**情報分析（第2回）　　　　　　　　　　　　　　　　　　　　　 2018年4月18日　辻慶太**

0. はじめに

今日はクラスタリングについて御紹介します。まずクラスター（Cluster）とは「グループ」とか「群」のことです。データマイニングや多変量解析で言うところのクラスタリングは「観測されたデータを，一定の意味ある体系に，自動的に組織立てること」です。

同じような性質のデータを1つのクラスタ（＝グループ）にまとめる（＝分類する）ことで，データが持つ特徴をあぶり出せる可能性が高まります。まとめられた結果から，これまで気付いていなかった体系を見つけることが出来るかもしれません。また同じクラスタに入れられたデータ同士に，これまで気付かれていなかった関係を見つけることが出来るかもしれません。

例えば，バラ，コスモス，チューリップ，つくし，シイタケ，ワカメをクラスタリングさせたところ，コンピュータはこれらを「バラ，コスモス，チューリップ」と「つくし，シイタケ，ワカメ」の2つに分けたとします。我々はその結果を眺めることで「前者にはきれいな花が咲く」などと気付くかもしれません。即ち，（大げさに言うと）植物は花の有無という観点で分けられることに気付くかもしれません。それはデータに内在する体系の発見とも言えます。

あるいはある商品（ノートPCとか）を評価した自由記述文を単語に切り分け，それら単語をクラスタリングしたとします。このとき肯定的な言葉（「良い」「気に入った」）と同じクラスタに入れられた言葉（「画質」「デザイン」）から購入者が何をポジティブに評価しているかが分かるかもしれません。逆に否定的な言葉（「悪い」「使いづらい」）と同じクラスタに入れられた言葉（「コネクタ」「テンキー」）から何の評判が悪いかが分かるかもしれません。人間が自由記述文を全て読んで記憶して考えれば気付けることかもしれませんが，クラスタリングを通じて少し楽に気付ける可能性があります。

ちなみにコンピュータ分野にも「クラスタリング」という用語があります。これは複数のコンピュータを相互に接続し，ユーザや他のコンピュータに対して全体で1台のコンピュータであるかのように振舞わせる技術です。Google で「クラスタリング」と検索すると，こちらに関するページが多数出て来るので注意しましょう。

さて次回以降に出て来るSVMや古典的な判別分析と，今回お話しするクラスタリングの違いは，前者には最初から分ける体系があるのに対して，今回のクラスタリングにはないことです。各データは被子植物か裸子植物のいずれかだと事前に分かっていて，各データがそのいずれに属するかを判定したいという場合はSVMや判別分析を用います。クラスタリングは用いません。あるいは図書や論文にNDCを自動付与したい（0～9類のいずれかに分類したい）というのも，最初から分けたい体系が存在するケースになります。この場合も同じくクラスタリングは用いません。クラスタリングは分けたい体系が事前にない場合に用いられます。従って発想のヒント的な使われ方が多いです。

以下ではクラスタリングを用いた研究例を概観した後，クラスタリングに適したデータの形についてお話しします。次にクラスタリングには階層的クラスタリングと非階層的クラスタリングの2種類があるので，それぞれについてお話しし，Rでの実行方法などを紹介します。

1. クラスタリングを用いた研究例

・上野覚ら (2016)「人の意味理解能力とテキストマイニングの大量データ処理技術とを融合した特許文献のハイブリッドクラスタリング手法」技術と経済.

・井ノ上直己ら (1991)「階層的クラスタリング手法の訳語選択への応用」情報処理学会研究報告自然言語処理.

・菊池望ら (2014)「Twitterにおける言及関係によるクラスタリングを利用したスパムアカウント判定手法の検討」研究報告情報システムと社会環境.

・花井俊介ら (2014)「スパムレシピ抽出のための酷似レシピクラスタリング手法」研究報告システムソフトウェアとオペレーティング・システム.

・張忠強ら (2014)「クラスタリングによるオンラインゲームにおける不正プレーヤー候補の検出」コンピュータセキュリティシンポジウム2014論文集.

・川瀬徹也ら (2013)「接客時間推定に向けた従業員の位置・音声データによる発話クラスタリング」研究報告音声言語情報処理.

・岸田和明 (2013)「小規模事例に基づく文書クラスタリング技法の実証比較 : 確率的モデルと非負行列分解とを中心に」TP&Dフォーラムシリーズ：整理技術・情報管理等研究論集.

・河原崎康友ら (2011)「顔形状特徴を用いた西洋宗教画のクラスタリング」映像情報メディア学会技術報告.

・齋藤良平ら (2011)「PC操作ログからの従業員PC利用パターンのクラスタリング」情報処理学会研究報告.

・井上雄介ら (2010)「Wikipedia記事中の意見文クラスタリング」情報処理学会研究報告. データベース・システム研究会報告.

・仁科朋也ら (2010)「単語グループに基づく Web 文書クラスタリング」自然言語処理.

・松村冬子ら (2009)「選好商品のクラスタリングに基づく嗜好の変化の検出」情報処理学会研究報告. MPS, 数理モデル化と問題解決研究報告.

・橋本泰一ら (2008)「社会課題発見のための文書クラスタリングとクラスタ評価指標」情報処理学会研究報告. 自然言語処理研究会報告.

・神田勇介ら (2008)「階層的重複クラスタリングによるウェブ検索結果の集約」電子情報通信学会技術研究報告.

・長谷川幹根ら (2007)「T-Scroll：時系列文書のクラスタリングに基づくトレンド可視化システム」情報処理学会論文誌.

・村上浩司ら (2007)「共起語に基づいた階層型文書クラスタリング手法」情報処理学会研究報告. 情報学基礎.

・平尾一樹ら (2006)「複合名詞に着目したWeb検索結果のクラスタリング」情報処理学会研究報告. 自然言語処理研究会報告.

・伊与田達也ら (2006)「テキスト特徴と画像特徴を併用したWeb上の写真付きニュース記事のクラスタリング」電子情報通信学会技術研究報告. PRMU.

2. 事物の性質をベクトルで表す（クラスタリングに適したデータの形）



ピタゴラス

「万物は数である」

→ 万物は数で表せる。少なくともそう考えて，なるべく

数で表そうとすると良い情報分析ができたりする（かも）。

さて有名な三国志のゲームでは各武将の能力を，例えば以下のように表したりします：



このとき張飛の能力は（50, 100, 60）という数値の並び（＝ベクトル）で表すことが出来ます。諸葛亮なら（100, 80, 90）です。

同様に，以下の『日本の図書館2010』のデータから言うと，例えば山形県立図書館の性質は（6085, 2, 580671, 18556, 88746, 197217）というベクトルで表すことが出来ます。このようにこの世の様々な事物の性質は，数値の並び（ベクトル）で表すことが出来ます。言うまでもないですが，数値はバラバラに並んでいてはダメで，取り上げると決めた項目の順（専有延床面積，専任司書，蔵書冊数...の順など）に並んでいないといけません：



「どんな性質も数値化できる」といっても例えば「県立」「町立」といった

性質は数値化できないのではないか？

→ 下のように0か1を付与することで数値化できます：



さて，では茨城県立図書館のベクトル（8700.69, 15, 819040, 17424, 121618, 771879, 1, 0, 0）は他のどの図書館のベクトルに近いでしょう？ 富山県立図書館の（7775.55, 21, 717147, 19869, 65687, 195174, 1, 0, 0）などが近そうです。少なくとも白岡町立図書館の（177, 2, 80527, 5612, 15252, 201594, 0, 0, 1）よりは近そうです。このようなベクトルの近さは後述する様々な式で算出することが出来ます。クラスタリングは，ベクトルが近いもの同士を1つにまとめる手法と言えます。

3. 階層的クラスタリング

では実際に上の図書館達を階層的クラスタリングでグループ分けしてみましょう。授業用HPから“libraries1.txt”をダウンロードし，Rのディレクトリをlibraries1.txtを落としたディレクトリに移動します。その上で次のように実行します：

> 図書館 <- read.table("libraries1.txt", header=TRUE, row.names=1, sep="\t")

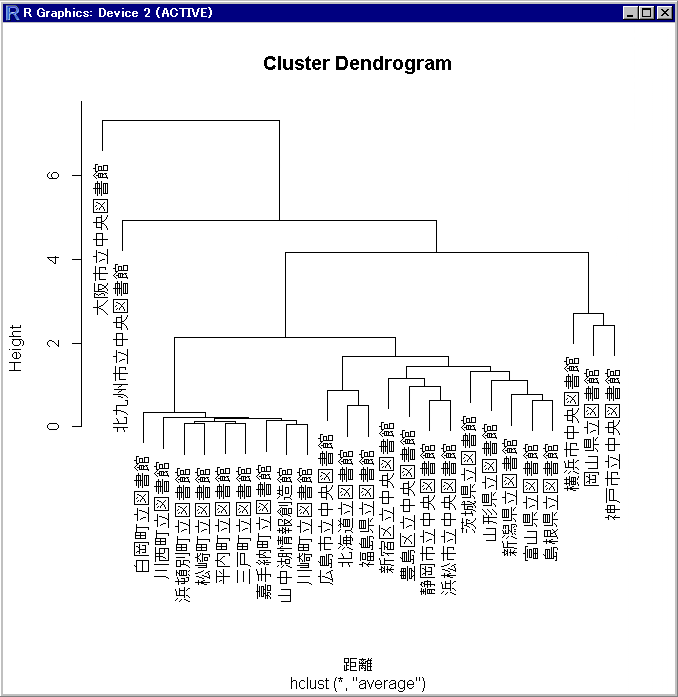
> 図書館 <- scale(図書館)

> 距離 <- dist(図書館, method="euclidean")

> 結果 <- hclust(距離, method="average")

> plot(結果)

すると次のような結果が得られます：

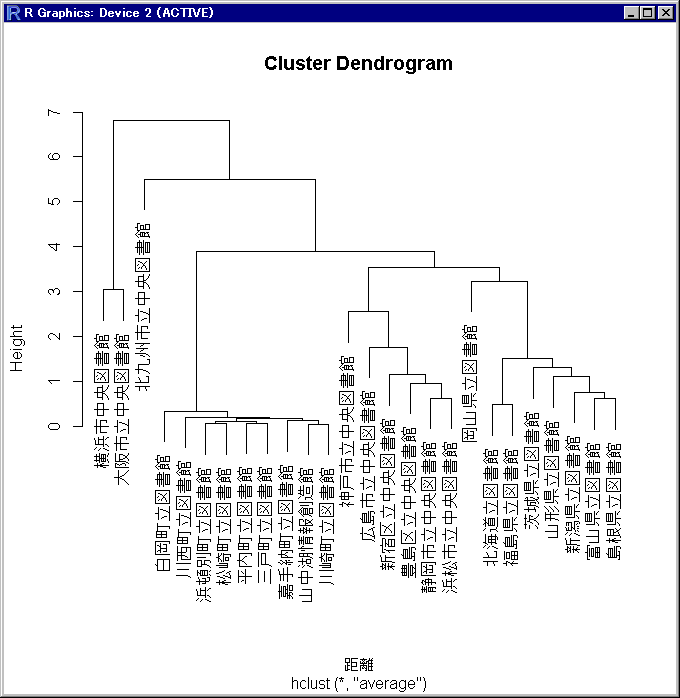


トーナメント式のような線で結ばれた図書館が「近い」図書館と判断されたものです。例えば真ん中に示されているように北海道立図書館と福島県立図書館は近いと判断されています。町立図書館は主に真ん中左に集められて1つのクラスタを作っていることが分かります。一方，県立図書館と市区立図書館はあまりきれいに分かれませんでした。これは市区立図書館の中でも今回は中央館だけを集めたため，県立図書館と規模や性質が変わらない図書館がサンプルになってしまったためと考えられます。例えば横浜市中央図書館は岡山県立図書館と規模などの面で変わらないため，一番右の同じクラスタに入れられています。

「同じクラスタとは？」と思われたかもしれません。階層的クラスタリングでは，上記のような結果を横線で切った時，それより下に固められているのがクラスタです。ですから横線をどの高さに取るかで，クラスタの総数や各クラスタ内のメンバーの数は違ってきます。例えば一番上の部分に横線を引いたらデータ全体は「大阪市立中央図書館」というクラスタと「残り全部」というクラスタの2つに分かれます。上から3番目の部分に横線を入れたら「大阪市立中央図書館」「北九州市立中央図書館」「横浜市中央図書館，岡山県立図書館，神戸市立中央図書館」「残り全部」という4つのクラスタに分かれます。階層的クラスタリングの良い所は，結果を眺めてどこに横線を引くとおもしろいか（きれいか），人間が考えられる点にあります。後述する非階層的クラスタリングにはそのような利点はありません。名前の通り「非階層的」なので上のようなトーナメント形式では出力されず，1つのクラスタ内におけるサブクラスタ（さらに小さいクラスタ）などは分かりませんし，あるクラスタを含む上位クラスタなども分かりません。

最後に，上の図を細かく見ると大阪市立と北九州市立がトーナメントの一番左に置かれ，「他の図書館とは違う」という位置付けになっています。これは両図書館の貸出数（と登録者数）が全体の1位2位で，他の図書館よりかなり多くなっていることが影響しているものと思われます。

さて県立，市区立，町立を0か1で表したバージョンもlibraries2.txtとして授業用HPに置いてみました。こちらを使った場合の結果は以下のようになりました。横浜市，大阪市，北九州市が左側で特別扱いされていますが，それ以外は町立，市区立，県立の3つにきれいに分かれました。



先ほどのコマンドについて簡単に説明してきます。

> 図書館 <- read.table("libraries1.txt", header=TRUE, row.names=1, sep="\t")

→ libraries1.txtというファイルを読み込んで「図書館」というオブジェクトに入れろと言っています。header=TRUEは1行目が変数名（専有延床面積，専任司書，蔵書冊数など）になっていることをRに伝えています。row.names=1は1列目がレコード名（北海道立図書館，山形県立図書館など）になっていることをRに伝えています。即ち，これらはデータではないからクラスタリングの計算には使うなとRに伝えています。

> 図書館 <- scale(図書館)

→ これはとても大事なポイントなので覚えておいて下さい。これは図書館における各変数の値を平均0，分散1に**標準化**しろと言っています。即ち，どの変数の値も互いの大小関係は維持しながらせいぜい－3から3くらいに収まるように変換しろ（3シグマ（σ）の法則：σ2が分散でσが標準偏差），と言っています。このように変換させておかないと，例えば蔵書冊数（数十万）は専任司書（数十）よりはるかに大きな数ですから，蔵書冊数が専任司書よりもずっと大きな影響をクラスタリングに及ぼしてしまいます。逆に言うと専任司書の値はほとんど無視されることになってしまいます。なのでそういう事態を避け，どの変数も平等に影響力を持てるように標準化を行うのです。具体的には，ある変数の値をxi（1≦i≦N），その平均をm，標準偏差をsとすると，各値はzi＝(xi－m)/s と変換します。こうすることでziは平均0，分散1になります。気になったら手計算で確かめてみて下さい。ちなみにこの標準化を行わないでクラスタリングを行った場合は，libraries1.txtでもlibraries2.txtでも全く変わらず結果は次ページの図のようになります。即ち「県立」「市区立」「町村立」の1か0かという，蔵書冊数などに比べればゴミのような値は，結果に全く影響を与えなくなります。

> 距離 <- dist(図書館, method="euclidean")

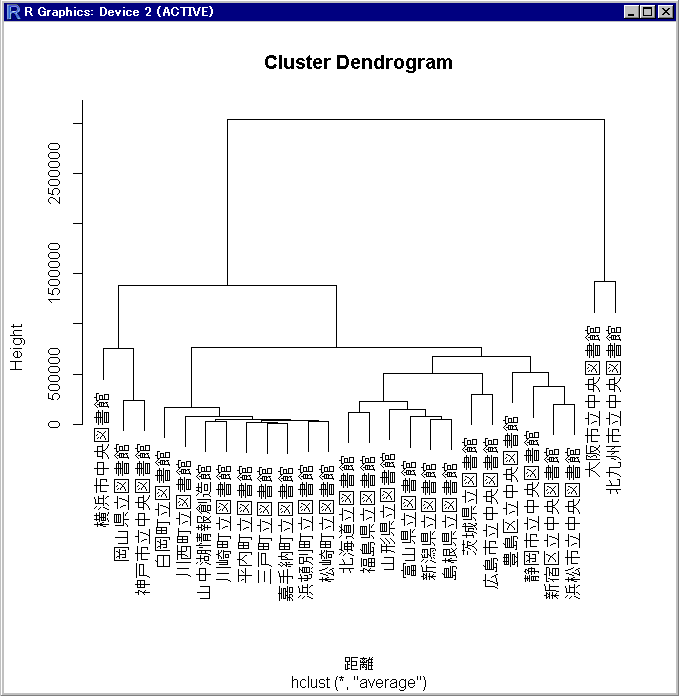
→ 図書館オブジェクト内のレコード同士の距離を，ユークリッド距離で算出しています。

> 結果 <- hclust(距離, method="average")

→ 上で得た距離を元に，後述する群平均法（average）でクラスタリングを行っています。averageの他にward，single，complete，mcquittyなどが選べます。

> plot(結果)

→ 結果を描画します。



図：標準化を行わなかった場合の結果

■階層的クラスタリングのアルゴリズム（流れ）

n個の個体を順番に結合させ，1つのクラスタになるまで繰り返します：

(1) n個の個体について，はじめはすべてバラバラの状態，つまり1個ずつの個体が入っているn個のクラスタを作ります。

(2) 次に，最も近い，すなわち距離が最小である2つの個体を結合させて新たなクラスタとします。これによりクラスタの総数はn－1個になります。

(3) さらにn－1個のクラスタの中から，最も近い2つのクラスタを結合させて新たなクラスタとします。これによりクラスタの総数はn－2個になります。

(4) 同様に，繰り返し結合させていき，最後に1つのクラスタになるまで実行します。

このアルゴリズムによって各個体は順々に結合されていきます。一度結合された個体が後のステップで分離されることはありません。ここが後述する非階層的クラスタリングとの違いになります（非階層的クラスタリングでは一度1つのクラスタにまとめられても後で離されたりします。集合離散を繰り返します）。階層的クラスタリングでは各クラスタがいつどのクラスタの結合によって生まれたかが明確なので，クラスタ同士に上下関係を認めることができ，最終的な結果も階層的なものとなります（トーナメントのような見た目になります）。

※上記のように，最も近い個体やクラスタを1つの新しいクラスタにまとめ，順次この作業を繰り返していく手法を「凝縮型」と言います。その逆に，最初にすべてを1つのクラスタとし，それを徐々に小さいクラスタに分けていく手法を「分岐型」と言います。が，凝縮型の方が一般的です。

■個体間の距離（ベクトル同士の距離）

階層的，非階層的を問わず，クラスタリングでは，まずすべての個体間の距離を算出します。個体をx=(x1,x2,...,xn)，y=(y1,y2,...,yn) とした時，以下に代表的な距離の定義を紹介します。

xが例えば先ほどの張飛であった場合はx＝(50, 100, 60)，yが諸葛亮であった場合はy=(100, 80, 90) となります。しつこいですが，xが山形県立図書館だったらx=(6085, 2, 580671, 18556...)，yが福島県立図書館だったらy=(9017, 21, 841292, 40766...) となります。

では以下では簡単に手計算できるよう，太郎，次郎，花子の3人を例として取り上げます。彼らに，肉，魚，ケーキ，のそれぞれをどれくらい好きかを5段階で尋ねたところ，太郎は肉5，魚4，ケーキ1と答えたとします。また次郎は肉4，魚4，ケーキ2と答え，花子は肉1，魚3，ケーキ5と答えたとします。このとき3人の好みはベクトルで，太郎=(5, 4, 1)，次郎=(4, 4, 2)，花子=(1, 3, 5)のように表せます。彼らの間の距離として以下の距離を練習で算出してみましょう。

(1) ユークリッド距離：多次元空間の単なる幾何的な距離。

距離

　　　　　　　　　　　　　　　　距離(太郎，次郎)=（　　　　　　　　　　　　　　　）

　　　　　　　　　　　　　　　　距離(太郎，花子)=（　　　　　　　　　　　　　　　）

(2) ユークリッド距離の２乗：より離れている個体に大きな重み（離れている度）を与えたい場合の距離。

距離

　　　　　　　　　　　　　　　　距離(太郎，次郎)=（　　　　　　　　　　　　　　　）

　　　　　　　　　　　　　　　　距離(太郎，花子)=（　　　　　　　　　　　　　　　）

(3) 市街地（マンハッタン）距離：単純な各次元の差の絶対値の和で表された距離。

距離

　　　　　　　　　　　　　　　　距離(太郎，次郎)=（　　　　　　　　　　　　　　　）

　　　　　　　　　　　　　　　　距離(太郎，花子)=（　　　　　　　　　　　　　　　）

(4) チェビシェフの距離：全次元の中で最も大きく異なる次元の値を距離としたい場合の距離。

距離

　　　　　　　　　　　　　　　　距離(太郎，次郎)=（　　　　　　　　　　　　　　　）

　　　　　　　　　　　　　　　　距離(太郎，花子)=（　　　　　　　　　　　　　　　）

(5) べき乗距離（ミンコフスキー距離）：に関して様々な重み付けを行いたい場合の距離。先ほどのユークリッド距離は，と言えます。パラメータは個々の次元の違いに与える重みを調整し，パラメータは個体間の差に与える重みを調整します。

距離

■クラスタ間の距離

　上記は2つの個体間の距離（ベクトル同士の距離）の話でした。以下ではいくつかの個体から成るクラスタ同士の距離についてお話しします。

(1) 最近隣法（最短距離法）

それぞれのクラスタ内の最も近い個体間の距離を用います。

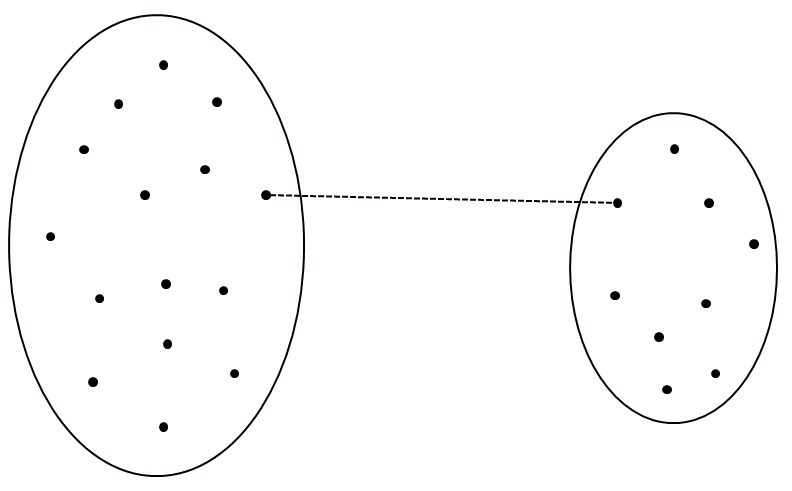
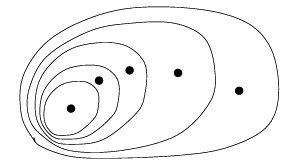


図7：最短距離法（最近隣法）

結果として得られるクラスタは，以下のように一方向に長い“チェーン状”になりやすいです。逆に言うとチェーン状になるべきではない時にはこの最近隣法は避けた方が良いです。



> 結果 <- hclust(距離, method="single")　　　※使う時は左のように single を指定します。

(2) 最遠隣法（最長距離法）

それぞれのクラスタ内の最も遠い個体間の距離を用います。

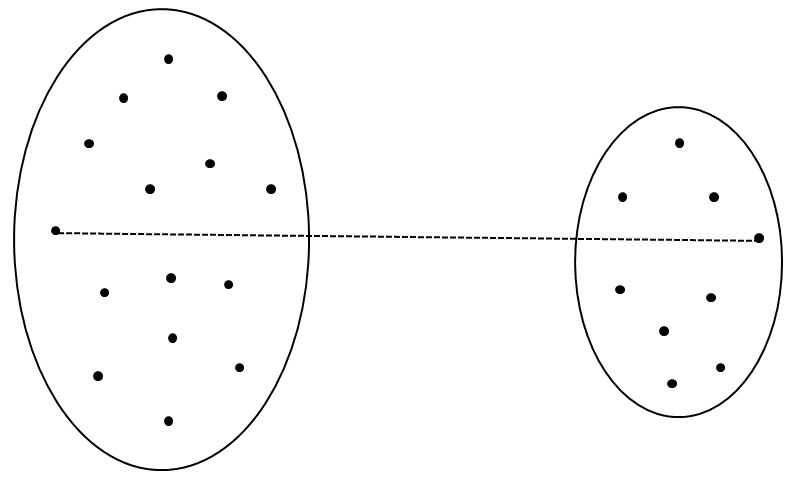


図8：最長距離法（最遠隣法）

この方法は，個体がはっきりとした“集団”を形成している場合には，非常によく機能します。が，先述の“チェーン状”になるべき時には適切でありません。

> 結果 <- hclust(距離, method="complete")　　　※使う時は complete を指定します。

(3) 群平均法

UPGMA (Unweighted Pair Group Method using Arithmetic average) ともいいます。それぞれのクラスタ内のすべての個体の組み合わせを考え，その個体間の距離の平均を用います。

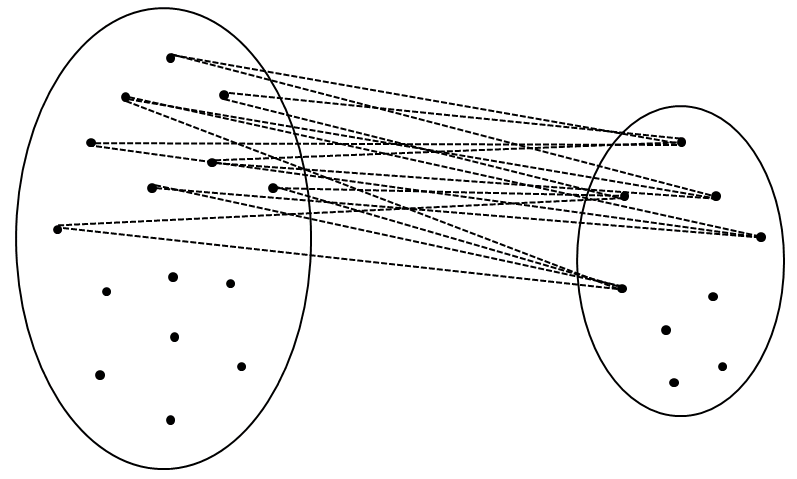


図9：群平均法

この方法も，個体がはっきりとした“集団”を形成している場合には非常に有効です。またチェーン状になるべき時にも有効です。使う時は先述のようにmethod="average" とします。

(4) ウォード法

他の方法とはかなり異なります。各ステップで形成される任意の2つのクラスタにおけるグループ内平方和を最小にするようにします。使う時はmethod="ward" とします。

■練習

(1) BIBLISから抽出した共著データファイル“BIBLIS\_1989\_2004\_200papers\_+2authors.dat”を授業用HPからダウンロードし，ダウンロードした先のディレクトリにRのカレントディレクトリを移動して以下のように実行してみて下さい：

> 共著 <- read.table("BIBLIS\_1989\_2004\_200papers\_+2authors.dat", header=TRUE, row.names=1, sep="\t")

> 共著 <- scale(共著)

> 距離 <- dist(共著, method="euclidean")

> 結果 <- hclust(距離, method="average")

> plot(結果)

するとこのファイルに示されている共著関係から著者をいくつかのクラスタ（＝グループ）に分けて表示してくれます。長尾真先生と原田勝先生が同じクラスタに入れられていますが：

論文(1)の著者： 谷口敏夫，澤田芳郎，吉田哲三，柿元俊博，長尾真，原田勝，石川徹也

論文(2)の著者： 原田勝，石川徹也，谷口敏夫，澤田芳郎，吉田哲三，柿元俊博，長尾真

→ 論文(1)でも(2)でも，長尾先生は石川徹也，谷口敏夫...と共著になっています。同様に原田先生も石川徹也，谷口敏夫...と共著になっています。従って長尾先生と原田先生は共著に関する性質のベクトルが似ていると言えるので，同じクラスタに入れられたと思われます。

3. 関連するWebページをクラスタリングでまとめてみる

サーチエンジンで何かを検索した時，同じようなページがランダムに出て来てうんざりすることがあります。同じようなページ，関連するページは１つにまとめて表示されたら便利ではないでしょうか？ 例えば Google で「慶太」と検索すると（以前は）図1のようになりました。雨宮慶太，斉藤慶太，五島慶太ごとにディレクトリが表示され，そこから目当ての人のディレクトリに入っていけば求める情報を効率よく集められるかもしれません。今日のクラスタリングはそれにうってつけの手法です。クラスタリングは，類似したアイテムや関連したアイテムを1つにまとめる方法です。

さて図1の検索結果には，よく見るとページごとに小さな要約（スニペットとも言います。どのように作っているかはhttps://www.suzukikenichi.com/blog/about-snippet-of-google-yahoo-bing/ などを参照して下さい。人が来てくれそうなスニペットになるようHP作成者は色々工夫したりします）が表示されています。例えば4番目のページには「特撮映画、日本特撮界の鬼才雨宮慶太、雨宮慶太オフィシャルファンサイト...」といった要約が示されています。とりあえず，この要約部分に使われている語をもとに「慶太」の検索結果上位10ページをクラスタリングしてみましょう。

まず形態素解析ソフト茶筌（<http://chasen-legacy.sourceforge.jp/>）で，要約部分を語に切り分けます。Windows 版を用いるなら，例えば図2のように切り分けることができます。茶筌で語を切り分けると，先ほどの要約部分に現れていた語は図3のようになりました（ここで「ひらがなや記号だけの語」及び「慶太」は除外しました。前者は，あまり意味がないのに頻度だけは高く，結果に悪影響を及ぼすからです）。

図4には，検索結果ページ1番目の「クラウド・雨宮慶太オフィシャルサイト」に現れている語に頻度を付与して示してみました。同様に，図5には検索結果ページ4番目の「旧雨宮慶太ファンサイト」に現れている語の頻度を付与してみました。図4図5から，これら雨宮慶太のページには「雨宮」の他に「特撮」といった語が高い頻度で現れていることが分かります。逆にこれらのページには，辻慶太のページに出てくる「図書館」といった語は現れていないことも分かります。こうした語の出方のパターン（頻度分布と言います）に基づいて，これら10個のページをクラスタリングすることが出来ます。



図1：Googleで「慶太」で検索した結果

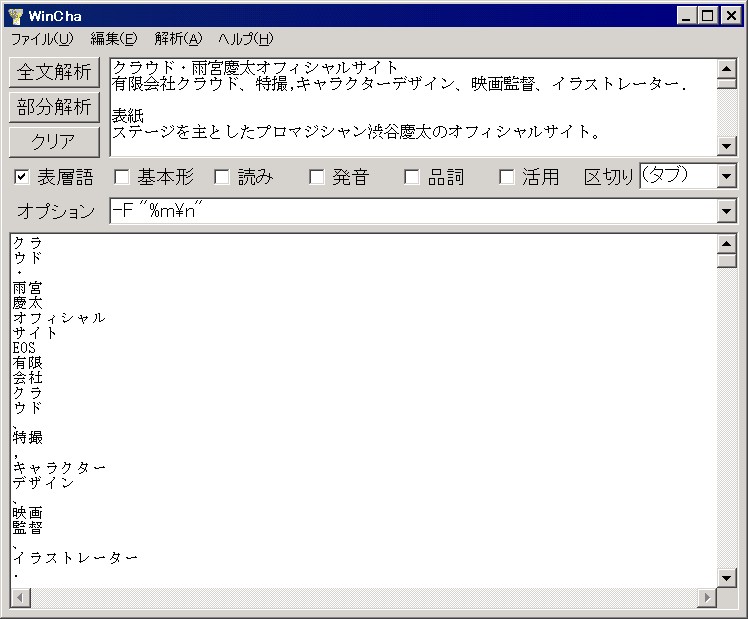


図2：茶筌の形態素解析結果

検索結果上位10ページは，181個の語の頻度情報データで，下図のように表現できます。これは先ほどの三国志の「知力」「武力」「魅力」が「アドレス」「イラストレーター」「ウド」といった言葉に変わっただけで，各ページの性質をベクトルで表していることに変わりはありません：

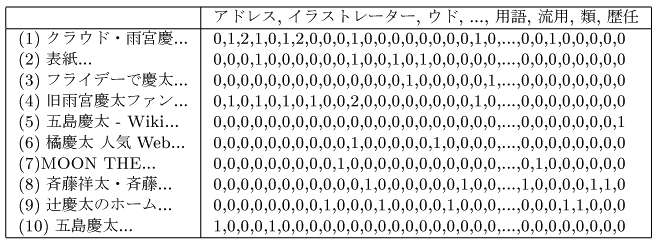


図3：検索結果10ページ中の各語の頻度（図4，5も参照のこと）

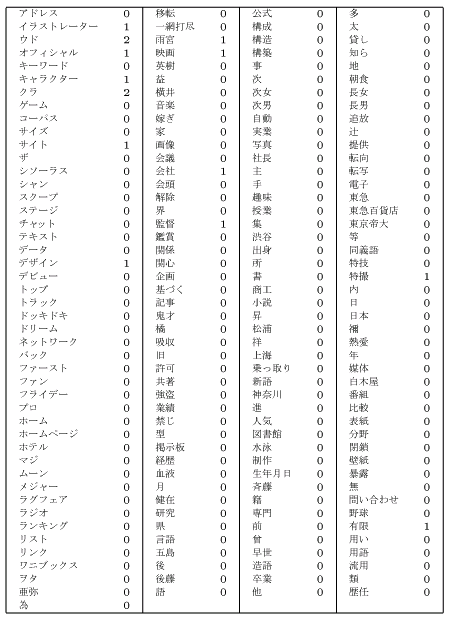


図4：「クラウド・雨宮慶太オフィシャルサイト」の語の頻度分布

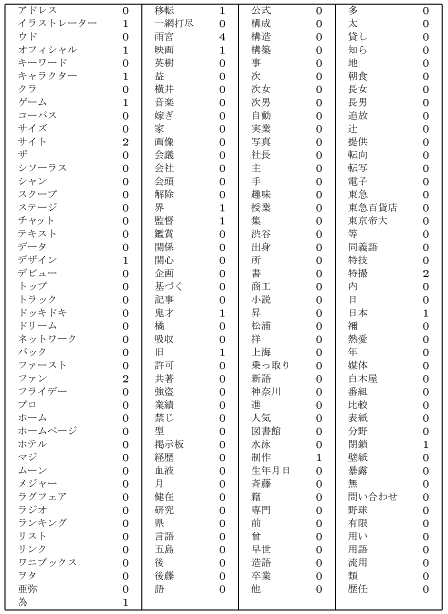


　図5：「旧雨宮慶太ファンサイト」の語の頻度分布

図3のデータは<http://slis.sakura.ne.jp/jb/keita.txt> に置いてみました。これをRにかけ，k-means法でクラスタリングしてみましょう。上記ファイルがあるディレクトリに移動し，以下のように入力します：

> 慶太 <- read.table("keita.txt", header=FALSE, row.names=1, sep="\t")

> kmeans(慶太, 5)

すると運が良ければ図6のような結果が得られます（後述しますが，クラスタの数は5に設定しました）。表6の①と④の雨宮慶太は「1」という1つのクラスタにまとめられ，⑦⑧の斉藤慶太は「3」という1つのクラスタにまとめられたことが分かります。また⑨の辻慶太はどこかに混ぜられることなく単独で「5」という1つのクラスタとなりました。もっとも渋谷慶太，後藤慶太と⑤の橘慶太は「4」というクラスタにいっしょくたにされてしまいました。その点ではうまく行っていません。

実は上の結果は “kmeans(慶太, 5)”を4，5回実行して比較的良かった結果を引っ張ってきたものです。k-means法は常に同じ結果を出すとは限りません。実行するたびに激しく変わることもデータによってはあります。この点は注意が必要で，後述します。

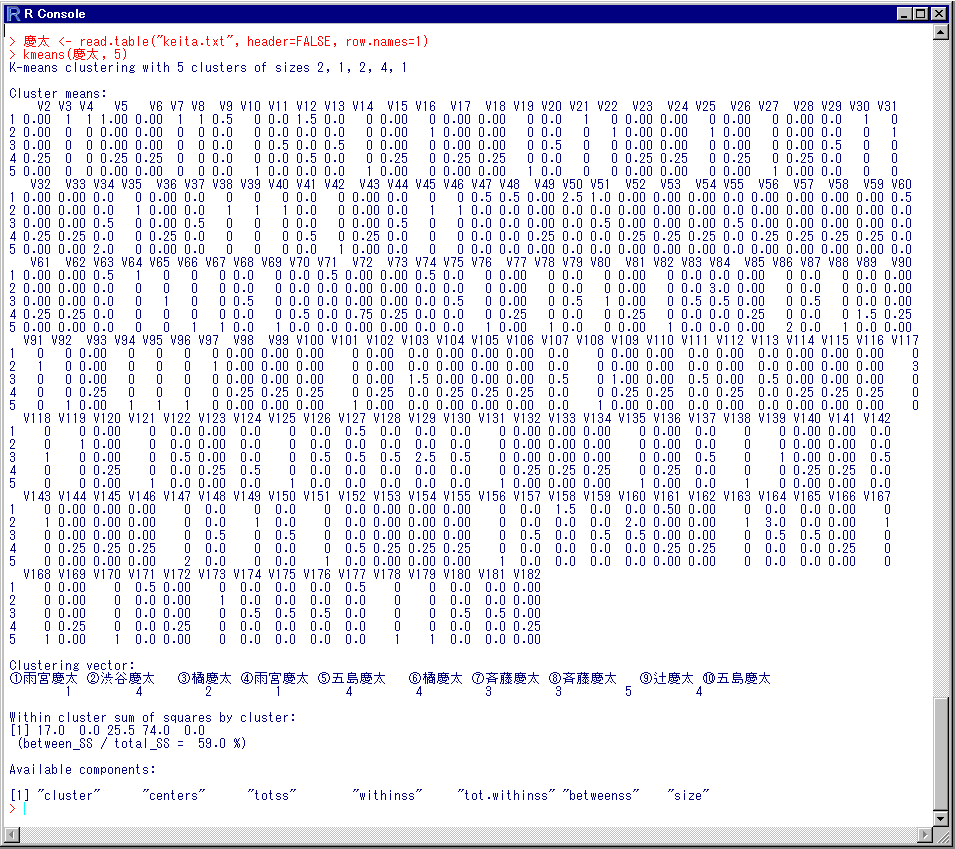
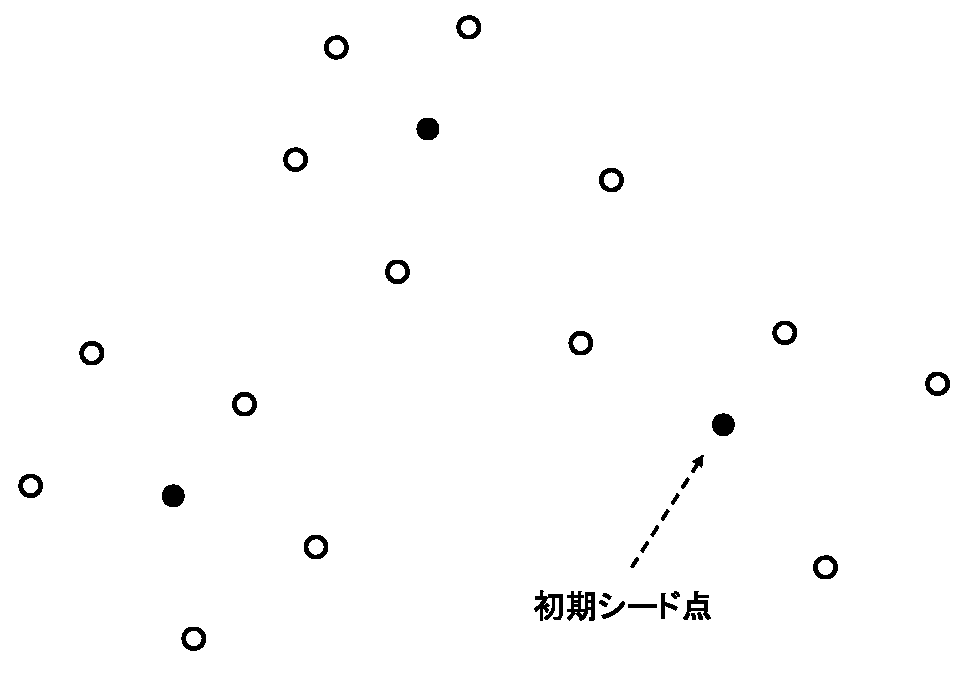


図6：k-means法によるクラスタリング結果（割と運が良かった場合の結果）

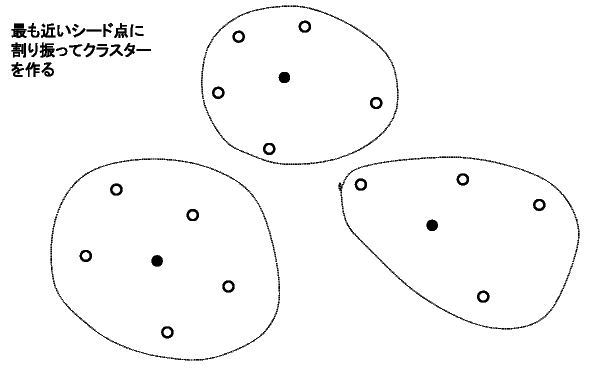
■k-means法のアルゴリズム

(1) 最終的にほしいクラスタの数を指定します。

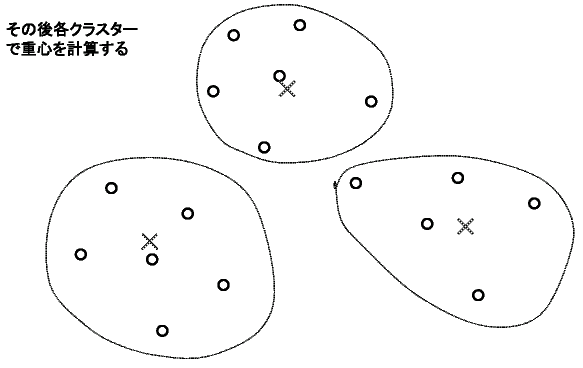
(2) 指定されたクラスタの数だけ，個体を選び，「シード点」とします。これを出発点にするのです（シードはseedで「種」）。シード点にはなるべく互いに離れている個体を選びます。具体的な選び方の1つとしては，まず何らかの個体を1つ目のシード点とします。次にそれから最も離れた個体を2つ目のシード点とします。次にこれら2つのシード点から最も離れた個体を3つ目のシード点とします（最遠点アルゴリズ：farthest point algorithm）（以下の図では互いに最も遠い点にはなっていません。すみません）。



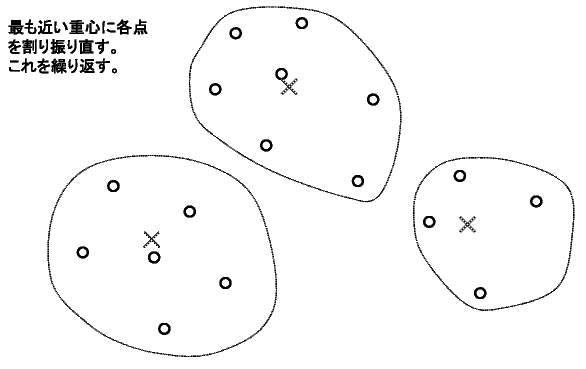
(3) 各個体とすべてのシード点との距離を計算し，各個体を最も近いシード点のクラスタに入れます。



(4) 各クラスタの重心（平均）を求め，新たなシード点とします。



(5) 新たなシード点と全個体との距離を計算し，各個体を最も近いシード点のクラスタに入れます。



(6) 上記を繰り返し，クラスタに変化が起こらないか，または繰り返し回数の上限に達した場合に終了し，最終結果を出力します。

以上のアルゴリズムにより，クラスタ内におけるデータ点のちらぼりが小さくなるクラスタを求めることが出来ます。

■階層的クラスタリングの長短所

・階層的クラスタリングは，個体数が多いと計算量が膨大になります。

・一方，階層的クラスタリングは，先述のように樹形図に横線を引くことで，最も良さそうなクラスタを見出すことが出来ます。

■非階層的クラスタリング（k-means 法）の長短所

・k-means法は初期値に依存します。即ち，どの個体を最初のシード点にするかで最終結果が全然違ってきます。

・最終的にほしいクラスタの数を人間が指定しないといけません。何個が良いのか分からないうちは何度も試すことになります。

・上と重なりますが，指定した数のクラスタしか得られず，階層的クラスタリングにおける横線のような技は使えないので，クラスタ数を指定し直して何度もクラスタリングを行うことになります。あるいは得られたクラスタ内のサブクラスタ（より小さいクラスタ）がどれなのか分からないと言うことも出来ます。

・得られるクラスタは局所最適解です。即ち，本当はもっと良い分け方があったかもしれないのに，そこに辿り着けるシード点を最初に選んでいなかったら辿り着けません。

・外れ値に敏感です。

・一方，k-means法は，個体数が多くても計算量が抑えられるため，大規模データに適しています。

■練習

(1) BIBLISから抽出した共著データファイル“BIBLIS\_1989\_2004\_200papers\_+2authors.dat”を授業用HPからダウンロードし，ダウンロードした先のディレクトリにRのカレントディレクトリを移動して以下のように実行してみて下さい：

> 共著 <- read.table("BIBLIS\_1989\_2004\_200papers\_+2authors.dat", header=TRUE, row.names=1, sep="\t")

> kmeans(共著, 5)

すると先ほど階層的クラスタリングで行ったように，著者がいくつかのクラスターに分類されます（上ではクラスター数を5に指定しているので5つのクラスターに分類されます）。長尾真先生と原田勝先生はやはり同じクラスターに分類されている（可能性が高い）と思います。

(2) 食品に関するデータファイル“shokuhin.csv”に挙げられている食品をk-means法でクラスタリングしてみましょう。“shokuhin.csv”をエディタなどで開くと分かりますが，こちらは先ほどのlibraries1.txtなどと違って各データが“,”で区切られています。そのため，読み込みでは以下のように sep=”,” というオプションが必要になります。覚えておいて下さい（libraries1.txtのようなタブ区切りのデータではsep="\t" になります）。

> 食品 <- read.table("shokuhin.csv", header=TRUE, row.names=1, sep=",")

■補足

(1) 全ての分類対象が：ちょうど一つだけのクラスタの要素となる場合：ハード，または，クリスプなクラスタと言います。

(2) 一つのクラスタが複数のクラスタに同時に部分的に所属する場合，ソフト，またはファジィなクラスタと言います。

(3) 非階層的手法には，k-means法の他にISODATA法などがあります。

(4) 他に次元の呪いといったクラスタリングの様々な問題点などについては：

<http://www.kamishima.net/jp/clustering/>