VII-3-2. 階層的クラスター分析

VII-3-2-1. 連結法

VII-3-2-1-1. デンドログラム（樹形図）

階層的クラスター分析では、類似度が高い(非類似度が低い)もの同士をつなげて積み上げていって階層構造を作ります。図 127 に階層構造のデンドログラム（樹形図）の作り方を示しました。２変数で表されるデータ A,B.C,D,E があったとします。各ステップの左側がデータの散布図、右側がデンドログラムです。階層的クラスター分析では類似度の高い（非類似度の低い）データからつないでいきます。step 1 の散布図を見ると、AB の間の距離が小さく非類似度が低いので、AB を一つのクラスとして AB をつなぎます。この時、AB をつなぐ横線の高さを AB の距離（非類似度）とします。Step2 では AB のクラスの代表点$\mathrm{M_{AB}}$を決めて、$\mathrm{M_{AB}}$,C,D,E の間で転換の距離を比較します。この例では CD 間の距離が最も短いので、CD を一つのクラスとします。CD をつなぐ横線の高さは CD の距離（非類似度）です。step3 では、CD のクラスの代表点を決めて、$\mathrm{M_{AB}}$, $\mathrm{M_{CD}}$,E の間で距離を比較します。この例では$\mathrm{M_{AB}}$, $\mathrm{M_{CD}}$間の距離が最も短いので、AB、CD のクラスをまとめて一つのクラスにします。二つのクラスを結ぶ横線の高さは$\mathrm{M_{AB}}$, $\mathrm{M_{CD}}$間の距離（非類似度）です。最後の step4 では、ABCD のクラスと E をまとめます。ABCD のクラスの代表点と E の距離（非類似度）が ABCD のクラスと E を結ぶ横線の高さです。最後にクラスを分ける非類似度の閾値を決めます。Step4 のデンドログラムの緑の鎖線で示したように、閾値を$|M_{AB}M_{CD}|$と$|M_{ABCD}E|$の間にとるとクラスターはクラス ABCD とクラス E の二つに分かれます。赤い鎖線のように$|CD|$と$|M_{AB}M_{CD}|$との間にとると、クラスターはクラス AB、クラス CD、クラス E の３つに、紺の鎖線のように$|AB|$と$|CD|$との間にとると、クラスターはクラス AB、クラス C、クラス D、クラス E の４つに分かれます。閾値をどこに置くかは分析者の判断です。

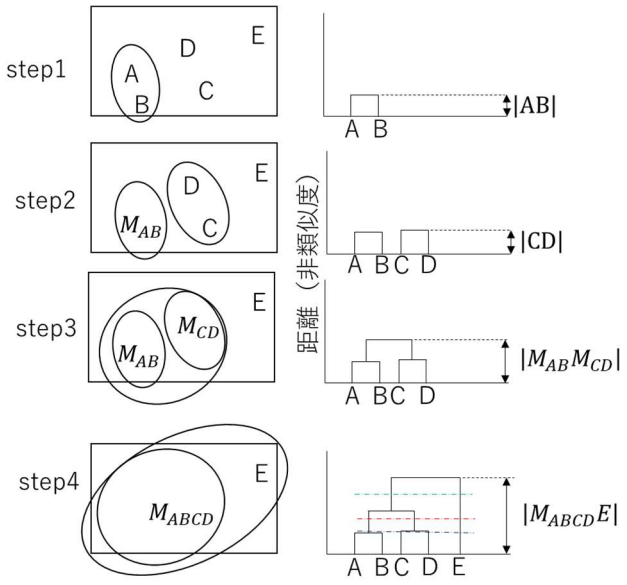


図 127. デンドログラム（右）の作り方

階層的クラスター分析とはデンドログラムを作って階層構造を明らかにする分析です。距離（非類似度）をどのように定義するか、途中経過的にできるクラスの代表点（M）を何にして連結するかについて様々な考え方があり、その組み合わせで、階層的クラスター分析には様々な方法があります。ここでは、まず代表的な連結法として、単連結法、完全連結法、群平均法、重心法、ウォード法について説明しますが、おそらくもっとたくさんあると思いますが、筆者もそのすべてを知っているわけではありません。

VII-3-2-1-2 単連結法(single linkage method)

単連結法(single linkage method)は最短距離法(minimum distance method)という名前の方が一般的です。単連結法ではクラスター間の距離を、それぞれのクラスター間で最も近い点の距離とします。この定義にしたがってあるクラス(C_a)とあるクラス(C_b)を結合すると、新たにクラス($C_{a\cup b}$)と他の$d(C_a, C_k)$との距離$d(C_{a\cup b},\ C_k)$は

$$d(C_{a\cup b}, C_k) = \min\left(d(C_a, C_k), d(C_a, C_k)\right)$$

となります。つまり、元のクラスターとC_kとの距離の小さい方が新たなクラスターと,C_kとの距離になります。数式で書かれるとわかりにくいという人のために、図 128 に具体的な手順を示しました。散布図に A から E までの点があります。各点の座標はその下の表のとおりです。最初のステップでは各点間の距離の総当たり表を作ります。その中で最も距離が近い組み合わせ（A,B）を一つのクラスとしてまとめます。距離(非類似度)は二つの点の距離$\sqrt{(2-1)^2+(2-1)^2} = \sqrt{2} \approx 1.414$です。第二ステップでは、クラス AB と他の点との距離を AB どちらかの点と近い方の距離とします。この場合、C,D,E のいずれとも近い点は B ですから、クラス AB と C,D,E の点の距離は、2, 4.123, 4.472 です。こうしてできた step2 の距離の総当たり表の中で最も距離が近いのはクラス AB と点 C ですから、クラス ABC を作ります。クラス AB と C の距離(非類似度)は 2 です。クラス ABC に含まれる点の中で、D,E と距離が近いのは C ですから、Step3 の総当たりの距離表のクラス ABC と D、E の距離はそれぞれ 2.236 と 2.828 となります。これを繰り返してすべての点が一つのクラスにま

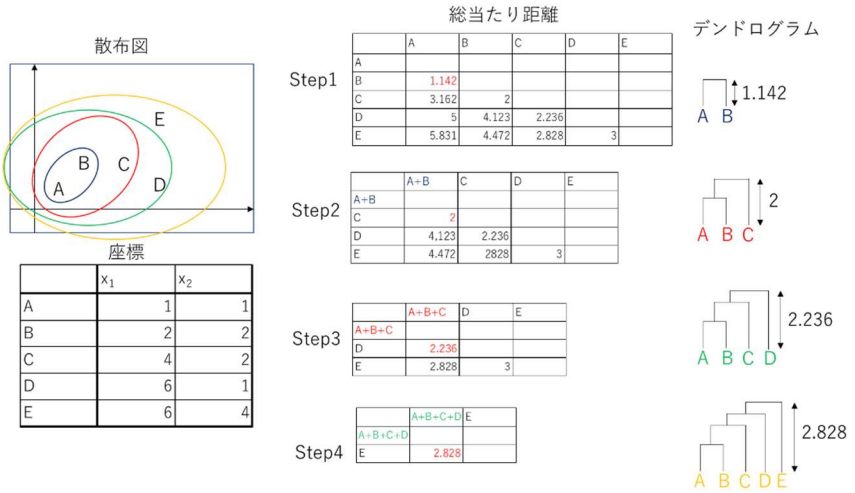


図 128.単連結法の距離と連結法

とまったところで作業終了です。ここであげた計算例では、一つのクラスが順次他の点を吸収して外延的にクラスが大きくなっているだけで、クラスターと認識できる構造を作っていません。もちろん単連結法でも感覚的に納得できるクラスターが形成されることはありますが、単連結法では意味のあるクラスターが形成されないことがしばしばあることも事実です。

VII-3-2-1-3. 完全連結法(complete linkage method)

完全連結法は最長距離法(maximum distance method)とも言います。最長距離法という名前は作業手順からついた名前ですが、結果を見ると完全連結法という名前にも納得がいきます。完全連結法では単連結法とは反対に、クラスター間の距離をそれぞれのクラスター間で最も遠い点間の距離とします。この定義にしたがってあるクラス(C_a)とあるクラス(C_b)を結合すると、新たにクラス($C_{a\cup b}$)と他の$d(C_a, C_k)$との距離$d(C_{a\cup b},\ C_k)$は

$$d(C_{a\cup b}, C_k) = \max\left(d(C_a, C_k), d(C_a, C_k)\right)$$

となります。元のクラスターとC_kとの距離の大きい方が新たなクラス($C_{a\cup b}$)と他のクラス(C_k)との距離になるということです。単連結法と同様に手順を図 129 に書きました。Step1 は単連結法と同じで各点間の距離の総当たり表を作ります。最も距離が近い組み合わせ（A,B）を一つのクラスとしてまとめます。距離(非類似度)は二つの点の距離$\sqrt{(2-1)^2+(2-1)^2} = \sqrt{2} \approx 1.414$です。第二ステップでは、クラス AB と他の点との距離を AB どちらかの点と遠い方の距離とします。この場合、C,D,E のいずれとも遠い点は A ですから、クラス AB と C,D,E の点の距離は、3.162, 5, 5.831 です。こうしてできた step2 の距離の総当たり表の中で最も距離が近いのは点 C と点 D ですから、これをまとめてクラス CD を作ります。C と D の距離(非類似度)は 2 .236 です。クラス CD に含まれる点の中で、E と距離が遠いのは D ですから、Step3 の総当たり表でクラス CD と点 E の距離は点 D と E の距離 3 になります。これを使って step4 の総当たり距離表を作ります。

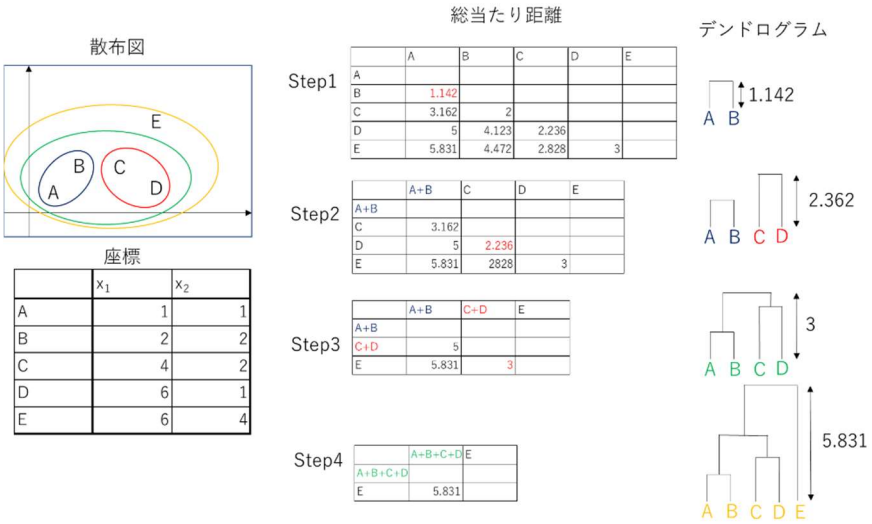


図 129.完全連結法の距離と連結法

これを繰り返してすべての点が一つのクラスにまとまったところで作業終了です。デンドログラムの形は、明らかに単連結法と違います。散布図からは AB が一つの塊、CD 一つの塊で、クラス AB、クラス CD、クラス E の 3 つのクラスがあるような印象を受けます。この場合、感覚的には完全連結法の結果の方が納得できるかもしれませんが、これはどちらが正しいかということではありません。重要なのは、単連結法は一つ一つ外延的につながっていく構造をとりやすいのに対して、完全連結法は、内部にサブクラスを持った枝分かれ図を作りやすく、クラスの数を決めやすいということです。非類似度 2.36 と 3 の間に閾値を取って、3 つのクラスを持つクラスターと躊躇なく判断できるかもしれません。完全連結法の方がまとまりやすいのです。それは、最も遠い点というのが、その内側にクラスター内のすべての点を含んでいるので、最も近い点に比べて代表性が高いからです。

VII-3-2-1-4. 群平均法(group average method)

群平均法は:UPGMA(unweighted pair group with Arithmetic average)とも言います。単連結法も完全連結法もクラスの中の 1 つの点を代表点としてクラス間の距離を定義します。一つの点にクラスを代表させるのではなくて、クラス内のすべての点を使ってクラス間の距離を決めた方が代表性があるという考えは自然です。群平均法はクラス内の全点（要素 element）を使って距離を計算します。図 130 に考え方を図示しました。クラス A にa_1からa_{n_a}までのn_a個の要素(element)があり、クラス B にb_1からb_{n_b}までのn_b個の要素(element)があるとき、要素a_iはクラス B のb_1からb_{n_b}までのn_b個の要素（element）と距離を持っており、その合計は

$$\sum_{j=1}^{n_b} d(a_i, b_j)$$

です。クラス A の要素はa_1からa_{n_a}までn_a個の個ありますから、すべての距離の合計は

$$\sum_{i=1}^{n_a}\sum_{j=1}^{n_b} d(a_i, b_j)$$

です。したがって群平均の距離は

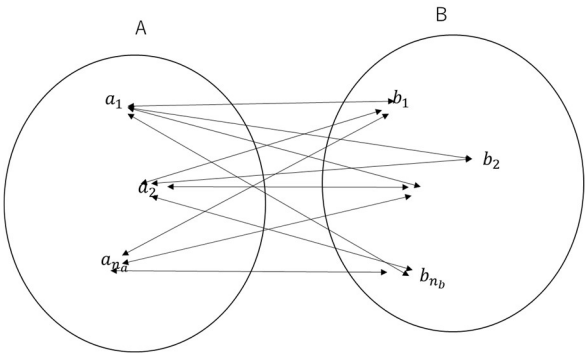


図 130 群平均法の考え方

$$d(A,B)=\frac{1}{n_a n_b}\sum_{i=1}^{n_a}\sum_{j=1}^{n_b}d\left(a_i,b_j\right) \qquad \text{i}$$

になります。クラス A とクラス B をまとめて、クラス$A\cup B$を作った時、他のクラス K とクラス$A\cup B$の距離は

$$d(K,A\cup B)=\frac{1}{n_k n_{a\cup b}}\sum_{l=1}^{n_k}\sum_{m=1}^{n_{a\cup b}}d(k_l,a\cup b_m) \qquad \text{ii}$$

で、クラスAとクラス K、クラス B とクラス K の距離はそれぞれ

$$d(K,A)=\frac{1}{n_k n_a}\sum_{l=1}^{n_k}\sum_{i=1}^{n_a}d(k_l,a_i) \qquad \text{iii}$$

$$d(K,B)=\frac{1}{n_k n_b}\sum_{l=1}^{n_k}\sum_{j=1}^{n_b}d\left(k_l,b_j\right) \qquad \text{iv}$$

ですが、

$$n_{a\cup b}=n_a+n_b$$

で

$$\sum_{m=1}^{n_{a\cup b}}d(k_l,a\cup b_m)=\sum_{i=1}^{n_a}d(k_l,a_i)+\sum_{j=1}^{n_b}d\left(k_l,b_j\right)$$

ですから式 ii は

$$d(K,A\cup B)=\frac{1}{n_k n_{a\cup b}}\sum_{l=1}^{n_k}\left(\sum_{i=1}^{n_a}d(k_l,a_i)+\sum_{j=1}^{n_b}d\left(k_l,b_j\right)\right)$$

$$=\frac{1}{n_k n_{a\cup b}}\sum_{l=1}^{n_k}\sum_{i=1}^{n_a}d(k_l,a_i)+\frac{1}{n_k n_{a\cup b}}\sum_{l=1}^{n_k}\sum_{j=1}^{n_b}d\left(k_l,b_j\right)$$

$$=\frac{1}{n_k n_{a\cup b}}n_k n_a d(K,A)+\frac{1}{n_k n_{a\cup b}}n_k n_b d(K,B)$$

$$=\frac{n_a}{n_{a\cup b}}d(K,A)+\frac{n_b}{n_{a\cup b}}d(K,B)$$

$$=\frac{n_a}{n_a+n_b}d(K,A)+\frac{n_b}{n_a+n_b}d(K,B)$$

この式は、クラスをまとめる前のクラス A、クラス B とクラス K の重み付きの平均になっています。図 131 に連結のステップと距離の計算を図持しました。Step はすべての要素間の距離ですから、今までと同じです。Step1 でクラス$A\cup B$が出来たので、Step2 では、

$$d(C, A \cup B) = \frac{n_a}{n_a + n_b} d(C, A) + \frac{n_b}{n_a + n_b} d(C, B)$$

$$= \frac{1}{2} \times 3.162 + \frac{1}{2} \times 2 = 2.581$$

$$d(D, A \cup B) = \frac{n_a}{n_a + n_b} d(D, A) + \frac{n_b}{n_a + n_b} d(D, B)$$

$$= \frac{1}{2} \times 5 + \frac{1}{2} \times 4.123 = 4.562$$

$$d(E, A \cup B) = \frac{n_a}{n_a + n_b} d(E, A) + \frac{n_b}{n_a + n_b} d(E, B)$$

$$= \frac{1}{2} \times 5.831 + \frac{1}{2} \times 4.472 = 5.152$$

と計算して、距離の総当たり表を作ります。$d(C, D)$=2.236 が最も小さいので、CD を連結しクラス CD とします。Step3 では

$$d(A \cup B, C \cup D) = \frac{n_c}{n_c + n_d} d(A \cup B, C) + \frac{n_d}{n_c + n_d} d(A \cup B, D)$$

$$= \frac{1}{2} \times 2.581 + \frac{1}{2} \times 4.562 = 3.571$$

$$d(E, C \cup D) = \frac{n_c}{n_c + n_d} d(E, C) + \frac{n_c}{n_c + n_d} d(E, D)$$

$$= \frac{1}{2} \times 2.828 + \frac{1}{2} \times 3 = 2.914$$

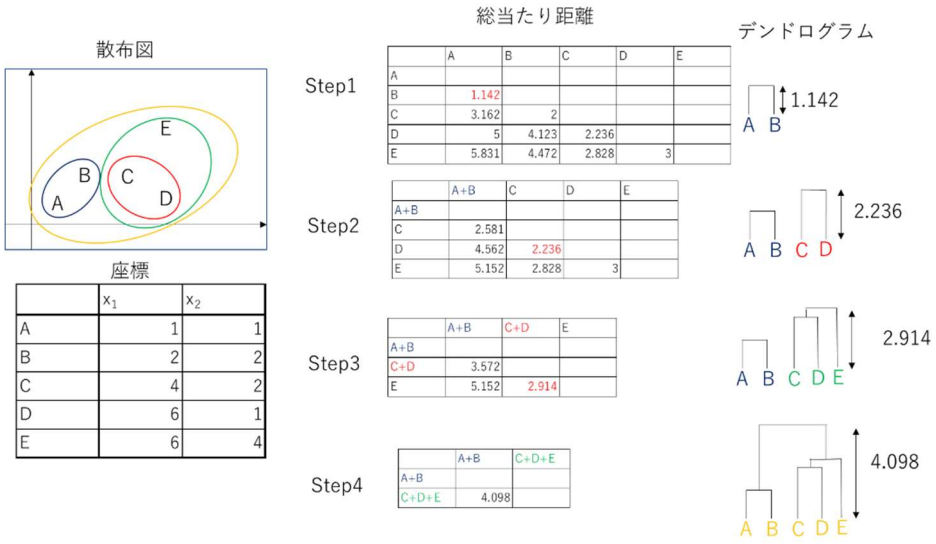


図 131.群連結法の距離と連結

$d(E, C \cup D)$が一番小さいので、Eと$C \cup D$を連結して、クラス CDE とします。Step4 では、$d(A \cup B, C \cup D \cup E)$の距離を計算します。

$$d(A \cup B, C \cup D \cup E) = \frac{n_{c \cup d}}{n_{c \cup d} + n_e} d(A \cup B, C \cup D) + \frac{n_e}{n_{c \cup d} + n_e} d(A \cup B, E)$$

$$= \frac{2}{3}\, d(A \cup B, C \cup D) + \frac{1}{3}\, d(A \cup B, E)$$

$$= \frac{2}{3} \times 3.571 + \frac{1}{3} \times 5.152 = 4.098$$

これが、クラス AB とクラス CDE の距離です。デンドログラムは単連結法とも完全連結法とも違って初めに大きく 2 つのグループに分かれる形になっています。これは全データを考慮してクラスをまとめた結果です。

VII-3-2-1-5. 重心法(centroid method)

重心法は: UPGMC unweighted pair group method using centroids)とも言います。クラスターに含まれる点の重心をそのクラスターの代表点とします。データを空間的な位置で表現しユークリッド距離やマハラノビス距離など距離で非類似度を表せる場合には自然で素朴な考え方だと言えるでしょう。重心法ではクラスの代表値をクラスの要素の平均（重心）とします。そして距離（非類似度）を重心間のユークリッド距離の二乗とします。ユークリッド距離にしない理由は、分散分析的に考える場合には偏差よりもその二乗である分散比を問題にするからです。その方が非類似度という概念に近いのです。これをデータに当てはめて実行してそのプロセスを示したのが図 132 です。Step1 各点の座標から点間のユークリッド距離を求め、その事情を距離（非類似度）とします。点 AB の距離が近いので、これをまとめてクラス AB とし、Step 2 AB の重心を求めこの重心と他の点の距離（非類似度）を計算します。CD が最も距離（非類似度）が小さいのでこれをまとめて、クラス CD とします。これを繰り返してデンドログラムを作ります。原理的な説明はこれで十分なのですが、要素の数が多くなると、重心の座標から毎回総当たりの距離表を作るという作業は大変で

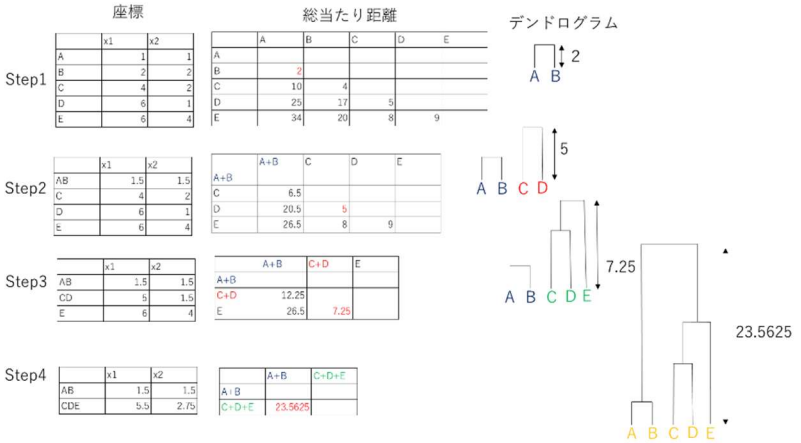


図 132.重心法によるデンドログラムの作成

す。群平均法では、直前のステップの距離データから、次の距離を計算する式をつくりました。

$$d(K, A \cup B) = \frac{n_a}{n_a + n_b} d(K, A) + \frac{n_b}{n_a + n_b} d(K, B)$$

この式は、Lance-Williams の更新式と言われている式で、この式を使うと毎回の距離の計算が楽になります。Lance-Williams の更新式は一般的には次の形をしています。

$$d(K, A \cup B) = \alpha_a d(K, A) + \alpha_b d(K, B) + \beta d(A, B) + \gamma |d(K, A) - d(K, B)|$$

群平均法では、$\alpha_a = \frac{n_a}{n_a+n_b}, \alpha_b = \frac{n_b}{n_a+n_b}$, $\beta = 0, \gamma = 0$となっているということです。ちなみに単連結法は$\alpha_a = \frac{1}{2}, \alpha_b = \frac{1}{2}$, $\beta = 0, \gamma = -\frac{1}{2}1$、完全連結法は $\alpha_a = \frac{1}{2}, \alpha_b = \frac{1}{2}$, $\beta = 0, \gamma = \frac{1}{2}$です。重心法でも Lance Williams の更新式を作ってみます。重心法のクラスター間の距離(非類似度)の定義は以下の式です。

$$d(A, B) = D(g_a, g_b)^2 = (\boldsymbol{g}_a - \boldsymbol{g}_b) \cdot (\boldsymbol{g}_a - \boldsymbol{g}_b) \qquad \text{i}$$

g_a, g_bはクラス A、B の重心
$D(g_a, g_b)$はg_a, g_b間のユークリッド距離
ドット$(\cdot)$*は内積*

$$\boldsymbol{g}_a = (g_{a1} \quad \cdots \quad g_{aP}), \qquad \boldsymbol{g}_b = (g_{b1} \quad \cdots \quad g_{bP})$$

重心とはサンプルサイズ（要素の数）で重みを付けた平均値だから、

$$\boldsymbol{g}_{a\cup b} = \frac{n_a}{n_a + n_b} \boldsymbol{g}_a + \frac{n_b}{n_a + n_b} \boldsymbol{g}_b \qquad \text{ii}$$

変数の数が P 個だとすると、クラス K とクラス$A \cup B$の距離は

$$d(K, A \cup B) = D(g_k, g_{a\cup b})^2 = (\boldsymbol{g}_k - \boldsymbol{g}_{a\cup b}) \cdot (\boldsymbol{g}_k - \boldsymbol{g}_{a\cup b}) \qquad \text{iii}$$

$$= \boldsymbol{g}_k \cdot \boldsymbol{g}_k - 2\boldsymbol{g}_k \boldsymbol{g}_{a\cup b} + \boldsymbol{g}_{a\cup b} \cdot \boldsymbol{g}_{a\cup b}$$

注:ベクトルでも分配法則は成り立ちます

式 iii に式 ii を代入、

$$d(K, A \cup B) = \boldsymbol{g}_k \cdot \boldsymbol{g}_k - 2\boldsymbol{g}_k \left(\frac{n_a}{n_a + n_b} \boldsymbol{g}_a + \frac{n_b}{n_a + n_b} \boldsymbol{g}_{bx_p}\right) + \left(\frac{n_a}{n_a + n_b} \boldsymbol{g}_a + \frac{n_b}{n_a + n_b} \boldsymbol{g}_{bx_p}\right) \cdot \left(\frac{n_a}{n_a + n_b} \boldsymbol{g}_a + \frac{n_b}{n_a + n_b} \boldsymbol{g}_{bx_p}\right)$$

$$= \boldsymbol{g}_k \cdot \boldsymbol{g}_k - \frac{2n_a}{n_a + n_b} \boldsymbol{g}_k \cdot \boldsymbol{g}_a - \frac{2n_b}{n_a + n_b} \boldsymbol{g}_k \cdot \boldsymbol{g}_b + \frac{{n_a}^2}{(n_a + n_b)^2} \boldsymbol{g}_a \cdot \boldsymbol{g}_a + 2\frac{n_a n_b}{(n_a + n_b)^2} \boldsymbol{g}_a \cdot \boldsymbol{g}_b + \frac{{n_b}^2}{(n_a + n_b)^2} \boldsymbol{g}_b \cdot \boldsymbol{g}_b \qquad \text{iii}$$

$$d(K, A) = D(g_k, g_a)^2 = (\boldsymbol{g}_k - \boldsymbol{g}_a) \cdot (\boldsymbol{g}_k - \boldsymbol{g}_a) = \boldsymbol{g}_k \cdot \boldsymbol{g}_k - 2\boldsymbol{g}_k \cdot \boldsymbol{g}_a + \boldsymbol{g}_a \cdot \boldsymbol{g}_a$$

$$2\boldsymbol{g}_k \cdot \boldsymbol{g}_a = \boldsymbol{g}_k \cdot \boldsymbol{g}_k + \boldsymbol{g}_a \cdot \boldsymbol{g}_a - d(K, A) \qquad \text{iv}$$

$$d(K, B) = D(g_k, g_b)^2 = (\boldsymbol{g}_k - \boldsymbol{g}_b) \cdot (\boldsymbol{g}_k - \boldsymbol{g}_b) = \boldsymbol{g}_k \cdot \boldsymbol{g}_k - 2\boldsymbol{g}_k \cdot \boldsymbol{g}_b + \boldsymbol{g}_b \cdot \boldsymbol{g}_b$$

$$2\boldsymbol{g}_k \cdot \boldsymbol{g}_b = \boldsymbol{g}_k \cdot \boldsymbol{g}_k + \boldsymbol{g}_b \cdot \boldsymbol{g}_b - d(K, B) \qquad \text{v}$$

$$d(A, B) = D(g_a, g_b)^2 = (\boldsymbol{g}_a - \boldsymbol{g}_b) \cdot (\boldsymbol{g}_a - \boldsymbol{g}_b) = \boldsymbol{g}_a \cdot \boldsymbol{g}_a - 2\boldsymbol{g}_a \cdot \boldsymbol{g}_b + \boldsymbol{g}_b \cdot \boldsymbol{g}_b$$

$$2\boldsymbol{g}_a \cdot \boldsymbol{g}_b = \boldsymbol{g}_a \cdot \boldsymbol{g}_a + \boldsymbol{g}_b \cdot \boldsymbol{g}_b - d(A, B) \qquad \text{vi}$$

式 iii に iv,v,vi を代入

$$d(K, A \cup B) = \boldsymbol{g}_k \cdot \boldsymbol{g}_k - \frac{n_a}{n_a + n_b}\left(\boldsymbol{g}_k \cdot \boldsymbol{g}_k + \boldsymbol{g}_a \cdot \boldsymbol{g}_a - d(K, A)\right) - \frac{n_b}{n_a + n_b}\left(\boldsymbol{g}_k \cdot \boldsymbol{g}_k + \boldsymbol{g}_b \cdot \boldsymbol{g}_b - d(K, B)\right) + \frac{{n_a}^2}{(n_a + n_b)^2}\boldsymbol{g}_a \cdot \boldsymbol{g}_a$$

$$+ \frac{n_a n_b}{(n_a + n_b)^2}\left(\boldsymbol{g}_a \cdot \boldsymbol{g}_a + \boldsymbol{g}_b \cdot \boldsymbol{g}_b - d(A, B)\right) + \frac{{n_b}^2}{(n_a + n_b)^2}\boldsymbol{g}_b \cdot \boldsymbol{g}_b$$

$$= \left(1 - \frac{n_a}{n_a + n_b} - \frac{n_b}{n_a + n_b}\right)\boldsymbol{g}_k \cdot \boldsymbol{g}_k + \left(\frac{{n_a}^2}{(n_a + n_b)^2} + \frac{n_a n_b}{(n_a + n_b)^2} - \frac{n_a}{n_a + n_b}\right)\boldsymbol{g}_a \cdot \boldsymbol{g}_a + \left(\frac{{n_b}^2}{(n_a + n_b)^2} + \frac{n_a n_b}{(n_a + n_b)^2} - \frac{n_b}{n_a + n_b}\right)$$

$$+ \frac{n_a}{n_a + n_b}\, d(K, A) + \frac{n_b}{n_a + n_b} d(A, B) - \frac{n_a n_b}{(n_a + n_b)^2}$$

$$1 - \frac{n_a}{n_a + n_b} - \frac{n_b}{n_a + n_b} = \frac{{n_a}^2}{(n_a + n_b)^2} + \frac{n_a n_b}{(n_a + n_b)^2} - \frac{n_a}{n_a + n_b} = \frac{{n_b}^2}{(n_a + n_b)^2} + \frac{n_a n_b}{(n_a + n_b)^2} - \frac{n_b}{n_a + n_b} = 0$$

だから

$$d(K, A \cup B) = \frac{n_a}{n_a + n_b} d(K, A) + \frac{n_b}{n_a + n_b} d(K, B) - \frac{n_a n_b}{(n_a + n_b)^2} d(A, B)$$

となります。Lance Williams の更新式は$\alpha_a = \frac{n_a}{n_a+n_b}, \alpha_b = \frac{n_b}{n_a+n_b}$, $\beta = \frac{n_a n_b}{(n_a+n_b)^2}, \gamma = 0$です。ここにあげた例では、デンドログラムの形は群平均法と重心法であまり違いがありません。

VII-3-2-1-6. ウォード法(Ward's method)

ウォード法は最小分散法(minimum variance method)とも呼ばれます。クラス内でのデータのバラツキの増加を最小にする方向でクラスを連結していくからです。次の式がウォード法の距離(非類似度）の定義です。

$$d(A, B) = L(A \cup B) - L(A) - L(B) \quad \text{i}$$

$d(A, B)$: *クラスAとクラスBの距離*

$L(A)$: *重心からの距離の二乗和*

$$L(A) = \sum_{i=1}^{n_a} D(\boldsymbol{x}_{ai}, \boldsymbol{g}_a)^2 = \sum_{i=1}^{n_a} (\boldsymbol{x}_{ai} - \boldsymbol{g}_a) \cdot (\boldsymbol{x}_{ai} - \boldsymbol{g}_a)$$

$$= \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} \cdot \boldsymbol{x}_{ai} - 2\sum_{i=1}^{n_a} \boldsymbol{g}_a \cdot \boldsymbol{x}_{ai} + n_a \boldsymbol{g}_a \cdot \boldsymbol{g}_a$$

$$= \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} \cdot \boldsymbol{x}_{ai} - 2\boldsymbol{g}_a \cdot \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} + n_a \boldsymbol{g}_a \cdot \boldsymbol{g}_a \quad \text{ii}$$

$$\boldsymbol{x}_{ai} = (x_{ai1} \quad \cdots \quad x_{aip}), \qquad \boldsymbol{g}_a = (g_{a1} \quad \cdots \quad g_{ap})$$

$D(A, B)$: *点AとBのユークリッド距離*

g_a *はクラスAの重心*

注: $\boldsymbol{g}_a \cdot \sum_{i=1}^{n_a} \boldsymbol{x}_{ai}$という表現が数式の記述として許されるかどうかは知りませんが、$\boldsymbol{g}_a$はiにかかわらず一定だからシグマの外に出せます。

同様に

$$L(B) = \sum_{j=1}^{n_b} D(\boldsymbol{x}_{bi}, \boldsymbol{g}_b)^2 = \sum_{j=1}^{n_b} \boldsymbol{x}_{bj} \cdot \boldsymbol{x}_{bj} - 2\boldsymbol{g}_b \cdot \sum_{j=1}^{n_b} \boldsymbol{x}_{ai} + n_b \boldsymbol{g}_b \cdot \boldsymbol{g}_b \quad \text{iii}$$

$$L(A \cup B) = \sum_{k=1}^{n_a+n_b} D(\boldsymbol{x}_{a\cup bk}, \boldsymbol{g}_{a\cup b})^{\,2} = \sum_{k=1}^{n_a+n_b} (\boldsymbol{x}_{a\cup bk} - \boldsymbol{g}_{a\cup b}) \cdot (\boldsymbol{x}_{a\cup bk} - \boldsymbol{g}_{aub})$$

これは$\boldsymbol{x}_{a\cup bk}$の総和は$\boldsymbol{x}_{ai}$の総和と$\boldsymbol{x}_{bj}$の総和の和だから

$$= \sum_{i=1}^{n_a} (\boldsymbol{x}_{ai} - \boldsymbol{g}_{a\cup b}) \cdot (\boldsymbol{x}_{ai} - \boldsymbol{g}_{aub}) + \sum_{j=1}^{n_b} (\boldsymbol{x}_{bj} - \boldsymbol{g}_{a\cup b}) \cdot (\boldsymbol{x}_{bj} - \boldsymbol{g}_{a\cup b})$$

$$= \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} \cdot \boldsymbol{x}_{ai} - 2\boldsymbol{g}_{aub} \cdot \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} + n_a\, \boldsymbol{g}_{aub} \cdot \boldsymbol{g}_{aub} + \sum_{j=1}^{n_b} \boldsymbol{x}_{bj} \cdot \boldsymbol{x}_{bj} - 2\boldsymbol{g}_{aub} \cdot \sum_{j=1}^{n_b} \boldsymbol{x}_{ai} + n_b\, \boldsymbol{g}_{aub} \cdot \boldsymbol{g}_{aub} \quad \text{iv}$$

i 式にもどって、ii, iii,iv を代入

$$d(A,B) = L(A \cup B) - L(A) - L(B)$$

$$= \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} \cdot \boldsymbol{x}_{ai} - 2\boldsymbol{g}_{aub} \cdot \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} + n_a\, \boldsymbol{g}_{aub} \cdot \boldsymbol{g}_{aub} + \sum_{j=1}^{n_b} \boldsymbol{x}_{bj} \cdot \boldsymbol{x}_{bj} - 2\boldsymbol{g}_{aub} \cdot \sum_{j=1}^{n_b} \boldsymbol{x}_{bj} + n_b\, \boldsymbol{g}_{aub} \cdot \boldsymbol{g}_{aub}$$

$$- \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} \cdot \boldsymbol{x}_{ai} + 2\boldsymbol{g}_a \cdot \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} - n_a \boldsymbol{g}_a \cdot \boldsymbol{g}_a - \sum_{j=1}^{n_b} \boldsymbol{x}_{bj} \cdot \boldsymbol{x}_{bj} + 2\boldsymbol{g}_b \cdot \sum_{j=1}^{n_b} \boldsymbol{x}_{bj} - n_b \boldsymbol{g}_b \cdot \boldsymbol{g}_b$$

$$= 2\left(\boldsymbol{g}_a \cdot \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} - \boldsymbol{g}_{aub} \cdot \sum_{i=1}^{n_a} \boldsymbol{x}_{ai}\right) + 2\left(\boldsymbol{g}_b \cdot \sum_{j=1}^{n_b} \boldsymbol{x}_{bj} - \boldsymbol{g}_{aub} \cdot \sum_{j=1}^{n_b} \boldsymbol{x}_{bj}\right) + (n_a + n_b)\boldsymbol{g}_{aub} \cdot \boldsymbol{g}_{aub} - n_a \boldsymbol{g}_a \cdot \boldsymbol{g}_a - n_b \boldsymbol{g}_b \cdot \boldsymbol{g}_b$$

$$= 2\left((\boldsymbol{g}_a - \boldsymbol{g}_{aub}) \cdot \sum_{i=1}^{n_a} \boldsymbol{x}_{ai}\right) + 2\left((\boldsymbol{g}_b - \boldsymbol{g}_{aub}) \cdot \sum_{j=1}^{n_b} \boldsymbol{x}_{bj}\right) + (n_a + n_b)\boldsymbol{g}_{aub} \cdot \boldsymbol{g}_{aub} - n_a \boldsymbol{g}_a \cdot \boldsymbol{g}_a - n_b \boldsymbol{g}_b \cdot \boldsymbol{g}_b \quad \text{v}$$

一方、重心とはサンプルサイズ（要素の数）で重みを付けた平均値だから、

$$\boldsymbol{g}_{a\cup b} = \frac{n_a}{n_a + n_b} \boldsymbol{g}_a + \frac{n_b}{n_a + n_b} \boldsymbol{g}_b$$

$$\boldsymbol{g}_a - \boldsymbol{g}_{aub} = \boldsymbol{g}_a - \frac{n_a}{n_a + n_b} \boldsymbol{g}_a - \frac{n_b}{n_a + n_b} \boldsymbol{g}_b = \frac{n_b}{n_a + n_b} (\boldsymbol{g}_a - \boldsymbol{g}_b)$$

$$\boldsymbol{g}_b - \boldsymbol{g}_{aub} = \boldsymbol{g}_b - \frac{n_b}{n_a + n_b} \boldsymbol{g}_b - \frac{n_a}{n_a + n_b} \boldsymbol{g}_a = \frac{n_a}{n_a + n_b} (\boldsymbol{g}_a - \boldsymbol{g}_b)$$

$$\boldsymbol{g}_{a\cup b} \cdot \boldsymbol{g}_{a\cup b} = \frac{{n_a}^2}{(n_a + n_b)^2} \boldsymbol{g}_a \cdot \boldsymbol{g}_a + \frac{2n_a n_b}{(n_a + n_b)^2} \boldsymbol{g}_a \cdot \boldsymbol{g}_b + \frac{{n_b}^2}{(n_a + n_b)^2} \boldsymbol{g}_b \cdot \boldsymbol{g}_b$$

また、

$$D(\boldsymbol{g}_a, \boldsymbol{g}_b)^{\,2} = (\boldsymbol{g}_a - \boldsymbol{g}_b) \cdot (\boldsymbol{g}_a - \boldsymbol{g}_b)$$

これらを v 式に代入すると

$$d(A,B) = 2\left((g_a - g_{aub}) \cdot \sum_{i=1}^{n_a} x_{ai}\right) + 2\left((\boldsymbol{g}_b - \boldsymbol{g}_{aub}) \cdot \sum_{j=1}^{n_b} \boldsymbol{x}_{bj}\right) + (n_a + n_b)\boldsymbol{g}_{aub} \cdot \boldsymbol{g}_{aub} - n_a \boldsymbol{g}_a \cdot \boldsymbol{g}_a - n_b \boldsymbol{g}_b \cdot \boldsymbol{g}_b$$

$$= 2\left(\frac{n_b}{n_a + n_b}(\boldsymbol{g}_a - \boldsymbol{g}_b) \cdot \sum_{i=1}^{n_a} \boldsymbol{x}_{ai} - \frac{n_a}{n_a + n_b}(\boldsymbol{g}_a - \boldsymbol{g}_b) \cdot \sum_{j=1}^{n_b} \boldsymbol{x}_{bj}\right) + \frac{{n_a}^2}{n_a + n_b} \boldsymbol{g}_a \cdot \boldsymbol{g}_a + \frac{2n_a n_b}{n_a + n_b} \boldsymbol{g}_a \cdot \boldsymbol{g}_b + \frac{{n_b}^2}{n_a + n_b} \boldsymbol{g}_b \cdot \boldsymbol{g}_b$$

$$- n_a \boldsymbol{g}_a \cdot \boldsymbol{g}_a - n_b \boldsymbol{g}_b \cdot \boldsymbol{g}_b$$

$$= 2\left(\frac{n_b}{n_a+n_b}(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot n_a\sum_{i=1}^{n_a}\frac{\boldsymbol{x}_{ai}}{n_a}-\frac{n_a}{n_a+n_b}(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot n_b\sum_{j=1}^{n_b}\frac{\boldsymbol{x}_{bj}}{n_b}\right)+\frac{{n_a}^2-n_a(n_a+n_b)}{n_a+n_b}\boldsymbol{g}_a\cdot\boldsymbol{g}_a+\frac{2n_an_b}{n_a+n_b}\boldsymbol{g}_a\cdot\boldsymbol{g}_b$$

$$+\frac{{n_b}^2-n_b(n_a+n_b)}{n_a+n_b}\boldsymbol{g}_b\cdot\boldsymbol{g}_b$$

$$= 2\left(\frac{n_bn_a}{n_a+n_b}(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot\sum_{i=1}^{n_a}\frac{\boldsymbol{x}_{ai}}{n_a}-\frac{n_an_b}{n_a+n_b}(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot\sum_{j=1}^{n_b}\frac{\boldsymbol{x}_{bj}}{n_b}\right)-\frac{n_an_b}{n_a+n_b}\boldsymbol{g}_a\cdot\boldsymbol{g}_a+\frac{2n_an_b}{n_a+n_b}\boldsymbol{g}_a\cdot\boldsymbol{g}_b-\frac{n_bn_a}{n_a+n_b}\boldsymbol{g}_b\cdot\boldsymbol{g}_b$$

$$= 2\left(\frac{n_bn_a}{n_a+n_b}(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot\left(\sum_{i=1}^{n_a}\frac{\boldsymbol{x}_{ai}}{n_a}-\sum_{j=1}^{n_b}\frac{\boldsymbol{x}_{bj}}{n_b}\right)\right)-\left(\frac{n_an_b}{n_a+n_b}\boldsymbol{g}_a\cdot\boldsymbol{g}_a-\frac{2n_an_b}{n_a+n_b}\boldsymbol{g}_a\cdot\boldsymbol{g}_b+\frac{n_bn_a}{n_a+n_b}\boldsymbol{g}_b\cdot\boldsymbol{g}_b\right)$$

$\sum_{i=1}^{n_a}\frac{\boldsymbol{x}_{ai}}{n_a}-\sum_{j=1}^{n_b}\frac{\boldsymbol{x}_{bj}}{n_b}=\boldsymbol{g}_a-\boldsymbol{g}_b$だから

$$d(A,B)=\frac{n_bn_a}{n_a+n_b}\left(2(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot(\boldsymbol{g}_a-\boldsymbol{g}_b)-(\boldsymbol{g}_a\cdot\boldsymbol{g}_a-2\boldsymbol{g}_a\cdot\boldsymbol{g}_b+\boldsymbol{g}_b\cdot\boldsymbol{g}_b)\right)$$

$\boldsymbol{g}_a\cdot\boldsymbol{g}_a-2\boldsymbol{g}_a\cdot\boldsymbol{g}_b+\boldsymbol{g}_b\cdot\boldsymbol{g}_b=(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot(\boldsymbol{g}_a-\boldsymbol{g}_b)=D(g_a,g_b)^2$だから、

$$d(A,B)=\frac{n_bn_a}{n_a+n_b}\left(2(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot(\boldsymbol{g}_a-\boldsymbol{g}_b)-(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot(\boldsymbol{g}_a-\boldsymbol{g}_b)\right)$$

$$d(A,B)=\frac{n_bn_a}{n_a+n_b}(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot(\boldsymbol{g}_a-\boldsymbol{g}_b)=\frac{n_an_b}{n_a+n_b}D(g_a,g_b)^2$$

つまり、ウォ―ド法における距離には

$$d(A,B)=L(A\cup B)-L(A)-L(B)\qquad 1$$

$$d(A,B)=\frac{n_an_b}{n_a+n_b}D(g_a,g_b)^2\qquad 2$$

という２つの表現の仕方があるということです。１の式の意味は、２つのクラスを統合したときに、重心からの距離の平方和と、統合する前の２つのクラス内での重心からの距離の平方和の差が距離（非類似度）だという意味です。自由度で割っていないので分散の差とは言えませんが、距離が短い（類似度が高い）ものを結合して新しいクラスターを作るときに分散が大きくならないように作っていくということです。２つ目の式の意味は以下のように変形するとわかりやすいと思います。

$$d(A,B)=\frac{n_an_b}{n_a+n_b}D(g_a,g_b)^2=\frac{D(g_a,g_b)^2}{\frac{1}{n_a}+\frac{1}{n_b}}$$

分母の形が印象的なこの式に見覚えがないでしょうか。下の式は「対になってないグループ間のｔ検定でｔ値を求める式に似ています。

$$\mathrm{t}=\frac{M_A-M_B}{\sigma_{A-B}\sqrt{\frac{1}{n_A}+\frac{1}{n_B}}}$$

分母のルートの中の式が同じなのですが、それだけではありません。この式を２乗します。

$$t^2 = \frac{(M_A - M_B)^2}{{\sigma_{A-B}}^2\left(\frac{1}{n_A}+\frac{1}{n_B}\right)}$$

$$t^2{\sigma_{A-B}}^2 = \frac{(M_A - M_B)^2}{\left(\frac{1}{n_A}+\frac{1}{n_B}\right)}$$

M_A, M_Bはそれぞれのグループの平均値ですが、重心は座標の平均値なのだから、２つの式の意味は同じです、t 検定は標本集団が小さい時を想定した検定で、母集団の平均値周りの標本集団の平均値の積率を考えます。その 1 次の積率が標準誤差です。標本が大きくなるとこの範囲が小さくなります。サンプルの２次の積率はサンプルサイズに比例します。これをサンプルサイズ割ったものが母集団の分散（２次の積率）で、その平方根が標準誤差です。

$$\mathrm{E}((M-\mu)^2) = \frac{\sigma^2}{n}$$

$$\mathrm{S.E}=\sqrt{\mathrm{E}((M-\mu)^2)} = \frac{\sigma}{\sqrt{n}}$$

σ^2は母集団の分散で、サンプル集団の平均値からの距離の平方和を自由度$n-1$で割ったものと習いました。「対になっていない t 検定」では、等分散の仮定に基づいて、２つのグループの分散からサンプルサイズで重みをつけて平均して２つのグループを統合してコミにした分散を求めます。この時、統合して出来たグループのサンプルサイズがいくつなのかが問題です。その結論が、

$$n_{a-b} = \frac{n_a n_b}{n_a + n_b}$$

なのです。T値は平均値の差の標準誤差に対する比です。本来、次の形をしています。

$$t = \frac{M_A - M_B}{\frac{\sigma_{A-B}}{\sqrt{n_{a-b}}}}$$

この形を、

$$\frac{1}{n_{a-b}} = \frac{n_a + n_b}{n_a n_b} = \frac{1}{n_a} + \frac{1}{n_b}$$

として変形したのが,

$$t = \frac{M_A - M_B}{\sigma_{A-B}\sqrt{\frac{1}{n_A}+\frac{1}{n_B}}}$$

という式です。$\frac{n_a n_b}{n_a+n_b}$はサンプルサイズなのです。$\frac{n_a n_b}{n_a+n_b}$は、$n_a = n_b$のとき最大値$\frac{n_a n_b}{n_a+n_b} = \frac{n_a}{2} =$

$\frac{n_b}{2}$となり、n_aとn_bの数の違いに応じて小さな値になります。$d(A,B)$の式からｔを求める式を作ってみます。

$$t^2 = \frac{D(g_a, g_b)^2}{{\sigma_{a\cup b}}^2\left(\frac{1}{n_a}+\frac{1}{n_b}\right)}$$

$$t^2{\sigma_{a\cup b}}^2 = \frac{D(g_a, g_b)^2}{\frac{1}{n_a}+\frac{1}{n_b}} = \frac{n_a n_b}{n_a+n_b}D(g_a, g_b)^2$$

$t=1$にして考えると分散${\sigma_{a\cup b}}^2$は重心の距離が離れると大きくなりますが、クラスの要素の数が多いと大きくなり、元のクラスの要素数に違いがあると小さくなります。ウォード法の距離は、重心間の距離だけでなく、要素の少ないクラス同士を先に連結させ、大きなクラスに小さなクラスを統合させ、同じような要素数のクラスを遅く連結させる連結の仕方だと言えます。

Lance-Williams の更新式を作ります。

$$d(A\cup B) = \frac{n_a n_b}{n_a+n_b}D(g_a, g_b)^2 = \frac{n_a n_b}{n_a+n_b}(\boldsymbol{g}_a-\boldsymbol{g}_b)\cdot(\boldsymbol{g}_a-\boldsymbol{g}_b)$$

$$\frac{n_a+n_b}{n_a n_b}d(A\cup B) = \boldsymbol{g}_a\cdot\boldsymbol{g}_a - 2\boldsymbol{g}_a\cdot\boldsymbol{g}_b + \boldsymbol{g}_b\cdot\boldsymbol{g}_b$$

$$2\boldsymbol{g}_a\cdot\boldsymbol{g}_b = \boldsymbol{g}_a\cdot\boldsymbol{g}_a + \boldsymbol{g}_b\cdot\boldsymbol{g}_b - \frac{n_a+n_b}{n_a n_b}d(A\cup B) \quad \text{i}$$

$$d(K\cup A) = \frac{n_a n_k}{n_a+n_k}D(g_a, g_k)^2 = \frac{n_a n_k}{n_a+n_k}(\boldsymbol{g}_a-\boldsymbol{g}_k)\cdot(\boldsymbol{g}_a-\boldsymbol{g}_k)$$

$$2\boldsymbol{g}_a\cdot\boldsymbol{g}_k = \boldsymbol{g}_a\cdot\boldsymbol{g}_a + \boldsymbol{g}_k\cdot\boldsymbol{g}_k - \frac{n_a+n_k}{n_a n_k}d(A\cup K) \quad \text{ii}$$

$$d(K\cup B) = \frac{n_b n_k}{n_b+n_k}D(g_b, g_k)^2 = \frac{n_b n_k}{n_b+n_k}(\boldsymbol{g}_b-\boldsymbol{g}_k)\cdot(\boldsymbol{g}_b-\boldsymbol{g}_k)$$

$$2\boldsymbol{g}_b\cdot\boldsymbol{g}_k = \boldsymbol{g}_b\cdot\boldsymbol{g}_b + \boldsymbol{g}_k\cdot\boldsymbol{g}_k - \frac{n_b+n_k}{n_b n_k}d(B\cup K) \quad \text{iii}$$

$$d(K\cup A\cup B) = \frac{n_k(n_a+n_b)}{n_k+n_a+n_b}D(g_{a\cup b}, g_k)^2 = \frac{n_k(n_a+n_b)}{n_k+n_a+n_b}(\boldsymbol{g}_{a\cup b}-\boldsymbol{g}_k)\cdot(\boldsymbol{g}_{a\cup b}-\boldsymbol{g}_k)$$

$$= \frac{n_k(n_a+n_b)}{n_k+n_a+n_b}(\boldsymbol{g}_{a\cup b}\cdot\boldsymbol{g}_{a\cup b} - 2\boldsymbol{g}_{a\cup b}\cdot\boldsymbol{g}_k + \boldsymbol{g}_k\cdot\boldsymbol{g}_k) \quad \text{iv}$$

$$\boldsymbol{g}_{a\cup b} = \frac{n_a}{n_a+n_b}\boldsymbol{g}_a + \frac{n_b}{n_a+n_b}\boldsymbol{g}_b \quad \text{v}$$

$$\boldsymbol{g}_{a\cup b}\cdot\boldsymbol{g}_{a\cup b}=\frac{{n_a}^2}{(n_a+n_b)^2}\boldsymbol{g}_a\cdot\boldsymbol{g}_a+\frac{2n_an_b}{(n_a+n_b)^2}\boldsymbol{g}_a\cdot\boldsymbol{g}_b+\frac{{n_b}^2}{(n_a+n_b)^2}\boldsymbol{g}_b\cdot\boldsymbol{g}_b \quad \text{vi}$$

式 iv に式 v,vi を代入、

$$d(K\cup A\cup B)=$$

$$=\frac{{n_a}^2n_k}{(n_a+n_b)(n_k+n_a+n_b)}\boldsymbol{g}_a\cdot\boldsymbol{g}_a+\frac{2n_an_bn_k}{(n_a+n_b)(n_k+n_a+n_b)}\boldsymbol{g}_a\cdot\boldsymbol{g}_b+\frac{{n_b}^2n_k}{(n_a+n_b)(n_k+n_a+n_b)}\boldsymbol{g}_b\cdot\boldsymbol{g}_b$$

$$-\frac{2n_an_k}{n_k+n_a+n_b}\boldsymbol{g}_a\boldsymbol{g}_k-\boldsymbol{g}_b\cdot\boldsymbol{g}_k+\frac{n_k(n_a+n_b)}{n_k+n_a+n_b}\boldsymbol{g}_k\cdot\boldsymbol{g}_k \quad \text{vii}$$

式 vii に式 i, ii, iii を代入

$$d(K\cup A\cup B)=\frac{{n_a}^2n_k}{(n_a+n_b)(n_k+n_a+n_b)}\boldsymbol{g}_a\cdot\boldsymbol{g}_a+\frac{n_an_bn_k}{(n_a+n_b)(n_k+n_a+n_b)}\left(\boldsymbol{g}_a\cdot\boldsymbol{g}_a+\boldsymbol{g}_b\cdot\boldsymbol{g}_b-\frac{n_a+n_b}{n_an_b}d(A\cup B)\right)$$

$$+\frac{{n_b}^2n_k}{(n_a+n_b)(n_k+n_a+n_b)}\boldsymbol{g}_b\cdot\boldsymbol{g}_b-\frac{n_an_k}{n_k+n_a+n_b}\left(\boldsymbol{g}_a\cdot\boldsymbol{g}_a+\boldsymbol{g}_k\cdot\boldsymbol{g}_k-\frac{n_a+n_k}{n_an_k}d(A\cup K)\right)$$

$$-\frac{n_bn_k}{n_k+n_a+n_b}\left(\boldsymbol{g}_b\cdot\boldsymbol{g}_b+\boldsymbol{g}_k\cdot\boldsymbol{g}_k-\frac{n_b+n_k}{n_bn_k}d(B\cup K)\right)+\frac{n_k(n_a+n_b)}{n_k+n_a+n_b}\boldsymbol{g}_k\cdot\boldsymbol{g}_k$$

$$=\left(\frac{{n_a}^2n_k}{(n_a+n_b)(n_k+n_a+n_b)}+\frac{n_an_bn_k}{(n_a+n_b)(n_k+n_a+n_b)}-\frac{n_an_k}{n_k+n_a+n_b}\right)\boldsymbol{g}_a\cdot\boldsymbol{g}_a$$

$$+\left(\frac{n_an_bn_k}{(n_a+n_b)(n_k+n_a+n_b)}+\frac{{n_b}^2n_k}{(n_a+n_b)(n_k+n_a+n_b)}-\frac{n_bn_k}{n_k+n_a+n_b}\right)\boldsymbol{g}_b\cdot\boldsymbol{g}_b$$

$$+\left(\frac{n_k(n_a+n_b)}{n_k+n_a+n_b}-\frac{n_an_k}{n_k+n_a+n_b}-\frac{n_bn_k}{n_k+n_a+n_b}\right)\boldsymbol{g}_k\cdot\boldsymbol{g}_k-\frac{n_an_bn_k}{(n_a+n_b)(n_k+n_a+n_b)}\frac{n_a+n_b}{n_an_b}d(A\cup B)$$

$$+\frac{n_an_k}{n_k+n_a+n_b}\frac{n_a+n_k}{n_an_k}d(A\cup K)+\frac{n_bn_k}{n_k+n_a+n_b}\frac{n_b+n_k}{n_bn_k}d(B\cup K)$$

$$\frac{{n_a}^2n_k}{(n_a+n_b)(n_k+n_a+n_b)}+\frac{n_an_bn_k}{(n_a+n_b)(n_k+n_a+n_b)}-\frac{n_an_k}{n_k+n_a+n_b}$$

$$=\frac{n_an_bn_k}{(n_a+n_b)(n_k+n_a+n_b)}+\frac{{n_b}^2n_k}{(n_a+n_b)(n_k+n_a+n_b)}-\frac{n_bn_k}{n_k+n_a+n_b}$$

$$=\frac{n_k(n_a+n_b)}{n_k+n_a+n_b}-\frac{n_an_k}{n_k+n_a+n_b}-\frac{n_bn_k}{n_k+n_a+n_b}=0$$

だから、Lance Williams の更新式は

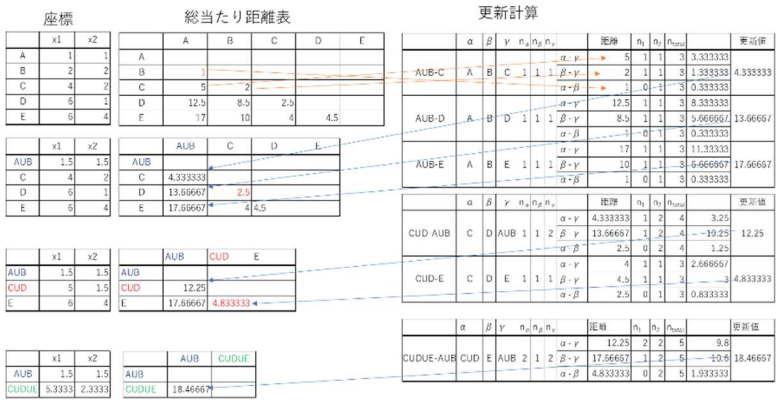
$$d(K\cup A\cup B)=+\frac{n_a+n_k}{n_a+n_b+n_k}d(A\cup K)+\frac{n_b+n_k}{n_a+n_b+n_k}d(B\cup K)-\frac{n_k}{n_a+n_b+n_k}d(A\cup B)$$

$$\alpha_a=\frac{n_a+n_k}{n_a+n_b+n_k},\alpha_b=\frac{n_b+n_k}{n_a+n_b+n_k},\qquad\beta=-\frac{n_k}{n_a+n_b+n_k},\gamma=0$$

となります。

表 68 は、この計算式を使った距離の更新の計算記録です。ウォード法には距離を定義する

表 68. ウォ―ド法による距離の更新



式が 2 つあるので、それらの式を使ってもクラスを統合したときの距離の更新の計算はできます。クラスター分析で距離(非類似度)として用いられるものは様々で、空間的な座標間の距離の概念とは異なるものがあります（例えばコサイン類似度)。そのような場合、ウォ―ド法で距離(非類似度)を更新するには更新式を使うことになります。各要素の座標は左上の座標の表のとおりです。それを使って Step1 では定義にしたがって総当たり距離表をつくります。距離が最も短いのは A-B 間ですから A と B を統合してクラス$A \cup B$を作ります。統合されたクラス$A \cup B$と C、$A \cup B$と D、$A \cup B$と E の距離の計算を右側の更新計算の表でしています。右上の表の最初の 3 行が$A \cup B$と C$(A \cup B - C)$の距離の計算です。 1 行目が、更新式の$\frac{n_a+n_k}{n_a+n_b+n_k} d(A \cup K)$の部分にあたり、$n_a = n_1$、$n_k = n_2$、$n_a + n_b + n_k = n_{total}$として、$\frac{1+1}{1+1+1} \times 5 = 3.3333$と計算しています。 2 行目が$\frac{n_b+n_k}{n_a+n_b+n_k} d(B \cup K)$の部分で$n_b = n_1$として、$\frac{1+1}{1+1+1} \times 2 = 1.3333$、3 行目が$\frac{n_k}{n_a+n_b+n_k} d(A \cup B)$の部分で、$\frac{1}{1+1+1} \times 1 = 0.3333$と計算し、最後に$3.3333 + 1.3333 - 0.33333 = 4.3333$となって、クラス$A \cup B$と C$(A \cup B - C)$の距離を得ます。クラス$A \cup B$と D$(A \cup B - D)$、クラス$A \cup B$と E$(A \cup B - E)$1 についても同じ計算をして、総当たり距離表を更新して step2 の総当たり距離表を作ります。これをすべての要素が 1 つのクラスターに含まれるまで繰り返してデンドログラムが出来上がります。

VII-3-2-2. 距離（非類似度）

ここまで、デンドログラムの方向の長さを距離(非類似度)と記述してきました。また、データを空間上の点で表してきたので、グラフ上で距離が認識しやすく、距離の近さが類似度、距離の遠さが非類似度だとなんとなく納得してきました。しかし、空間的な概念である距離と質的な概念である類似度はかなり性質が異なります。ここで用いている距離(非類似度)とは、似ていないこと（非類似）を比喩的に「距離」と表現しているのです。数学的には距離

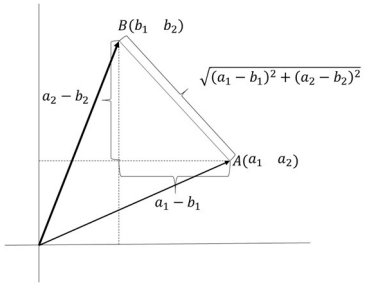


図 133. ピタゴラスの定理とユークリッド距離

とは「距離の公理」を満たす変数です。距離の公理とは、(M1)正値性:正の値であること。(M2)対称性:点 A から B の距離が点 B から A の距離に等しいこと。(M3)三角不等式:点 A から点 B の距離が A から B 間の最短距離であること（任意の点 C を考えて、点 A から任意の点 C をへて B へ向かう距離が必ず点 A から B への距離よりも長いこと）です。距離が負の値であったら困りますし、逆向きに計ったら長さが変わるというのも困ります。また、最短距離より短い距離もあり得ません。これが私たちが持っている世界の認識の仕方です。公理は論理の出発点であり公理の証明はありません。距離としてよく知られているのはユークリッド距離です。ユークリッド距離は、私たちが日常的にいだいている距離の概念を実 n 次元空間(R^n)に拡張したものです。2 次元空間(R^2)で距離を定義するときに、2 点A, Bをの座標を$A = (a_1 \quad a_2),\ B = (b_1 \quad b_2)$として、図 133 のように、中学生の時に習ったピタゴラスの定理を使って、A,B 間の距離を

$$d(A,B) = \sqrt{\sum_{i=1}^{2} {a_i}^2 + \sum_{i=1}^{2} {b_i}^2} = \left(\sum_{i=1}^{2} {a_i}^2 + \sum_{i=1}^{2} {b_i}^2\right)^{\frac{1}{2}}$$

と定義します。これを n 次元空間に拡張したものがユークリッド距離です。これを 3 次元空間のユークリッド距離の定義としても良いのですが、多次元に拡張することを考えて、$\boldsymbol{A}, \boldsymbol{B}$を n 次元のベクトルとして、

$$d(A,B) = \left((\boldsymbol{A}-\boldsymbol{B})\cdot(\boldsymbol{A}-\boldsymbol{B})\right)^{\frac{1}{2}}$$

と定義しておいた方が後々便利です。定義式の外側の括弧の中は同一ベクトルの内積だから、ベクトル間の角度のコサインは 1 で距離の 2 乗になります。

$$d(A,B) = \sqrt{\sum_{i=1}^{n} {a_i}^2 + \sum_{i=1}^{n} {b_i}^2} = \left(\sum_{i=1}^{n} {a_i}^2 + \sum_{i=1}^{n} {b_i}^2\right)^{\frac{1}{2}} = \left((\boldsymbol{A}-\boldsymbol{B})\cdot(\boldsymbol{A}-\boldsymbol{B})\right)^{\frac{1}{2}}$$

この式を見れば、n 次元でも、(M1)、(M2)が成立つことは明らかでしょう。(M3)の三角不等式について考えます。ベクトル$\boldsymbol{A}-\boldsymbol{B}$上にない点$\boldsymbol{C}$**を考えます。**

$$\boldsymbol{A}-\boldsymbol{B} = (\boldsymbol{A}-\boldsymbol{C}) + (\boldsymbol{C}-\boldsymbol{B})$$

$$(\boldsymbol{A}-\boldsymbol{B})\cdot(\boldsymbol{A}-\boldsymbol{B}) = (\boldsymbol{A}-\boldsymbol{C})\cdot(\boldsymbol{A}-\boldsymbol{C}) + 2(\boldsymbol{A}-\boldsymbol{C})\cdot(\boldsymbol{C}-\boldsymbol{B}) + (\boldsymbol{C}-\boldsymbol{B})\cdot(\boldsymbol{C}-\boldsymbol{B})$$

$$d(A,C) = \left((\boldsymbol{A}-\boldsymbol{C})\cdot(\boldsymbol{A}-\boldsymbol{C})\right)^{\frac{1}{2}}$$

$$d(B,C) = \left((\boldsymbol{B}-\boldsymbol{C})\cdot(\boldsymbol{B}-\boldsymbol{C})\right)^{\frac{1}{2}} = \left((\boldsymbol{C}-\boldsymbol{B})\cdot(\boldsymbol{C}-\boldsymbol{B})\right)^{\frac{1}{2}}$$

だから、

$$(\boldsymbol{A}-\boldsymbol{B})\cdot(\boldsymbol{A}-\boldsymbol{B}) = d(A,C)^2 + 2(\boldsymbol{A}-\boldsymbol{C})\cdot(\boldsymbol{C}-\boldsymbol{B}) + d(C,B)^2$$

内積の定義より

$$(\boldsymbol{A}-\boldsymbol{C})\cdot(\boldsymbol{C}-\boldsymbol{B}) = d(A,C)d(A,C)\cos\theta \le d(A,C)d(B,C)$$

$$\because \theta\ \text{はベクトル}(\boldsymbol{A}-\boldsymbol{C})\ \text{と}(\boldsymbol{C}-\boldsymbol{B})\ \text{の角度、}\ -1 \le \cos\theta \le 1$$

$$d(B,C) = d(B,C)$$

$$d(A,C)^2 = (\boldsymbol{A}-\boldsymbol{B})\cdot(\boldsymbol{A}-\boldsymbol{B}) \le d(A,C)^2 + 2d(A,C)d(B,C) + d(B,C)^2$$

$$= \left(d(A,C) + d(B,C)\right)^2$$

平方根をとっても大小関係は変わらないから

$$d(A,C) \le d(A,C) + d(B,C)$$

この証明ですが、座標間の距離を計算してコーシー・シュワルツの不等式を使って証明する方が普通です。コーシー・シュワルツの不等式を証明はいくつかありますが、内積を使って証明するやり方もあります。ですから、内積の方がより原理的なのでこちらの方がより本質的で洗練されていると思います。

ユークリッド距離以外にも距離の公理を充たす変数は、たとえば、分散や相関を考慮したマハラノビスの距離とか、図 133 の$|a_1 - b_1| + |a_2 - b_2|$を距離とするマンハッタン距離とか、その他にも距離として使われている変数はたくさんあって、ネットで探せばたくさん出てきます。筆者がそれらすべて知っているわけがないし、公理を満たしているのかどうか一つ一つ確かめたこともありません。ただ、相関係数やコサイン類似度なども、空間座標的な位置づけとは関係がない類似度を使った距離が「距離の公理」を満たしているのかを確認しておいた方が良いでしょう。そこで形態の類似性などに使われるコサイン類似度を使った距離について確かめてみます。

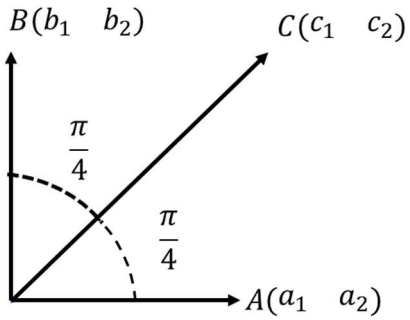


図 134. コサイン類似度を使った距離の例

コサイン類似度はベクトル間の角度のコサインで次の式で表せます。

$$\frac{\boldsymbol{A}\cdot\boldsymbol{B}}{|\boldsymbol{A}||\boldsymbol{B}|}=\cos\theta_{A-B}$$

この式自体は、内積の定義$\boldsymbol{A}\cdot\boldsymbol{B}=|\boldsymbol{A}||\boldsymbol{B}|\cos\theta_{A-B}$を変形しただけのものですが、ここから、$\boldsymbol{A}\cdot\boldsymbol{B}=\sum_{i=1}^{n}a_ib_i$、$|\boldsymbol{A}|=\sqrt{\sum_{i=1}^{n}{a_i}^2}$、$|\boldsymbol{B}|=\sqrt{\sum_{i=1}^{n}{b_i}^2}$とすると、$-1\leq\cos\theta_{A-B}\leq 1$なので、以下のコーシー・シュワルツの不等式が簡単に導き出せます。

$$\frac{\boldsymbol{A}\cdot\boldsymbol{B}}{|\boldsymbol{A}||\boldsymbol{B}|}\leq 1$$

$$\frac{\sum_{i=1}^{n}a_ib_i}{\sqrt{\sum_{i=1}^{n}{a_i}^2}\sqrt{\sum_{i=1}^{n}{b_i}^2}}\leq 1$$

$$\sum_{i=1}^{n}a_ib_i\leq\sqrt{\sum_{i=1}^{n}{a_i}^2}\sqrt{\sum_{i=1}^{n}{b_i}^2}$$

$$\left(\sum_{i=1}^{n}a_ib_i\right)^2\leq\sum_{i=1}^{n}{a_i}^2\sum_{i=1}^{n}{b_i}^2$$

コサイン類似度はマイナスの値を取りますから、そのままでは距離として使えませんそこで、１－コサイン類似度（非類似度）を距離とします。

$$d(A,B)=1-\cos\theta_{A-B}$$

結論を先に言うと１－コサイン類似度（非類似度）は距離の公理を満たしません。成り立たないことの証明はそうならない事例を示せばよいので、一例を示します。図 134 で、ベクトル A と B の角度は$\frac{\pi}{2}$、A と C の角度は$\frac{\pi}{4}$ 、B と C の角度は$\frac{\pi}{4}$です。

$$d(A,B)=1-\cos\theta_{A-B}=1-\cos\frac{\pi}{2}=1$$

$$d(A,C)=1-\cos\theta_{A-C}=1-\cos\frac{\pi}{4}=1-\frac{1}{\sqrt{2}}=0.29893$$

$$d(B,C)=1-\cos\theta_{B-C}=1-\cos\frac{\pi}{4}=1-\frac{1}{\sqrt{2}}=0.29893$$

$$d(A,C)+\ d(B,C)=0.585786$$

で、明らかに C を経由した距離の方が短くなります。1－コサイン類似度は、「距離の公理」を満たす距離ではありません。しかし、だからと言って、1－コサイン類似度を非類似度としてクラスター分析に使ってはいけないということではありません。１－コサイン類似度を距離(非類似度)としても、更新式を使えば、クラスを連結して、総当たり距離表を更新していくことが出来ます。そのようにして、形や性質の違いを強調したクラス分け

が出来ることもしばしばあります。反対に、全く意味の解らないクラス分けになってしまうこともあります。

VII-3-2-3.階層的クラスター分析の使い方

階層的クラスター分析の長所はデンドログラムが出来るので、クラスターがどのように分かれているのか構造が視覚的に理解できることです。非階層的クラスター分析ではクラスの数をあらかじめ決めなくてはなりません。階層的クラスター分析ではクラスの数を事後的に決められるというのもよく言われる長所です。しかし、デンドログラムがあるから事後的に決められるので、本質的には、クラスター構造がわかるデンドログラムを得られるというのがメリットで、それが階層的クラスター分析の目的だと考えて良いでしょう。短所はサンプルサイズが大きくなると計算時間がかかってしまうということです。サンプルサイズが大きくて、K-means 法などの非階層的クラスター分析を使わざるを得ない場合も多いでしょう。コサイン類似度など質的な類似度を評価する場合、K-means 法が使えないという問題もありますが、直観的には、極座標に変換すればやれないことはないような気がします。最近は非階層的クラスター分析にもいろいろなやり方が提案されています。それでも、階層的クラスター分析で視覚的にクラスター構造を確認するというのは大事な作業です。データが多い場合、データデータの一部をランダムサンプリングして、いくつかの方法で階層的クラスター分析をして、使えそうな距離(非類似度)を見つけてから、非階層的クラスター分析をするというのも、確実に効率的に作業を進める方法かもしれません。

VII-3-2-4.Python　階層的クラスタ分析プログラムの実装

Python で、ライブラリーscipy.cluster.hierarchy を使って階層的クラスタ分析を行うプルグラムのコードリストの例を示します。

リスト VII-3-2-i.分析準備・主成分分析

```
#[A]必要なライブラリーの読み込み
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
from scipy.cluster.hierarchy import linkage,dendrogram,fcluster
#[B]データの読み込み
df =pd.read_csv("sample10.csv")
xn,D=df.shape
D1=D-1
#データフレームをつくる
dfX=pd.DataFrame(df)
X=df.values
X=np.delete(X,0,1)
#列名を付ける
for i in range (D):
```

```
    dfX=dfX.rename(columns={i:"X"+str(i+1)})
print (dfX)
D1=D-1
#[C]標準化後主成分分析を実行
import urllib.request
import matplotlib.pyplot as plt
import sklearn    #機械学習のライブラリ
from sklearn.decomposition import PCA   #主成分分析
from sklearn.preprocessing import StandardScaler #標準化
from IPython.display import display
#標準化
std_sc = StandardScaler()
std_sc.fit(X)
std_data = std_sc.transform(X)
std_data_df = pd.DataFrame(std_data)
display(std_data_df)
#主成分分析の実行
pca = PCA()
pca.fit(std_data_df)
# データを主成分空間に写像
pca_cor = pca.transform(std_data_df)
print(pca.get_covariance()) # 分散共分散行列
# 固有ベクトルのマトリックス表示
eig_vec = pd.DataFrame(pca.components_.T, ¥
                       columns = ["PC{}".format(x + 1) for x in range(len(std_data_df.columns))])
display(eig_vec)
# 固有値
eig = pd.DataFrame(pca.explained_variance_, index=["PC{}".format(x + 1) for x in
range(len(std_data_df.columns))], columns=['固有値']).T
display(eig)
# Rによるソースコードだと、固有値（分散）ではなく標準偏差を求めているので、一応標準偏差も計算。
# 主成分の標準偏差
dv = np.sqrt(eig)
dv = dv.rename(index = {'固有値':'主成分の標準偏差'})
display(dv)
# 寄与率
ev = pd.DataFrame(pca.explained_variance_ratio_, index=["PC{}".format(x + 1) for x in
range(len(std_data_df.columns))], columns=['寄与率']).T
display(ev)
# 累積寄与率
t_ev = pd.DataFrame(pca.explained_variance_ratio_.cumsum(), index=["PC{}".format(x + 1) for x in
range(len(std_data_df.columns))], columns=['累積寄与率']).T
display(t_ev)
# 主成分得点
print('主成分得点')
cor = pd.DataFrame(pca_cor, columns=["PC{}".format(x + 1) for x in range(len(std_data_df.columns))])
display(cor)
PC=cor.values
dfS=pd.concat([dfX,cor],axis=1)
S=dfS.values
```

*[A]*python の標準装備にない機能を library から読み込む。

pandas:csv などのテキスト r ファイルを読み込んだり、データフレームを作る

numpy:多次元配列間で数値計算を行う機能。

matplotlib.pyplot:グラフを作る機能
sklearn.model_selection.train_test_split:配列や行列の中から、学習用のデータとテスト用のデータをランダムに取り分ける機能
scipy.cluster.hierarchy.linkage:階層的クラスター分析で非類似度を決めて連結する
.dendrogram:連結の結果にしたがってデンドログラムを描く
.fcluster:決められてクラス数にしたがって、各データがしょぞくするクラスの番号を返す。

*[B]*エクセルで作った csv ファイルを読み込みます。csv ファイルの 1 行目にはデータを識別する番号（あらかじめ何かのクラスに属している場合はその識別番号）があるので、それを残してデータフレームを作り列名(X1,X2…….)をつけたもの(dfX)と、１行目を取り除いた行列(X)という内容が同じ 2 つのデータセットを作ります。

*[C]*すべての変数を使ってクラスター分析をするのは計算の負荷が大きく時間がかかります。また、普通いくつかの説明変数は相関を持っています。そこで、主成分分析をして、直交するいくつかの主成分でデータを表現した方が無駄がありません。また、今のところ Python の scipy.cluster.hierarchy や KMeans はユークリッド距離に大きく依存していて、他の距離(非類似度)を使うと、ward 法のような一般的な連結法が使えなかったり、KMeans の中に距離（metric）の指定が出来なかったりします。自分でプログラムを作れば、更新式を使って連結法を使えるようにしたり、ユークリッド距離以外でも K-means 法を使えるようにしたりできますが、プログラムのコードリストを新しく書くのは面倒です。主成分得点は直交系で共分散が 0 ですから、ユークリッド距離もマハラノビス距離も同じだから、マハラノビス距離で計算することの代替になります。主成分分析も機械学習の一つで、主成分分析後に視覚的にクラスター分けが出来る場合もあります。ですから、まず、主成分分析をして、変数を圧縮しておくというのは一つのお作法です。

最後の 3 行で、主成分得点(cor)を行列（PC）として保存し、もう一つ、データフレームとして、dfX の列の後ろに、主成分得点(cor)のデータフレームを結合して、主成分得点のデータを含むデータフレーム(dfS)をつくり、さらに、データ部分だけを取り出した配列（S）を作っておきます。

リスト VII-3-2-ii.主成分の数を決定し、データをシャッフルしてトレーニングデータとテストデータに分けて保存

```
#[A]主成分寄与率・累積寄与率から主成分の数を決定する
P=2
#[B]データをシャッフルしトレーニングデータとテストデータに分割して保存
TrainingRatio=0.5
S_train, S_test = train_test_split(S, train_size=TrainingRatio, random_state=1)
n,cul=S_train.shape
T_train=np.zeros((n,1))
X_train=np.zeros((n,D1))
PC_train=np.zeros((n,P))
```

```
for i in range(1):
    T_train[:,i]=S_train[:,i]
for i in range(D1):
    X_train[:,i]=S_train[:,1+i]
for i in range(P):
    PC_train[:,i]=S_train[:,1+D1+i]
n,col=S_test.shape
T_test=np.zeros((n,1))
X_test=np.zeros((n,D1))
PC_test=np.zeros((n,P))
for i in range(1):
    T_test[:,i]=S_test[:,i]
for i in range(D1):
    X_test[:,i]=S_test[:,1+i]
for i in range(P):
    PC_test[:,i]=S_test[:,1+D1+i]
```

*[A]*主成分分析の結果（寄与率・累積寄与率）を見て、分析対象にする主成分の数を決定します（寄与率の小さい成分を無視する。）。この例では説明変数が二つしかないから P=2 とするしかありません。

*[B]*データを訓練用のデータ、テスト用のデータに分けます。理由は２つあります。一つはクラスター分析の結果が少数の離れ値や部分的に偏りに対して過剰に適合（オーバーフィッティング）して、一般性がないクラスを分けをしてしまうことを防ぐことです。オバーフィティングしていれば、訓練用データとテスト用データの結果が大きく異なります。２つ目の理由は、時間の節約です。階層的クラスター分析は時間がかかります。階層的クラスター分析はクラスターの構造がわかるので、事後的に根拠を持ってクラスの数を決められます。連結法や距離の組み合わせがたくさんあるので、いくつかの組み合わせを試してみるべきですが、それを効率的にやるためにはサンプルサイズを小さくする必要があります。

TrainingRatio=r で training データとして取り出す割合を決めます。この場合は r=0.5 (50%) としました。S_train, S_test = train_test_split(S, train_size=TrainingRatio, random_state=1): がそのコードです。取り分ける元の配列は S です。Train_saize =TrainingRatio とします。ここではランダム抽出に用いる乱数に 1 という乱数を使いました（番号で指定しますが何番でも構いません。別に指定しなくても良いのですが、使う乱数を固定しておくと、データの取り出しで後々便利なことがあります。わからなくならないように書いておいた方が良いでしょう。）。

取り出した、S_train も S_test も最初の１行はデータ識別番号（あるいはクラス）です。これを、T_train あるいは T_test として取り出します。次にもとの変数のデータが変数の数の行数（D1）続きます。これを X_train あるいは X_test として取り出します。最後に、主成分得点のデータが選択した主成分の数だけ続きます。最終的に T_train, X_train,PC_test、T_train,X_train,PC_train の配列が取り出せます。

リスト VII-3-2-iii.データ分布の確認（クラスの識別ナシ）

```
#データ分布の確認
```

```
#[A]散布図の描画法を定義する
def show_data(x):
    plt.plot(x[:,x0],x[:,y0],linestyle='none',marker='o',markeredgecolor='black',color="white",alpha=0.
    8)
    plt.grid(True)
def show_data1(x,t):
    col=["b","r","g","y","w","c","m","k"]
    for c in range (C):
        plt.plot(x[t[:,0]==c+1,x0],x[t[:,0]==c+1,y0],linestyle='none',marker='o',markeredgecolor='black
        ',color=col[c],alpha=0.8)
    plt.grid(True)#元データの分布
#[B]元データの散布図
#変数の選択
x=1
y=2
x_range=[-2,2] #項目 1 の範囲
y_range=[-2,2] #項目 2 の範囲
x0=x-1
y0=y-1
#実行
plt.figure(1,figsize=(8,3.7))
plt.subplot(1,2,1)
show_data(X_train)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("X"+str(x))
plt.ylabel("X"+str(y))
plt.title('Training data')
plt.subplot(1,2,2)
show_data(X_test)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("X"+str(x))
plt.ylabel("X"+str(y))
plt.title('Test data')
plt.show()
#[C]主成分得点のデータ分布の確認
#主成分の選択
x=1
y=2
x_range=[-3,3] #項目 1 の範囲
y_range=[-3,3] #項目 2 の範囲
x0=x-1
y0=y-1
#実行
plt.figure(1,figsize=(8,3.7))
plt.subplot(1,2,1)
show_data(PC_train)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("PC"+str(x))
plt.ylabel("PC"+str(y))
plt.title('PC_Training data')
plt.subplot(1,2,2)
```

```
show_data(PC_test)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("PC"+str(x))
plt.ylabel("PC"+str(y))
plt.title('PC_Test data')
plt.show()
```

データの分布を元のデータの変数の 2 次元の組み合わせ、主成分得点の 2 次元の組み合わせの平面図で確認します。

*[A]*散布図の描き方を定義します。散布図には 2 通りあって、show data (x)はクラスの識別がわからない状態、show data1(x,t)はデータが所属するクラスがわかっている場合に使い、クラスをマーカーの色で識別して示します。ただし、現時点では色の指定が 8 個しかありません、ローマ字 1 文字で色指定できるのが 8 色までだからです。それ以上の色が必要な場合は、matplotlib で調べて、別の色指定のしかたで指定してください。

元の変数から 2 つを選びその組み合わせで 2 次元の散布図を作ります。

*[B]*元データを使った散布図を作ります x=, y=のところで、変数を指定します。x=1 なら

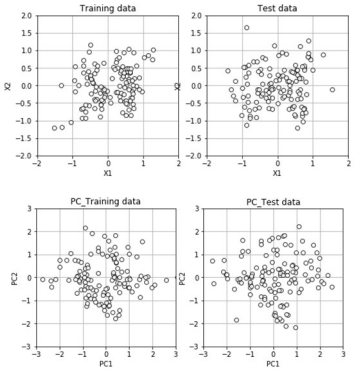


図 135.sample 10 のデータ分布

ば、横軸に変数 X1 をとり、y=2 ならば縦軸に変数 X2 をとります。続いて描画するデータの範囲を指定します。x_range=[-2,2]で横軸の範囲を-2 から 2 に指定、y_range=[-2,2]で縦軸の範囲を-2 から 2 に指定しています。

*[C]*主成分元を使った散布図を作ります x=, y=のところで、主成分を指定します。x=1 ならば、横軸に第一主成分(PC1)をとり、y=2 ならば縦軸に第二主成分（PC2）をとります。続いて、描画するデータの範囲を指定します。x_range=[-3,3]で横軸の範囲を-3 から 3 に指定、y_range=[-3,3]で縦軸の範囲を-3 から 3 に指定しています。実行すると、図 135 のような散布図が得られます。一見したところ training data と test data で分布に大きな違いはないようです。また、いくつかのグループがありそうに見えますが、あまりはっきりしません。これは主成分得点で分布を表しても同じです。Sample 10 はもともとあった 5 つのグループを重ね合わせたものです。元のグループを識別できるようにドットに色をつけてみました。

リスト VII-3-2-iv.データ分布の確認（クラスの識別つき）

```
#データ分布の確認(クラス識別)
#[A]元データの分布
C=5 #クラスの数
#変数の選択
x=1
y=2
x0=x-1
y0=y-1
x_range=[-2,2] #項目 1 の範囲
y_range=[-2,2] #項目 2 の範囲
plt.figure(1,figsize=(8,3.7))
plt.subplot(1,2,1)
show_data1(X_train,T_train)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("X"+str(x))
plt.ylabel("X"+str(y))
plt.title('Training data')
plt.subplot(1,2,2)
show_data1(X_test,T_test)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("X"+str(x))
plt.ylabel("X"+str(y))
plt.title('Test data')
plt.show()
#[B]主成分得点のデータ分布の確認
#主成分の選択
x=1
y=2
x0=x-1
y0=y-1
x_range=[-3,3] #項目 1 の範囲
y_range=[-3,3] #項目 2 の範囲
plt.figure(1,figsize=(8,3.7))
plt.subplot(1,2,1)
show_data1(PC_train,T_train)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("PC"+str(x))
plt.ylabel("PC"+str(y))
plt.title('PC_Training data')
plt.subplot(1,2,2)
show_data1(PC_test,T_test)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("PC"+str(x))
plt.ylabel("PC"+str(y))
plt.title('PC_Test data')
plt.show()
```

内容は II-3-2-iii.と同じで、用いる変数・主成分を指定指定して散布図を描きます。グラフを描画する範囲も新たに指定します。この例で使っているサンプルデータ(sample10)は人

為的に合成したデータでもともと所属するグループがわかっているので、show_data1(x,t)を使ってクラスを色で識別して表すことが出来ます(図 136)。結果は下図のように出てきます。この図を見ると、元データの分布と主成分の分布は鏡像の関係で裏返しになっていることがわかります。このデータは判別分析の例で使ったデータの各クラス内の距離の偏差を 10 倍に拡大したものです。元データでも主成分得点でも、白のドットで示した中央に分布する 5 番目のクラスの分布が大きく広がり、他のクラスと識別できないことがわかります。つまり、視覚的に、クラス 5 を他のクラスと分けて識別することはできません。

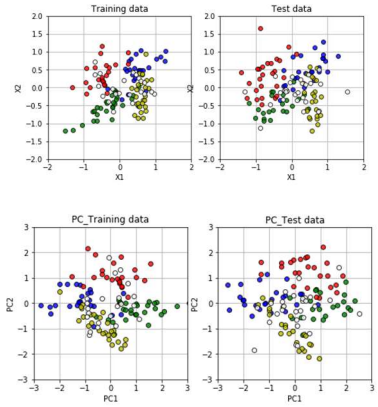


図 136.sample 10 のデータ分布(元のクラスを色で識別)

以下、距離(非類似度)と連結法の組み合わせを変えていくつか階層的クラスター分析を実施します。

リスト VII-3-2-v. ユークリッド距離を非類似度として単連結法でクラスター分析

```
#ユークリッド距離を非類似度として単連結法でクラスター分析
z1 = linkage(X_train, metric='euclidean', method="single")
z2 = linkage(X_test, metric='euclidean', method="single")
# 結果を可視化
plt.figure(1,figsize=(8,3.7))
plt.subplot(1,2,1)
dendrogram(z1)
plt.title("Training data.euclid-single")
plt.subplot(1,2,2)
dendrogram(z2)
plt.title("Test data.euclid-single")
plt.show()
```

ユークリッド距離を使って単連結法でデンドログラムを作成します。z1 = linkage(X_train, metric='euclidean', method="single")、z2 = linkage(X_test, metric='euclidean', method="single")で分析対象のデータとして、X_train と X_test、距離(非類似度)としてユークリッド距離、連結法に単連結法を選択しています。

リスト VII-3-2-vi. 上記のクラスター分析の結果を散布図で表す。

```
#クラス分けの結果を散布図で表示
#[A]クラスの数を決める
```

```
C=4
#変数の選択
x=1
y=2
x0=x-1
y0=y-1
x_range=[-2,2] #項目１の範囲
y_range=[-2,2] #項目２の範囲
#[B]training data の分布
clusters = fcluster(z1, t=C, criterion='maxclust')#用いるデンドログラムを指定
n,nn=X_train.shape
for i in range(n):
    T_train[i,0]=clusters[i]
plt.figure(1,figsize=(8,3.7))
plt.subplot(1,2,1)
show_data1(X_train,T_train)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("X"+str(x))
plt.ylabel("X"+str(y))
plt.title('Training data')
#[C]test data の分布
clusters = fcluster(z2, t=C, criterion='maxclust')  #用いるデンドログラムを指定
n,nn=X_test.shape
for i in range(n):
    T_test[i,0]=clusters[i]
plt.subplot(1,2,2)
show_data1(X_test,T_test)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("X"+str(x))
plt.ylabel("X"+str(y))
plt.title('Test data')
plt.show()
```

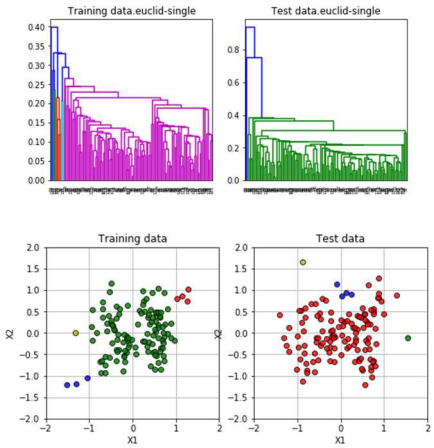


図 137. Sample 10 のデータをユークリッド距離を非類似度として
単連結法でクラスター分析した結果

*[A]*デンドログラムの形からクラスの数を決定し、グラフの縦軸横軸の範囲も決めます。この場合は、C=4、 x=1、 y=2、 x_range=[-2,2]、 y_range=[-2,2]としました。

[B]で training data の分布とクラス分けを示します。clusters = fcluster(z1, t=C, criterion='maxclust')で z1 のデンドログラムを使うことを指定しています。
[C]で test data の分布とクラス分けを示します。clusters = fcluster(z2, t=C, criterion='maxclust')で z2 のデンドログラムを使うことを指定しています。
Sample10 の分析結果は図 137 のようになりました。デンドログラムは、単調に近いものをつなげていって、最後に離れたいくつかの点をつなげていったような構造になっています。散布図をみると、数の多い中央のクラスターがまとまって拡大しながら単調に周辺を吸収していることがわかります。training data と test_data のクラスターの構造が異なり、クラスターを見つけるという機能を果たしていません。単連結法がこの場合不適切であることを示しています。以下、様々な距離(非類似度)や連結法を試してみました。

リスト VII-3-2-vii. 以下（各種のデンドログラムの作成）

```
#＊＊＊＊＊を非類似度として＊＊＊＊＊法でクラスター分析
z3 = linkage(X_train, metric='euclidean', method="complete")  #[A]
z4 = linkage(X_test, metric='euclidean', method="complete") #[B]
# 結果を可視化
plt.figure(1,figsize=(8,3.7))
plt.subplot(1,2,1)
dendrogram(z3)
plt.title("Training data.euclid-complete")
plt.subplot(1,2,2)
dendrogram(z4)
plt.title("Test data.euclid-complete")
plt.show()
```

[A][B]: linkage(DATA, metric='距離(非類似度)', method="連結方法")、DATA 部分は T_train, T_test, PC_train, PC_test のいずれかを選択、metric='距離(非類似度)'は距離(非類似度)を選択、method="連結方法"は T_train, T_test、ユークリッド距離、完全連結法を選択している。

リスト VII-3-2-viii. 以下（各種のデンドログラムからデータ分布とクラス分けを表示）

```
#クラス分けの結果を散布図で表示
#[A]クラスの数を決める
C=5
#変数の選択
x=1
y=2
x0=x-1
y0=y-1
x_range=[-2,2] #項目 1 の範囲
y_range=[-2,2] #項目 2 の範囲
#[B]training data の分布
clusters = fcluster(z3, t=C, criterion='maxclust')#クラスの数を決める
n,nn=X_train.shape
for i in range(n):
    T_train[i,0]=clusters[i]
plt.figure(1,figsize=(8,3.7))
plt.subplot(1,2,1)
show_data1(X_train,T_train)
```

```
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("X"+str(x))
plt.ylabel("X"+str(y))
plt.title('Training data')
#[C]test data の分布
clusters = fcluster(z4, t=C, criterion='maxclust')
n,nn=X_test.shape
for i in range(n):
    T_test[i,0]=clusters[i]
plt.subplot(1,2,2)
show_data1(X_test,T_test)
plt.xlim(x_range)
plt.ylim(y_range)
plt.xlabel("X"+str(x))
plt.ylabel("X"+str(y))
plt.title('Test data')
plt.show()
```

*[A]*デンドログラムの形からクラスの数を決定し、グラフの縦軸横軸の範囲も決めます。この場合は、C=5、x=1、y=2、x_range=[-2,2]、y_range=[-2,2]としました。

[B]で training data の分布とクラス分けを示します。clusters = fcluster(z3, t=C, criterion='maxclust')で z3 のデンドログラムを使うことを指定しています。

[C]で test data の分布とクラス分けを示します。clusters = fcluster(z4, t=C, riterion='maxclust')で z4 のデンドログラムを使うことを指定しています。

ライブラリーscipy.cluster.hierarchy から linkage, dendrogram, fcluster をインポートして階層的クラスタ分析を行う場合、連結法（引数 method）と距離(非類似度)の指定のためのコードの一覧表を示します。

表 69.scipy.cluster における連結法(引数 method)のコード

連結法	コード（引数 method=
単連結法（simple linkage method）	single
完全連結法(complete linkage method)	complet
群平均法(group average method)	average
重み付き平均法（weighted average method）	weighted
重心法(centroid method)	centroid
ウォード法(Ward's method)	ward

表 70.scipy.cluster における距離(非類似度)(引数 metric)のコード

距離(非類似度)	コード（metric=）	内容(n 次元データ)
ユークリッド距離	euclidian	$\left(\sum_{j=1}^{p}\left(\lvert x_j - y_j\rvert^2\right)\right)^{\frac{1}{2}}$

ユークリッド距離の２乗	sqeuclidian	$\sum_{j=1}^{p}\left(\left\|x_j-y_j\right\|^2\right)$
標準化されたユークリッド距離	seuclidian	元データを偏差で割る
マハラノビスの距離	mahalanobis	$\left((\boldsymbol{x}-\boldsymbol{y})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{y})\right)^{\frac{1}{2}}$
ミンコフスキー距離	mincowski	$\left(\sum_{j=1}^{p}\left(\left\|x_j-y_j\right\|^p\right)\right)^{\frac{1}{p}}$
マンハッタン距離	cityblock	$\sum_{j=1}^{p}\left\|x_j-y_j\right\|$
キャンベラ距離	canberra	$\sum_{j=1}^{p}\frac{x_j-y_j}{x_j+y_j}$
1-Pearson の相関係数	correlation	$1-\frac{S_{xy}}{\sqrt{S_{xx}S_{xy}}}$
コサイン非類似度	cosine	$1-\frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\lvert\boldsymbol{x}\rvert\lvert\boldsymbol{y}\rvert}$
ジャカード非類似度	jaccard	$1-\frac{\lvert A\cap B\rvert}{\lvert A\cup B\rvert}$
ダイス非類似度	dice	$1-\frac{2\lvert A\cap B\rvert}{\lvert A\rvert+\lvert B\rvert}$

以下、いくつかの事例を示します。図 138 は sample10 をユークリッド距離を使って完全連結法でデンドログラムを作成した例です。単連結法よりは改善されていますが、クラスターの位置が Training data と Test data で違っているようです。たまたまたま存在する最長距離のデータに過適応してクラス分けが決まっているように感じます。図 139 は、ユークリッド距離を使って Ward 法で連結した例です。デンドログラムを注意深く見ると、Training data も Test data で、クラスターの枝分かれが上の方で４つと５つに違っているのですが、test data のデンドログラムはクラスが５つという可能性を強く推しているようです。実際、クラス数=5 で分布図を書いてみると、Training data でも Test data でも中央部に一つのまとまったクラスが描かれます。Sample10 は筆者が５つのグループを角度や位置を変えて重ねあわせて作ったデータで、もともとクラスが５つあったのです。それを知っているので、中央部のクラスがもともとのクラス５に由来するのだと、私たちは解釈しますが、普通はそんな事前情報はありません。私たちが分析開始時点で目にするのは、図 135 のような、色分けのない白いドットが分布している散布図です。つまり、この分析

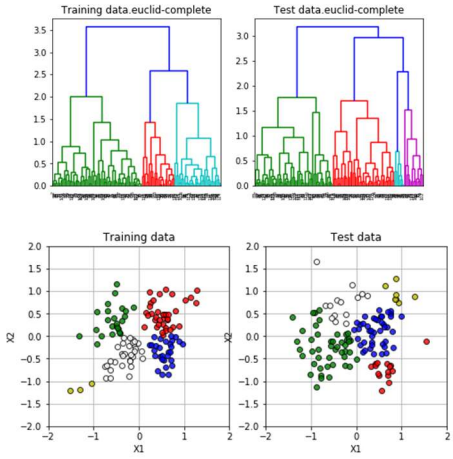


図 138. Sample 10 のデータをユークリッド距離を非類似度として
完全連結法でクラスター分析した結果

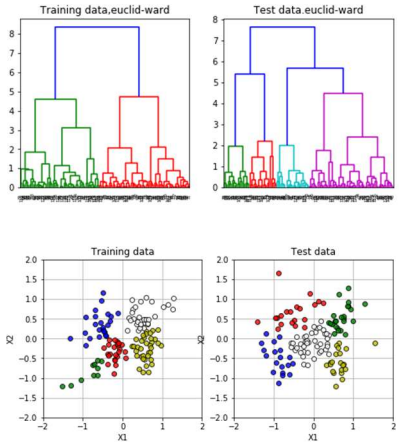


図 139. Sample 10 のデータをユークリッド距離を非類似度として
Ward 法でクラスター分析した結果

をすることによって、はじめて、中央部のクラスの存在に気付くということです。これは発見です。この中央部のクラスが何であるか、このクラスの特性を他のクラスの特性と比べた時にわかってきます。その時点で、このクラスター分けの妥当性が評価できます。つまり、クラスター分けの結果は研究・学習の原点（きっかけ）で、これが教師なし学習の本質です。今回は、たまたま、ユークリッド距離を使って Ward 法で連結した結果、新発見に至りました。Ward 法が唯一絶対の連結法というわけではありませんが、試しにユークリッド距離を使って重心法で連結すると図 140 のようになります。非常に微妙で、training data では赤で示した中央のクラスが、元々のクラスター 5 を認識しているようにも見えますが、むしろ、左下の青い 3 点を外れ値として処理したために、一見中央にクラスがあるように見えるだけです。このことは test data の散布図からも明らかです。Ward 法では分散を最小化しながら連結しているので、分布の重なり合いによって埋もれてしま

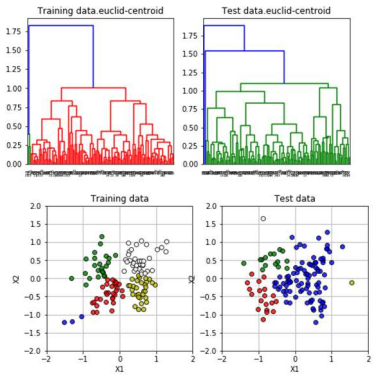


図 140. Sample 10 のデータをユークリッド距離を非類似度として
重心法でクラスター分析した結果

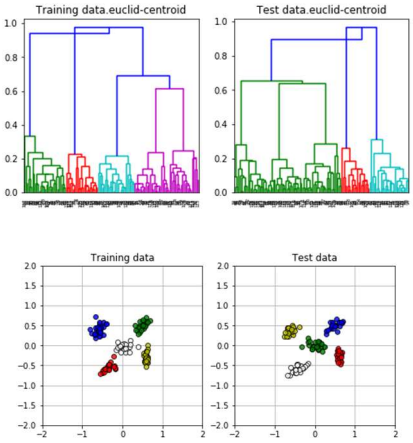


図 141. Sample 11 のデータをユークリッド距離を非類似度として
重心法でクラスター分析した結果

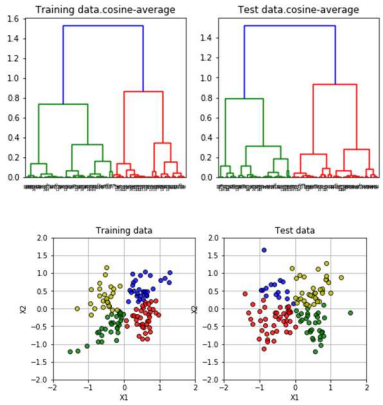


図 142. Sample 10 のデータをコサイン非類似度を距離として
群平均法でクラスター分析した結果

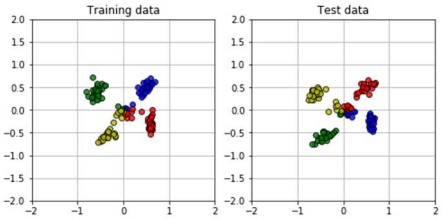


図 143. Sample11 のデータをコサイン非類似度を距離として
群平均法でクラスター分析した結果

った潜在的なクラスターを見つけるのに適しています。そのため、連結方法としては最もよく使われています。図 141 は、sample10 の各クラス内の偏差を 10 分の 1 にした sample11 のデータを同じ方法でクラスター分析した結果です。デンドログラムを見ると、training data でも test data でも最後に連結したクラスタ間の距離が、連結前のクラスタ内のクラス間の距離よりも短くなっています。これを距離の逆転と言いますが、重心法だと距離の逆転が起こることがあります。連結していくにしたがってクラスの代表点間の距離が遠くなっていくことを単調性と言います。重心法には単調性がないのです。これが気になるという立場と気にならないという立場があり得ますが、重心法を使うのであればその可能性を許容するしかありません。そうではあっても Sample11 の場合は、元のクラスの分布をほぼ完璧にとらえています。つまり、重なり合いが少なければ、重心法でもクラスの違いを認識することはできます。反対に言えば、重なり合いが多いデータから何かを拾い出すことが目的ならば、Ward 法を試してみるということです。実用的な意味で ward 法の問題点は、Python の scipy.cluster.hierarchy では、距離(非類似度)と連結法の組み合わせに制約が多く、ユークリッド距離以外の距離を選ぶと ward 法が使えないことが多いということです。更新式を使えば ward 法で連結することが出来る距離(非類似度)はもっと多いと思います。その点では R を使う方が自由度が高いかもしれません。これは、非階層的なクラスター分析でも同じで、Python の sklearn.cluster ではユークリッド距離以外で K-means 法が使えるものは限られています。図 142 はコサイン非類似度（1 －コサイン類似度）を距離として sample10 をクラスター分液をした例です。Ward 法が使えないので群平均法で連結しました。コサイン類似度は、原点から各データ点へのベクトルの角度だけを問題にします。距離は無視しています。Sample 10 は各クラスを第 I 象限、第 II 象限、第 III 象限、第 IV 象限に放射上に一つづつ 4 つのクラスの分布中心をおいて、中心に 5 番目のクラスの分布中心をおいています。デンドログラムはきれいに 4 つに分かれます。しかし、5 番目のクラスは全象限にまたがって存在しますから、一つのクラスとして認識されません。念のためにため sample 11 について、同じ方法でクラスター分析すると図 143 のように中央部の塊が角度によってきれいに 4 分割されます。この事例では、コサイン非類似度を距離(非類似度)として使うことは不適切ですが、原点からの距離を無視して角度だけが特性となる場合、つまり、比率のようなもの類似性だけを考えるのであれば有力なツールになります。コサイン非類似度は心理学、文学、社会学のような分野では多用されま

す。
変数間の相関が強い場合には、マハラノビスの距離を使って、クラスター分析することが考えられますが、scipy.cluster.hierarchy ではマハラノビスの距離を使った場合、重心法や ward 法が使えません。重心というのがユークリッド空間的な概念だと考えているからだと思いますが、マハラノビスの距離でも、マンハッタン距離あるいはその他の距離でも、すべての点への距離の総和が最小になる点という意味での重心は考えられますし、更新式を使ってプログラムを組むことは可能だと思います。ただ、そのプログラムを組むのは大変ですし、計算量が膨大になるかもしれません。そこで、それは将来の課題とします。マハラノビスの距離を使った場合でも群平均法は使えますから、マハラノビスの距離を使って群平均法で Sample10 をクラスター分析してみましたが(図 144)、期待どおりに安定してきれいにクラスがわかれません。これは、マハラノビスの距離が不適切なのではなくて、この場合、群平均法が不適切なのだと考えるべきでしょう。主成分分析は、データを直交成分に圧縮する作業ですから、主成分分析後に主成分得点のユークリッド距離を使って ward 法でクラスター分析することは、マハラノビスの距離を使って ward 法でクラスター分析することの代替になると考えられます。図 145 は、主成分得点間のユークリッド距離を使って ward 法でクラスター分析した結果です。やはり、中心部のクラスの存在を完璧にとらえています。
このように、階層的クラスター分析にはいろいろな」方法があります。何を捉えようとしているのを意識しながら、連結方法や距離の特性を考えて、いくつかの方法を試してみることが新発見につながります。

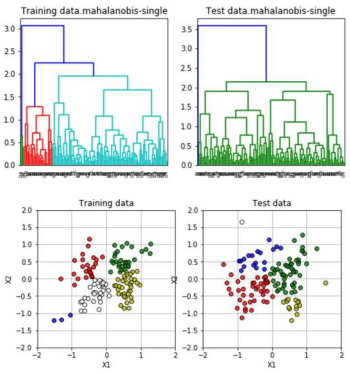
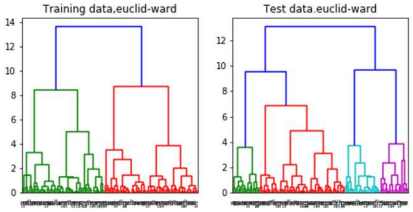


図 144. Sample 10 のデータをマハラノビスの距離を距離(非類似度)として
群平均法でクラスター分析した結果



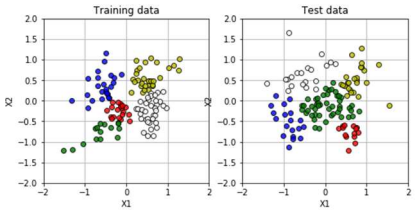


図 145. Sample 10 のデータの主成分得点を間のユークリッド距離として距離(非類似度)として Ward 法でクラスター分析した結果