldots,1,1)$$

残存するクラスターは必ず要素数が 1 以上である。残存する識別番号が $CNV[k,i]$ であるクラスターの要素数を N とすれば、$CNNV[k,i]$ には値 N が保持されることになる。また、$CNV[k,i]$ と同じく、i が n から $n-k+2$ の範囲における $CNNV[k,i]$ にも値 0 を入れることにしよう。

② 工程 1 における非類似度行列の計算と保持

非類似度の初回の計算であり、更新式 (2-3) を使うことはできない。式 (2-2) を用いて全ての組み合わせについて非類似度の計算を行うことになる。

分析したい多変量データ（標準化などの前処理が完了しているもの）から、初回の非類似度行列を作成するまでの手順を図で説明する。多変量データの構造と初回のクラスター i の重心ベクトルを図 3 に示す。

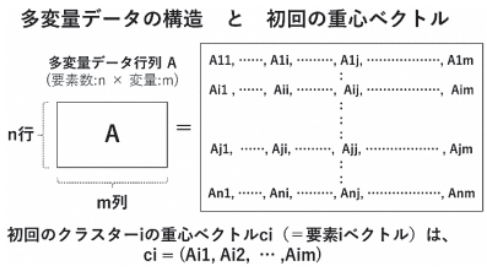

図 3　多変量データの構造

平方距離行列の作成イメージを図 4 に示す。

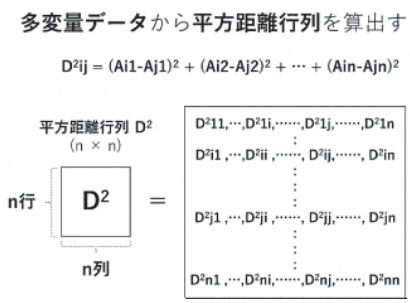

図 4　平方距離行列の作成

プログラム中では平方距離行列は R の dist 関数の出力結果の正方行列の各要素を二乗することによって作成することができる。dist 関数のデフォルト出力はユークリッド距離なので取り扱いには注意が必要である。

平方距離行列が出来上がったら、式 (2-2) に基づいて初回の ward 非類似度行列を作成する。式 (2-2) の説明でも述べた通り、初回におけるクラスター i（＝要素 i）と j（＝要素 j）の ward 非類似度（$DSij$）はクラスターに含まれ

る要素数が $Ni = Nj = 1$ であることから、各クラスター重心間（=要素間）の平方ユークリッド距離を 1/2 倍することによって作成することができる。このイメージを図 5 に示す。初回の ward 非類似度行列は対角要素がすべて 0 の $n \times n$ 対称行列である。

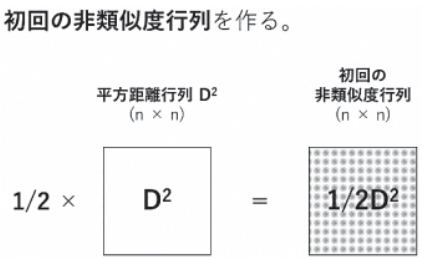

図 5　初回の ward 非類似度行列

工程 1 の ward 非類似度行列を $DisS[1,i,j]$ なる 3 次元データで保存することにしよう。工程 k における ward 非類似度行列は $DisS[k,i,j]$ に保存されることになり、後で任意の工程における ward 非類似度行列を参照することが可能となる。この $DisS[k,i,j]$ をプログラム変数として**「非類似度行列キューブ」**と呼ぶことにしよう。

③ 凝集クラスターの選定

先ほど計算した非類似度行列 $DisS[1,i,j]$ の有意な要素（対角項や工程が進むことにより便宜上 0 を挿入した項を除いた要素）の最小値が $DisS[1,p,q]$ であったとする（$p<q$ としよう）。この場合は次の工程 2 でクラスター p（$= CNV[1,p]$）と q（$= CNV[1,q]$）が凝集することになるが、この $DisS[1,p,q]$ を保存するために変数を用意する。この変数を $\boldsymbol{minDS[k]}$ と定義する。工程 k における非類似度行列 $DisS[k,i,j]$ の有意な要素の最小値を保存するものである。R の hclust 関数の出力で言えば height と同じものである。工程 1 では、

$$minDS[1] = DisS[1,p,q]$$

である。工程 k において考えると、$DisS[k,i,j]$ の有意な要素の最小値が $DisS[k,s,t]$ であったとき、この s と t に該当するクラスター識別番号は、$CNV[k,s]$ 及び $CNV[k,t]$ に保持されている。工程が進んでゆくと行列の添え数字 s や t が必ずしもクラスター識別番号とは一致しないので注意が必要である。凝集するクラスター識別番号のペアは、後のデンドログラム描画で必要となるので $CNV[k,s]$ と $CNV[k,t]$ を保存しておく。（独自プログラムでは $\boldsymbol{CVmerge[k,h]}$（ここで、$\boldsymbol{h}$ は 1 または 2）を保存変数として定義している。R の hclust 関数の返り値は hclust クラスの S3 オブジェクトである。$CVmerge[k,h]$ は、S3 オブジェクトの hclust クラスの merge と仕様を合わせている。）

以上ここまでを、工程 1 とする。

④ 工程 2 でのクラスターの消滅と誕生

工程 2 におけるクラスター凝集で、クラスター p と q が消滅し、新たなクラスターが 1 つ誕生する。工程 2 で生成される新たなクラスターの識別番号に $n+1$ を付与することにしよう。つまり、新しく誕生したクラスターには前工程までに付与されたクラスター識別番号の最大の値に 1 を加えたものとする。各工程で新たなクラスターが 1 つ誕生することになる。工程 k において誕生するクラスターのクラスター識別番号は $n+k-1$ となる。

⑤ 工程 2 での凝集後の各要素とクラスターとの関係及びクラスター要素数

工程 2 では、凝集した要素 p と要素 q 以外については、要素 i はクラスター i に含まれたままである。それらのクラスターの要素数も全

て1のままである。要素pと要素qは新たに誕生したクラスター$n+1$に含まれ、クラスター$n+1$の要素数は2である。従って、

$$CV[2,i] = i \ (i \neq p \text{ かつ } i \neq q \text{ のとき})$$
$$= n+1 \ (i = p \text{ もしくは } i = q \text{ のとき})$$

$$CNV[2,i] = (1,\ldots,p-1,p+1,\ldots,q-1,q+1,\ldots,n+1,0)$$

$$CNNV[2,i] = (1,1,1,\ldots\ldots\ldots\ldots\ldots\ldots,1,2,0)$$

変数の定義で決めた通り、$CNV[2,n]$と$CNNV[2,n]$には0を入れる。

⑥ 工程2での非類似度行列の計算

工程2以降での非類似度行列の計算においては更新式(2-3)を用いることができる。$DisS[2,i,j]$, $minDS[2]$, $CVmerge[2,h]$を保持して、工程2は終了である。

以後、④から⑥と同様な処理を繰り返すことによって工程を進め、工程nで④相当の最後のクラスター凝集を行って終了である。

以上の主なプログラム変数について、構造を図に表すと以下のようになる。

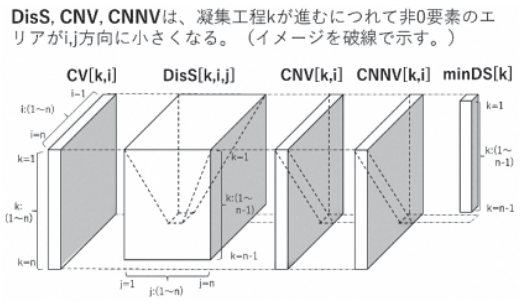

図6　主な変数の立体構造

各変数におけるkの取りうる範囲はCV, CNV, $CNNV$についてはkが1からn、$DisS$, $minDS$, $CVmerge$の3つの変数においてはkが1から$n-1$までである。これは、工程nにおいてはクラスターが1つに凝集してしまうため、さらなる凝集クラスターペアは存在しないし、非類似度行列の計算を行うことも無く、このため最小非類似度も存在しないためである。

⑦ 各変数の更新処理の具体的イメージ

図6に示している通り、$DisS$, CNV, $CNNV$は有効要素のサイズが1つずつ減少してゆく。

この処理をプログラムで実装するのは意外に厄介である。プログラムを読み解く際に最も分かり難いところなので、少し丁寧に解説しておきたい。工程kにおける各変数と、工程$k+1$における各変数の変化を図示して説明しよう。

工程kの非類似度行列$DisS[k,i,j]$の有意な最小要素が$DisS[k,s,t]$だとしよう。工程$k+1$では、クラスター識別番号$CNV[k,s]$とクラスター識別番号$CNV[k,t]$の2つのクラスターが凝集し、新たなクラスター識別番号$n+k$のクラスターが生成されることになる。これを考慮し$CNV[k+1,i]$を確定させることになる。図7に$CNV[k+1,i]$と$CNV[k,i]$との関係を示す。凝集してクラスター識別番号$n+k$となった$CNV[k,s]$と$CNV[k,t]$を削除して、残りの要素を詰める。$CNV[k+1,n-k]$には新しく生成されたクラスターのクラスター識別番号$n+k$を保持する。残りの$CNV[k+1,n-k+1]$から$CNV[k+1,n]$には全て0が入る。

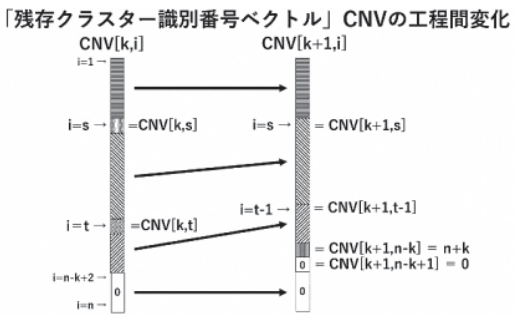

図7　CNV[k,i]とCNV[k＋1,i]の関係

さらに、*CNNV*[*k*,*i*] と *CNNV*[*k* + 1,*i*] の関係も、*CNV*[*k*,*i*] と *CNV*[*k* + 1,*i*] の関係と同様に図で示しておこう。図 8 に示すように、*CNNV*[*k*,*s*] と *CNNV*[*k*,*t*] を削除して残りの要素を詰める。*CNNV*[*k* + 1,*n* − *k*] には新しく生成されたクラスター識別番号 *n* + *k* の要素数 *CNNV*[*k*,*s*] + *CNNV*[*k*,*t*] を保持する。*CNNV*[*k* + 1,*n* − *k* + 1] から *CNNV*[*k* + 1,*n*] は全て 0 である。

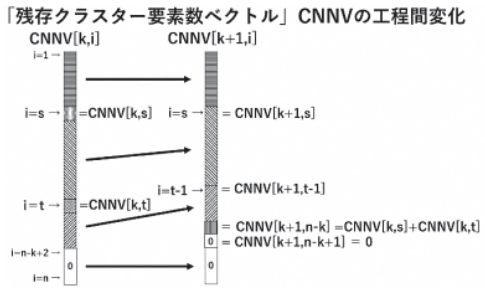

図 8　CNNV[k,i] と CNNV[k + 1,i] の関係

次に *DisS*[*k*,*i*,*j*] と *DisS*[*k* + 1,*i*,*j*] の関係を示す。

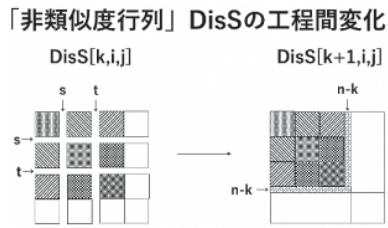

図 9　DisS[k,i,j] と DisS[k + 1,i,j] の関係

図 9 の通り、*k* + 1 工程においてクラスター識別番号 *CNV*[*k*,*s*] と *CNV*[*k*,*t*] が凝集することから、*DisS*[*k*,*i*,*j*] の *s* 列と *t* 列、及び *s* 行と *t* 行を削除して残りの要素を詰めて *DisS*[*k* + 1,*i*,*j*] の 1 ≦ *i* ≦ *n* − *k* − 1 及び 1 ≦ *j* ≦ *n* − *k* − 1 の部分を作成する。*DisS*[*k* + 1,*i*,*j*] の *n* − *k* 列と *n* − *k* 行には、凝集により新しく生成されたクラスター識別番号 *CNV*[*k* + 1,*n* − *k*] のクラスターとその他の残存クラスターとの非類似度が保存される。この非類似度は更新式 (2-3) を用いて算出する。*n* − *k* + 1 行以上と *n* − *k* + 1 行以上には該当する残存クラスターが無いため、要素として 0 を入れる。*DisS*[*k*,*i*,*j*] の *u* 行と *u* 列が示すクラスター識別番号は *CNV*[*k*,*u*] であり、*v* 行と *v* 列が示すクラスター識別番号は *CNV*[*k*,*v*] であることに注意が必要だ。同様に *DisS*[*k* + 1,*i*,*j*] の *u* 行と *u* 列が示すクラスター識別番号は *CNV*[*k* + 1,*u*] であり、*v* 行と *v* 列が示すクラスター識別番号は *CNV*[*k* + 1,*v*] である。工程が進むにつれて動的に変化する。

⑶　plot 関数によるデンドログラム描画

デンドログラムを描画する plot 関数はジェネリック関数で、plot 関数に引き渡されるオブジェクトのクラスに応じた適切なメソッドが適用される。hclust 関数の返り値は S3 オブジェクトの“hclust”クラスであり、対応する plot メソッドが plot.hclust である。独自実装でも plot 関数が利用できるように、分析結果を hclust クラス S3 オブジェクトに編集する処理を最後に加えている。

3. おわりに

以上、独自作成 ward 法プログラムに関する設計から実装に至るまでの詳細説明を行った。本稿を参考に R プログラムを是非読んでみて欲しい。各種分析プログラムを独自に設計作成する際の参考になれば幸いである。

＊参考文献
[1] 木村敦（2022）「R の hclust 関数による正しい ward 法分析の方法 ～その 1：ward 法の特徴と検証実験～」『ESTRELA』No.337, pp.22-27.
[2] Joe H. Ward, Jr.(1963)“Hierarchical Grouping to Optimize an Objective Function”, Journal of the American Statistical Association, Vol. 58, No. 301(Mar., 1963), pp.236-244.