教師なし学習（クラスタリング）演習ガイド

タスク

Iris Species Data Set （http://archive.ics.uci.edu/ml/datasets/iris） は 3 種類の Iris（アヤメ）各 50 個体について、Sepal（萼、がく）の長さと幅、Petal（花弁）の長さと幅を計測した結果をまとめたデータセットである。R には組み込みデータセットとして用意されており、str(iris)でそのデータ構造を確認することができる。1 行が 5 つの変数からなり、1 個体ごとに Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, Species が並んでいる。最後の Species は Iris の種類のことであり、Iris setosa, Iris versicolor, Iris virginica のいずれかである。

萼や花弁の計測値からアヤメの種類を判別するタスクの場合は教師あり学習となる。しかし、ここでは、アヤメが 3 種類に分類されるという前提知識が無いものとして、萼や花弁の計測値だけからクラスタリングによっていくつかのグループに分類できないか、という問題を考えるものとする。これは教師なし学習のクラスタリングのタスクである。この演習では、アルゴリズムとして k 平均法を利用したモデル構築を行う。

k 平均法を用いたモデル構築

冒頭でも述べた通り、なすべきタスクは、Iris Species Data Set を利用して、萼や花弁の計測値だけからクラスタリングによっていくつかのグループに分類する機械学習モデルを構築することであり、教師なし学習のクラスタリングの問題となる。本章では、アルゴリズムとして k 平均法を利用したモデル構築を行う。

データの準備

新規プロジェクト（ディレクトリ名は iris とする）を立ち上げ、新規スクリプトを開いておく。Iris Species Data Set は、R には組み込みデータセットとしてあらかじめ用意されている。

str(iris)でそのデータ構造を確認することができる。

```
> str(iris)
'data.frame':  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

5 つの変数を持つ 150 個体のデータからなるデータフレームである。、1 個体ごとに Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, Species が並んでいる。Species は 3 水準の因子ベクトルであり、水準はそれぞれ setosa, virginica, versicolor である。1 種類につき 50 個体からなることが table(iris$Species)により確かめられる。

```
> table(iris$Species)
    setosa versicolor  virginica
        50         50         50
```

ただし、萼と花弁の長さと幅といった計測値だけからグループに分けるクラスタリングを行うことが目的である。このため、Sepcies（5 列目）は分けておく。

```
> iris_data <- iris[,-5]
> iris_class <- iris[,5]
> str(iris_class)
 Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> str(iris_data)
'data.frame':  150 obs. of  4 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

データの範囲を把握するため、基本統計量を summary()関数で求める。

```
> summary(iris_data)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

取り得る値の範囲が特徴量によって多少異なる。ただし、特徴量の単位はすべて cm であり、取り得る値の桁も変わらないことからここでは正規化・標準化等の前処理を行わない。ただし、前処理を行う場合は次の節に従う。

k 平均法のための前処理

k 平均法では特徴量空間における距離に基づいてクラスタリングを行う。これは k 近傍法でも考慮した事項であるが、距離の計算に関して、特徴量の間で取り得る値の範囲が大きく異なると望ましくない。今のデータの場合どの特徴量も取りうる値の桁はそれほど変わらないが、それでも取り得る値の範囲は多少異なる。これをふまえ、どの特徴量に関しても取り得る値の範囲が同等になるように正規化・標準化を行った方が良い場合があるだろう。

ここでは各特徴量について平均を 0、標準偏差を 1 にする標準化（Z スコア標準化）を行う例を示す。これには R 組み込みの scale()関数を用いる。scale()関数の戻り値はリストとなるため、as.data.frame()関数によりリストからデータフレームへと変換する。

```
> iris_data_z <- as.data.frame(scale(iris_data))
```

```
> summary(iris_data_z)
 Sepal.Length      Sepal.Width       Petal.Length      Petal.Width
Min.   :-1.86378   Min.   :-2.4258   Min.   :-1.5623   Min.   :-1.4422
1st Qu.:-0.89767   1st Qu.:-0.5904   1st Qu.:-1.2225   1st Qu.:-1.1799
Median :-0.05233   Median :-0.1315   Median : 0.3354   Median : 0.1321
Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
3rd Qu.: 0.67225   3rd Qu.: 0.5567   3rd Qu.: 0.7602   3rd Qu.: 0.7880
Max.   : 2.48370   Max.   : 3.0805   Max.   : 1.7799   Max.   : 1.7064
```

どの特徴量も平均が 0 となり、値の範囲がおおむねそろっていることが確認できる。Z 標準化のため、最大値や最小値は特徴量によって異なるが、どの特徴量も標準偏差は 1 となっている。

```
> lapply(iris_data_z, sd)
$Sepal.Length
[1] 1
$Sepal.Width
[1] 1
$Petal.Length
[1] 1
$Petal.Width
[1] 1
```

ただし、前節でも述べた通り、元のデータであっても、取り得る値の桁は変わらず、単位も同じ cm であることから、今回は標準化を施さない元のデータ iris_data でクラスタリングを行う。なお、標準化したデータ iris_data_z であっても以降の手順は同様である。

モデルの訓練

k 平均法（k means method）の実装 kmeans()を用いる。kmeans()は stas パッケージに含まれており、stats パッケージ自体 R インストール時にデフォルトで含まれている。見当たらない場合は install.packages(“stats”)して、library(stats)すればよい。

stats パッケージの kmeans()関数は少なくとも 2 つの引数の指定が必要である。構文は以下のとおりである。

myclusters <- kmeans(mydata, k)

- ✓ mydata クラスタリングされるインスタンスを格納する行列かデータフレーム
- ✓ k 作成したいクラスタの数

戻り値は、クラスタについての情報が格納されたオブジェクトである。次のようにしてクラスタの解析を行うことができる。

myclusters$cluster: 各インスタンスが所属するクラスタ番号の並んだベクトル
myclusters$centers: 各クラスタのクラスタ中心の行列
myclusters$size: 各クラスタに割り振られたインスタンスの数

データからいくつのクラスタを作るかは分析者が決定しなければならない。k の値はいろいろ試せばよいのであるが、自然なグループの本当の数についての手がかりがあればそれを用いるのがよい。今回はこの点には深入りせず、k=3 と決めたときに、実際のグループに近いクラスタリングができるかどうか検証する。

```
> iris_clusters <- kmeans(iris_data, 3)
```

モデルの性能評価

モデルの性能評価のため、クラスタについての情報が格納されたオブジェクト iris_clusters を表示する。

```
> iris_clusters
K-means clustering with 3 clusters of sizes 50, 38, 62
```

3 つのクラスタはそれぞれ 50, 38, 62 個体が含まれるような分け方となっている。

```
Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     6.850000    3.073684     5.742105    2.071053
3     5.901613    2.748387     4.393548    1.433871
```

クラスタ 1 は Petal が極端に小さいグループのようである。クラスタ 2 とクラスタ 3 は比較的似ているが、クラスタ 2 の方が大型のようである。

```
Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [45] 1 1 1 1 1 1 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3
 [89] 3 3 3 3 3 3 3 3 3 3 3 3 2 3 2 2 2 2 3 2 2 2 2 2 3 3 2 2 2 2 3 2 3 2 3 2 2 3 3 2 2 2 2
[133] 2 3 2 2 2 2 3 2 2 2 3 2 2 2 3 2 2 3
```

このデータでは 150 個体のうち最初の 50 個体がクラスタ 1 に分類され、51 番目からの多くはクラスタ 3 に、最後の方はクラスタ 2 に分類されているようである。

```
Within cluster sum of squares by cluster:
[1] 15.15100 23.87947 39.82097
 (between_SS / total_SS =  88.4 %)
```

クラスタリングはクラスタ内平方和が小さく、クラスタ間平方和が大きくなるように実行される。最終的に全体平方和に占めるクラスタ間平方和の割合が 88.4%まで到達して計算が終了したことを示している。

```
Available components:
[1] "cluster"    "centers"     "totss"       "withinss"    "tot.withinss" "betweenss"
[7] "size"        "iter"        "ifault"
```

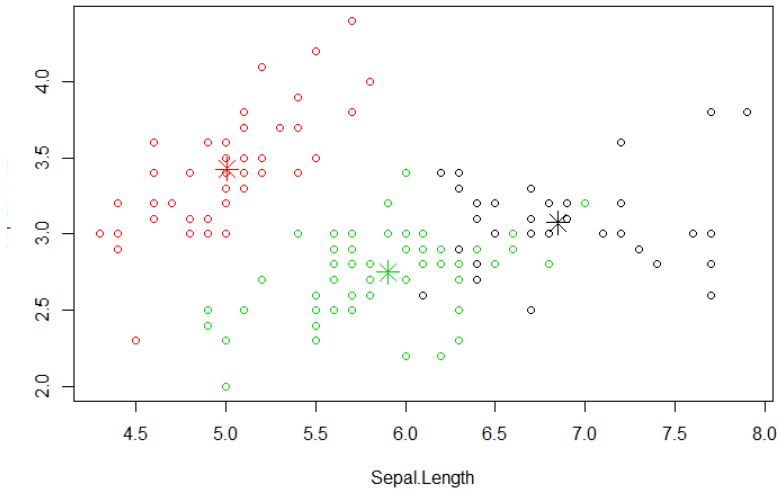
　罌と花弁の長さと幅のデータだけに基づいたクラスタリングがどの程度実際のグループに合致しているかを検証する。先ほど分離しておいた iris_class が正解クラスのベクトルで、iris_clusters$cluster がデータだけに基づいて 3 つにクラスタリングした結果のクラスのベクトルである。この二つについて table()関数でクロス集計する。

```
> table(iris_class, iris_clusters$cluster)

iris_class    1   2   3
  setosa      0  50   0
  versicolor  2   0  48
  virginica  36   0  14
```

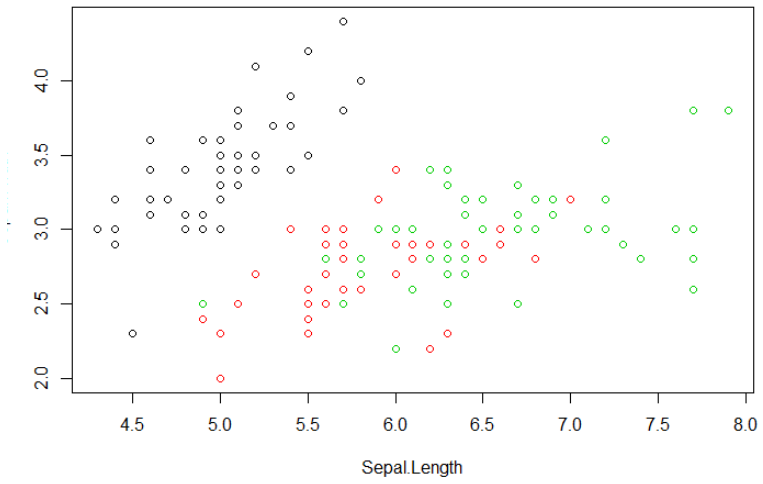
　クラスタリングの結果に付与されたクラスタ番号（1, 2, 3）は便宜的なものであるが、クラスタ 2 は setosa に対応しているようであり、50 個体すべてを完全に一つのクラスタとして分離できている。クラスタ 3 は versicolor に、クラスタ 1 は主に verginica に対応しているようである。しかし、virginica のうち 14 個体が versicolor に主に対応するであろうクラスタ 3 に分類されており、setosa に比べて、virginica と versicolr の分類が難しいことがわかる。ここで、Sepal.Length と Sepal.Width の平面において 150 個体がどのようにクラスタリングされたか、色分けし、クラスタの重心と合わせて図示する。

```
> plot(iris_data[c("Sepal.Length", "Sepal.Width")], col = iris_clusters$cluster)
> points(iris_clusters$centers[,c("Sepal.Length", "Sepal.Width")], col=1:3, pch=8, cex=
2)
```



なお、本来のクラスで色分けした場合は以下の図のようになる。

```
> plot(iris_data[c("Sepal.Length", "Sepal.Width")], col = iris_class)
```



やはり元々setosa 以外の二つのクラスの分類が難しいことがわかる。しかし、萼と花弁の長さと幅のデータのみからクラスタリングしたにしては、本来の種別に近い形で分類できていると言える。

なお、今回は k の値として 3 をアプリオリに与えたが、k を様々な値で試し、それに対してグループ内の同質性ないしは異質性がどのように変化するか plot して最適な k を見つける方法として Elbow method がある。しかし、厳密なクラスタ数 k が必要な場面は少なく、応用上は便宜的に k の値を選べば十分であるため、ここでは取り扱わない。