階層クラスター分析はちょっときまぐれ

松田　史生[1]*・川瀬　雅也[2]

C君：あの，ちょっといいですか？

B君：やあ．Aさんは学会で留守だけど．

C君：知ってます．

B君：なんか聞きたいことでもあるの？ちょうどこっちもC君に聞きたいことがあるんだよね．この連載の第5回と6回で主成分分析を取り上げたじゃん．その後Aさん，主成分分析をえらく気に入って，研究進捗報告会とか，今行ってる学会でも主成分分析をバリバリ使った発表をしているんだわ．

C君：Aさんらしいですね．どんどん新しい研究ツールを活用してて．

B君：でもさ，主成分分析ってちょっとわかりにくくない？それで「階層クラスター分析」を今試していたところなんだよね．

C君：階層クラスター分析も論文でよく見ますよね．オミクスデータの中の類似の挙動を示す遺伝子や代謝物をクラスターに分類したり，一目瞭然でわかりやすいですよね．A先輩にも教えてあげたらいいんじゃないですか？

B君：それでさ，もう何がなんだかわかんなくなってきちゃった．

階層クラスター分析の基礎

C君：え……そんなことになっているんですか？

B君：授業ではこう習うわけ．たとえば，フィッシャーのアヤメのデータ［3種のアヤメ（*Iris setosa*，*I. virginica*，*I. versicolor*）各50個体の花弁（Petal）とがく（Sepal）それぞれの長さ（length）と幅（width）を計測したもの．単位はcm．4次元で総計150個体のデータ．“iris.csv”を生物工学会のHPからダウンロードして，C：直下のpydataに置く］のうち，4つを取り出したのが図1とするよね．

C君：あ……ええ，次は距離関数が出てくるんですよね．たとえば，ユークリッド距離だと2点間の直線距離をしらべるのでaとbの間の距離Dは

$$D=\sqrt{(5.1-4.9)^2+(3.5-3.0)^2+(1.4-1.4)^2+(0.2-0.2)^2}$$

$D=0.54$ですね．

B君：全組合せで計算したのが距離行列だよね（図1）．この中で一番距離Dが近い（小さい）ものはa–bなので，まずabを1つにまとめる（図2の1）．Complete法では，まとめるときにクラスター間でもっとも離れたサンプル間の距離を採用する．つまり，(ab) とcの距離Dは，4.10（aとcとの距離）と4.0のうちの4.1になる．でもこの場合，次に距離が近いcとdをまとめる．最後に（ab）と（cd）の間の距離Dは5.34（bとdとの距離）となる．

C君：こうやってできたのがデンドログラムっていうんですよね．すごいシンプルでわかりやすいじゃないですか．

B君：最初はそう思ったんだよ．できたデンドログラムもわかりやすいし，付き合いやすそうだなと．

C君：違うんですか？

B君：そうなんだよ．ちょっとPythonで階層クラスター分析をやってみてよ．

	Sepal.Lentgh	Sepal.Width	Petal.Length	Petal.Width
a	5.1	3.5	1.4	0.2
b	4.9	3.0	1.4	0.2
c	7.0	3.2	4.7	1.4
d	6.3	3.3	6.0	2.5

図1．アヤメのデータを抜粋した例

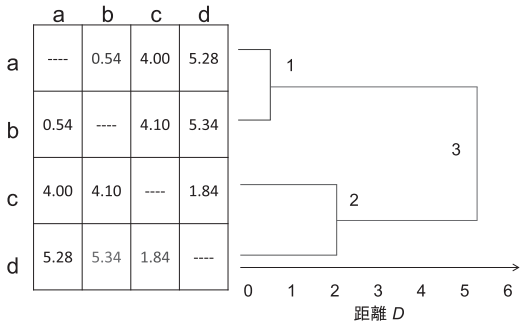

図2．距離行列とデンドログラム

著者紹介　[1]大阪大学大学院情報科学研究科（教授）　E-mail: fmatsuda@ist.osaka-u.ac.jp
[2]長浜バイオ大学（教授）　E-mail: m_kawase@nagahama-i-bio.ac.jp

Pythonで階層クラスター分析

C君：……こんな感じですかね（リスト1）.

リスト1

```
import pandas, numpy # pandas numpyをインポート
from scipy.cluster import hierarchy # Scipyの階層化クラスタリングモジュールをインポート
from matplotlib import pyplot # pyplotのモジュールをインポート
#irisのデータ読み込み
iris = pandas.read_csv("iris.csv", sep=',')
data = iris.iloc[:, 1:5]#データの切り出し
species = iris.iloc[:, 5]#種データの切り出し
#階層化クラスタリングの実行
hct = hierarchy.linkage(data, metric = 'euclidean', method= 'complete')
hierarchy.dendrogram(hct, labels = list(species)) #デンドログラムを作成
result = numpy.ndarray.flatten(hierarchy.cut_tree(hct,3))
#pandas.crosstabで結果を集約
print(pandas.crosstab(species,result))
pyplot.show() #デンドログラムを表示
```

C君：“Python　階層クラスター分析”で検索して出てきた例をいくつか参考にしています．Scipyに階層クラスター分析が実装されています．scipy.clusterからhierarchyをインポートしているのがそれです．距離関数をユークリッド距離(metric = 'euclidean')，結合方法をcomplete法(method= 'complete')にしました（図3）.

B君：デンドログラムの下のアヤメの種名が見にくいけどまあいいや.

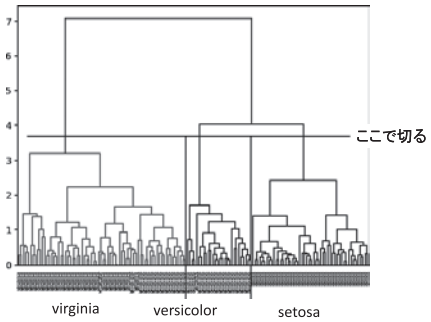

図3．実行結果（画像に追記してある）metric = 'euclidean', method = 'complete'

C君：図3に追記しておきました．スクリプトでは，hierarchy.cut_tree(hct,3)で3つのクラスターに分けて，numpy.ndarray.flattenでデータを成型して，pandas.crosstab(species,result)っていうので，集約した結果をコンソールに表示しました.

実行結果（コンソールに表示される）

col_0	0	1	2
Species			
setosa	50	0	0
versicolor	0	23	27
virginica	0	49	1

B君：一番右のクラスター0にはsetosaが50個体，右から2番目のクラスター1はversicolorが23個体とvirginicaが49個体含まれている．さらに，クラスター2には，versicolorが27個体とvirginicaが1個体が分類されたってことになる.

C君：以前やった主成分分析の結果ともよく合致しますよね.

距離関数と結合方法で結果が変わる

B君：でも，「こうなるはず」という結果になったからそれでいい，なんて説明すると，X教授に怒られそうじゃん．それで距離関数と結合方法を変えてみても，結果が変わらないから，良い結果が得られたって言えるのかなと思って，調べてみたんだわ.

C君：“scipy.cluster.hierarchy.linkage”で検索してScipyのユーザーガイドのページを見つけました．なるほど結合方法にはsingle，complete，average，weighted，centroid，median，wardの7種類が，距離関数はcityblock，correlation，euclideanなど全部で23種ありますね（各方法の意味はScipyのユーザーガイドを参照のこと）.

B君：それでさ，まず距離関数euclidean（ユークリッド距離）に固定して，結合方法をsingle（まとめるときにクラスター間でもっとも近いサンプル間の距離を採用する），average（平均の距離），weighted（平均値にクラスターのサンプル数の重みづけを加える）に変更するとこんな結果になった（図4-1，2，3）.

C君：だいぶ見た感じが違いますね．ちなみにscipy.cluster.hierarchy.linkageのデフォルトはmetric = 'euclidean', method= 'single' のようですね．でもこれが一番うまく分かれていないですね.

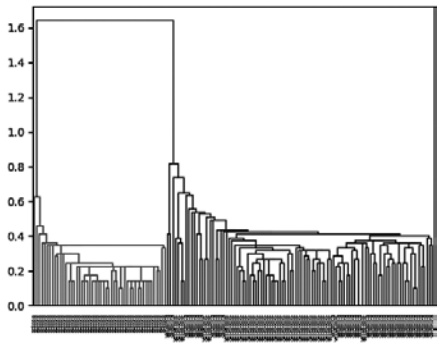

図4-1. metric = 'euclidean', method= 'single'

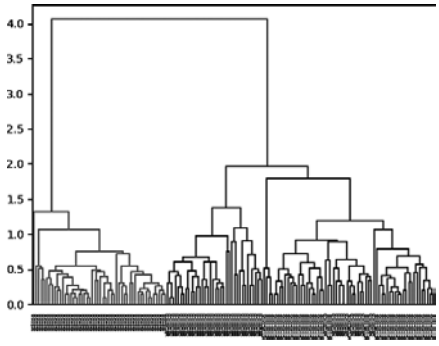

図4-2. metric = 'euclidean', method= 'average'

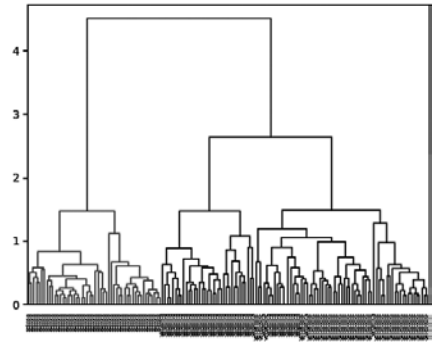

図4-3. metric = 'euclidean', method= 'weighted'

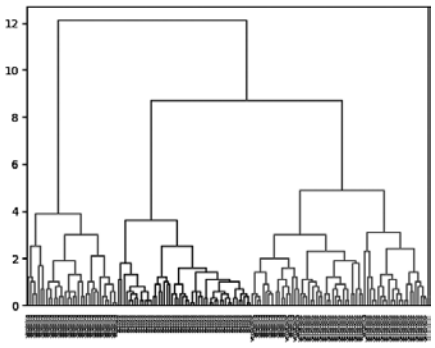

図5-1. metric = 'cityblock', method= 'complete'

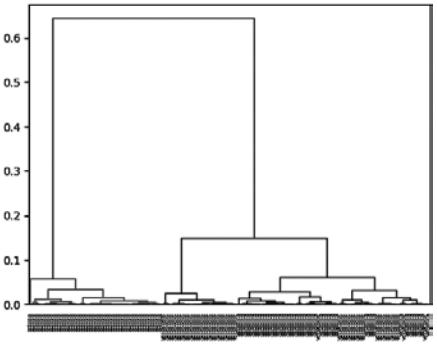

図5-2. metric = ' correlation ', method= 'complete'

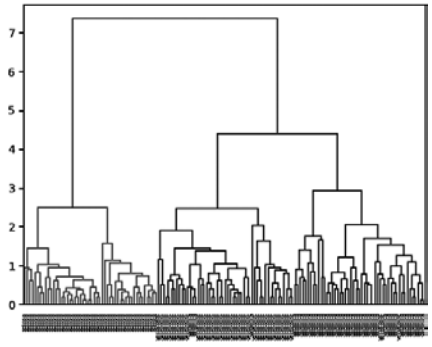

図6. metric = 'cityblock', method= 'weighted'

B君：つぎは結合方法をcompleteに固定して，距離関数を変えてみてよ．

C君：了解です．やはり見た感じがかわりますね．cityblockはマンハッタン距離とよばれるもので，correlationは相関係数を距離としたものですね（図5-1，2）．じゃあ，ちょっと調べてみましょうか（スクリプトを作成している）．

B君：お！全組合せを試してみるんだね．さすがC君！

C君：for文を使えば簡単ですよ．あ，出ました（図6）．metric = 'cityblock', method= 'weighted'にした時が，品種をもっとも反映したクラスターができてますね．

B君：ということで，この距離関数と結合方法の組合せが万能なのかなと思って，ネットや参考書の実施例を調べたけど，そうでもないみたいなんだわ．方法の違いで結果がころころ変わるんで，なんか気まぐれだな，どう取り扱ったらいいものやらって，悩んでいたところにC君がやってきたんだよね．

C君：先輩．やっぱり気まぐれって困りますよね．僕も相談したかったんです．X教授のところに相談に行きましょう．

総合的に判断する

B君：X教授！……というわけなんですが，どのように考えるといいんでしょうか？

X教授：今日は珍しい二人組だね．ふむふむ．階層クラスター分析の気まぐれにすっかり翻弄されたってわけだ．いろいろ振り回されて大変だね！青年諸君，“見た目を気にしない”“共通点を探す”“結論を急がない”のがコツだね．まず，デンドログラムのぱっと見た目を気にしすぎちゃだめだ．そもそも階層クラスター分析の目的ってなんだったっけ？

C君：似たサンプルを集めてクラスターを作り，分類するための手法です．

X教授：その通り．デンドログラムの分岐の右側と左側が，逆になっても分類結果は同じだ．だから横軸の並び方を気にせずよく見ると，どの方法でもsetosaは独立したクラスターになっている．

C君：ほんとだ．

X教授：問題はversicolorとverginiaのクラスターが，距離関数と結合方法によってばらつく点だよね．

B君：前回行った主成分分析のスコアプロットでは，versicolorの50サンプルとverginiaの50サンプルは一部重なり合うグループになっていましたっけ？

C君：ということは，「分類する」目的からすると，そもそもversicolorとverginiaを異なるクラスターに明確に分類できるわけではないってことですね．

X教授：私の経験的には，距離関数はユークリッド距離，結合法はcomplete法を基準に，いろんな方法を試してみて，総合的に判断して主張することを決めるしかないかなぁ．また，結果を示すときに距離関数と結合方法を必ず明記する，ことは忘れないようにね．あと，くれぐれも「予想していた結果と合致する」という理由で手法を選ばないように．では，今回の結果の解釈はどうなると思う？

C君：まず，setosaは独立したクラスターに分類できる．versicolorとverginiaは明確なクラスターに分類できない，でしょうか．

X教授：その通り．主成分分析と違って，階層クラスター分析結果からは，setosaと他のサンプルでは形態にどのような違いがあるのかは読み解くことができない．だから，クラスターを特定後，他の方法でsetosaと他のクラスターを比較して違いを調べる必要があるね．

C君：なるほど．つまり1番上の分岐には意味があるが，2つ目の分岐は再現性が乏しい，ということですね．では，どの分岐が信頼できるのか？というのはどうやって決めればいいんでしょうか？

X教授：これは結構むずかしい問題なんだ．似たような問題は分子系統樹の作成法でも起きたんだよ．

C君：遺伝子配列の相同性から分子進化の系統樹を書く手法の1つですね．2遺伝子の配列間の相同性を計算して，相同性が高いものからまとめて系統樹を作成する，という手法もまったく同じですね．

X教授：実際，用いるデータや相同性（距離に相当）の定義，結合方法に依存して大きく異なる分子進化系統樹が得られることがある．そこで，分子進化系統樹の信頼性を評価するブートストラップ法が開発されている．

B君：靴ひも法？ですか？

X教授：データの一部をランダムに選び（アライメント後の配列を復元抽出法でサンプリングし），系統樹を書くという作業を何度も繰り返す．ほとんどの試行で観察される分岐が信頼できる分岐になる．階層クラスター分析でこれを行うのはかなり上級レベルだね．要するにじっくり分析する相手を見て，結論を急ぎすぎないことが，大事なんだよ．頑張りたまえ青年！