Кластеризация

**Функция ‘kmeans’ –** метод k средних.

**Использование:**

kmeans(x, centers, iter.max = 10, nstart = 1,algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"))

**Аргументы:**

**formula** – формула

**x** – численная матрица, содержащая объекты;

**centers** – или число кластеров, или множество исходных центров кластеров. Если аргумент представляет собой число, то выбирается случайное множество центров кластеров;

**iter.max** – максимальное число итераций;

**nstart** – если **centers** число, то данный аргумент определяет, как много случайных множеств может быть выбрано;

**algorithm** – символ, определяющий используемый алгоритм.

**Выходное значение**:

Объект класса **kmeans**, который представляет собой список следующих компонент:

**cluster** – вектор целых чисел, определяющих, в каком кластере размещены объекты;

**centers** – матрица центров кластеров;

**withinss** – сумма квадратов расстояний между точками для каждого кластера;

**size** – число точек в каждом кластере.

**Пример 1.**

**Сгенерированное обучающее множество**

Сгенерируем случайные данные, которые разбиваются на2 кластера, и используем kmeans для построения этих кластеров:

x <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2), matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))

colnames(x) <- c("x", "y")

cl <- kmeans(x, 2)

plot(x, col = cl$cluster)

points(cl$centers, col = 1:2, pch = 8, cex=2)

**Пакет ‘cluster’ –** кластеризация.

**Функция ‘clara’ –** метод k медоидов.

**Использование:**

clara(x, k, metric = "euclidean", stand = FALSE, samples = 5, sampsize = min(n, 40 + 2 \* k), trace = 0, medoids.x = TRUE, keep.data = medoids.x, rngR = FALSE)

**Аргументы:**

**x** – матрица данных (или фрейм данных), где каждая строка соответствует наблюдению, и каждый столбец соответствует переменной. Все переменные должны быть численными. Отсутствующие значения (NA) допускаются.

**k** – число кластеров. Требуется, чтобы 0<k<n, где n число объектов;

**metric** символьное описание метрики, используемой для вычисления различий между объектами. Доступны значения “euclidean” или “manhattan”;

**stand** – логическая переменная, определяющая необходимость стандартизации данных. Измерения стандартизуются для каждой переменной путем вычитания значения математического ожидания и делением на стандартное отклонение данной переменной;

**samples** – число сэмплов (вспомогательных наборов), извлекаемых из исходного набора данных;

**sampsize** – число наблюдений в каждом сэмпле (вспомогательном наборе данных). **sampsize** должно быть больше, чем число кластеров (k) и не больше числа объектов (n);

**trace** – целое число, указывающее необходимый уровень вывода хода работы алгоритма;

**medoids.x** – логическая переменная, указывающая, возвращать ли медианы (медианы совпадают с некоторыми объектами из исходных данных). Если значение **medoids.x** равно FALSE, то алгоритм возвращает только номера строк матрицы x, содержащих медианы;

**keep.data** – логическая переменная, указывающая стоит ли сохранять в возвращаемом объекте копию исходных данных;

**rngR** – логическая переменная, определяющая, использовать ли встроенный в R генератор случайных чисел, вместо встроенного в clara().

**Возвращаемое значение:** Возвращается объект класса **clara.object**, содержащий следующие компоненты:

**sample** – номера объектов, содержащихся в лучшем сэмпле (вспомогательном наборе данных): номера объектов, используемые clara() для построения финального разбиения на кластеры.

**medoids** – медианы для каждого кластера. Представляет собой матрицу, каждая строка которой содержит координаты медианы для кластера. Данное значение может быть равно NULL (если **medoids.x** равно FALSE);

**i.med** – индексы медиан кластеров (medoids <- x[i.med,]).

**clustering** – объект класса **partition.object**;

**objective** – целевая функция для финального разбиения на кластеры на исходном наборе данных;

**clusinfo** – матрица, каждая строка которой содержит численную информацию для одного кластера. Это число объектов в кластере, максимальное и среднее различие между объектами в кластере и медианой кластера. Последний столбец это максимальное различие между объектами в кластере и медианой кластера, поделенное на минимальное различие между медианой кластера и медианами других кластеров. Если это отношение мало, то кластер хорошо отделен от других кластеров;

**diss** – различие (см. **partition.object**);

**silinfo** – список с информацией о ширине для лучшего сэмпла (вспомогательного набора данных) (см. **partition.object**).

**call** – используемый вызов функции;

**data** – матрица данных (возможно стандартизованная), или NULL.

**Пример 2.**

**Обучающее множество: ruspini**

Фрейм данных**ruspini**из пакетаclusterсодержит координаты 75 точек на плоскости.

library(cluster)

#### Разобьем точки на 4 кластера:

cl <- clara(ruspini, 4)

plot(ruspini, col = cl$clustering, xlab = "x", ylab = "y")

#### Теперь разобьем то же множество точек на 5 кластеров:

cl <- clara(ruspini, 5)

plot(ruspini, col = cl$clustering, xlab = "x", ylab = "y")

**Пакет ‘cluster’ –** кластеризация.

**Функция ‘clara’ –** агломеративная иерархическая кластеризация.

**Использование:**

agnes(x, diss = inherits(x, "dist"), metric = "euclidean", stand = FALSE, method = "average", par.method, keep.diss = n < 100, keep.data = !diss)

**Аргументы:**

**x** – матрица данных или набор данных (либо матрица различий), в зависимости от значения аргумента **diss**. В случае, если **x** – матрица или набор данных, то каждая строка соответствует объекту и каждый столбец соответствует переменной. Все переменные должны быть численными. Отсутствующие значения (NA) допускаются. В случае, если **x** – матрица различий, то она обычно является выводом функций **daisy** или **dist**. Также допускается вектор длины n(n-1)/2, где n – число объектов, который будет интерпретирован тем же образом, как и вывод вышеприведенных функций. Отсутствующие значения не допускаются.

**diss** – логическая переменная, указывающая на способ рассмотрения матрицы **x** (TRUE соответствует тому, чтобы рассматривать x, как матрицу различий; FALSE – как матрицу, содержащую объекты).

**metric** – строка, соответствующая используемой метрике для вычисления различий между объектами. Доступно использование следующих значений: “euclidean” и “manhattan”. Если **x** – матрица различий, то данный аргумент игнорируется.

**stand** – логическая переменная, определяющая необходимость стандартизации данных. Измерения стандартизуются для каждой переменной путем вычитания значения математического ожидания и делением на стандартное отклонение данной переменной. Если **x** – матрица различий, то данный аргумент игнорируется.

**method** – символьная строка, определяющая метод кластеризации. Доступны 6 методов: “average” (усредняющий агломеративный алгоритм), “single” (алгоритм “простой связи”), “complete” (алгоритм “полной связи”), “ward” (метод Варда), “weighted” (взвешенный усредняющий агломеративный алгоритм) и его обобщение “flexible”, который использует константную версию формулы Ланса-Вильямса. По умолчанию используется “average”.

**par.method** – если **method="flexible"**, численный вектор длины 1, 3 или 4, содержащий параметры для функции Ланса-Вильямса.

**keep.diss, keep.data** – логические переменные, указывающие на то, необходимо ли сохранять различия и(или) входные данные x в объекте, возвращаемом функцией.

**Возвращаемое значение:** Возвращается объект класса agnes, представляющий собой список со следующими компонентами:

**order** – вектор, содержащий перестановку оригинальных объ-ектов, которая позволяет построение таким образом, чтобы ветви дендрограммы не пересекались.

**order.lab** – вектор, похожий на **order**, но содержащий метки объектов, а не их номера. Этот компонент доступен только в том случае, если объекты имеют метки.

**height** – вектор с расстояниями между объединяемыми кластерами.

**ac** – агломеративный коэффициент, измеряющий кластерную структуру набора данных. Для каждого объекта i обозначим m(i) его различие с первым кластером, с которым он объединялся, разделенное на различные объединения на финальном шаге алгоритма. **ac** – это среднее среди всех 1-m(i). Его также можно рассматривать, как среднюю ширину (или процент заполняемости). Так как **ac** растет вместе с ростом числа объектов, эта метрика не может быть использована для сравнения наборов данных с сильно отличающимися размерами.

**merge** – матрица размера (n-1)x2, где n – число объектов. Строка i описывает объединение кластеров на шаге i. Если число j в строке отрицательное, то единичный объект |j| присоединяется на данной стадии. Если j положительно, то осуществляется присоединение кластера, сформированного на стадии j работы алгоритма.

**diss** – объект класса **dissimilarity**, представляющий матрицу различий для исходных данных.

**data** – матрица, содержащая исходные или стандартизованные данные (в зависимости от значения аргумента **stand**).

**Пример 3.**

**Обучающее множество: Agriculture**

Набор данных**agriculture**из пакетаclusterсодержитданные по странам о валовом национальном продукте на душу населения (первая переменная) и проценте населения, работающего в сельском хозяйстве.

|  |  |  |
| --- | --- | --- |
| Страна | x | y |
| B | 16.8 | 2.7 |
| DK | 21.3 | 5.7 |
| D | 18.7 | 3.5 |
| GR | 5.9 | 22.2 |
| E | 11.4 | 10.9 |
| F | 17.8 | 6.0 |
| IRL | 10.9 | 14.0 |
| I | 16.6 | 8.5 |
| L | 21.0 | 3.5 |
| NL | 16.4 | 4.3 |
| P | 7.8 | 17.4 |
| UK | 14.0 | 2.3 |

Построим дендрограмму:

library(cluster)

data(agriculture)

plot(agnes(agriculture))

Как видно из рис. 1, в один кластер объединены страны с большой долей сельского хозяйства (Греция, Португалия, Испания и Ирландия).

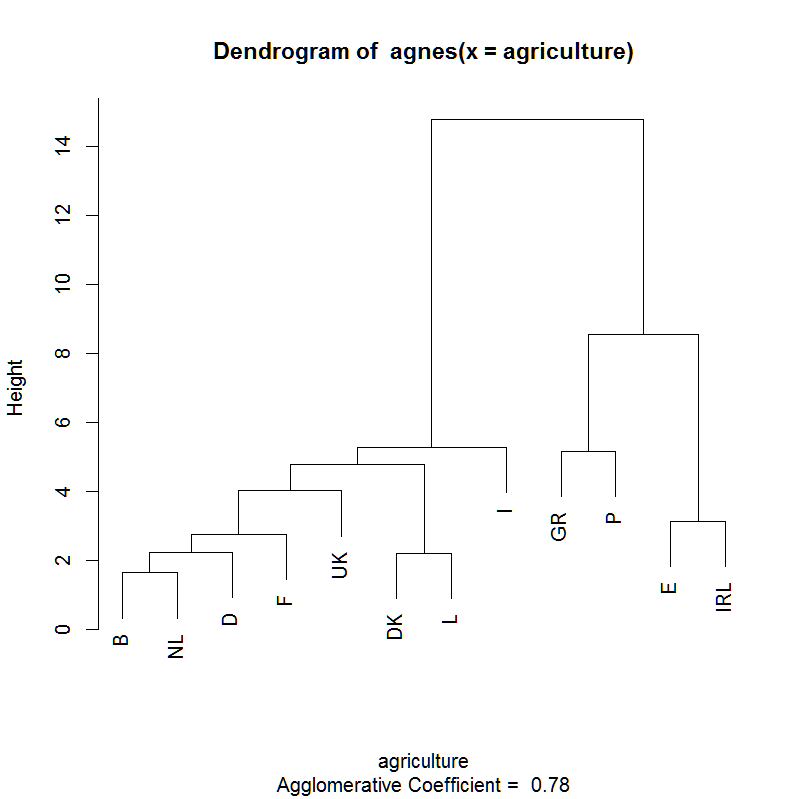


Рис. 1.

**Задание**

1. Разбейте множество объектов из набора данных pluton в пакете «cluster» на 3 кластера методом центров тяжести (kmeans). Сравните качество разбиения в зависимости от максимального числа итераций алгоритма.
2. Сгенерируйте набор данных в двумерном пространстве, состоящий из 3 кластеров, каждый из которых сильно “вытянут” вдоль одной из осей. Исследуйте качество кластеризации методом clara в зависимости от 1) использования стандартизации; 2) типа метрики. Объясните полученные результаты.
3. Постройте дендрограмму для набора данных votes.repub в пакете «cluster» (число голосов, поданных за республиканцев на выборах с 1856 по 1976 год). Строки представляют 50 штатов, а столбцы - годы выборов (31). Проинтерпретируйте полученный результат.
4. Постройте дендрограмму для набора данных animals в пакете «cluster». Данные содержат 6 двоичных признаков для 20 животных. Переменные - [ , 1] war теплокровные; [ , 2] fly летающие; [ , 3] ver позвоночные; [ , 4] end вымирающие; [ , 5] gro живущие в группе; [ , 6] hai имеющие волосяной покров. Проинтерпретируйте полученный результат.
5. Рассмотрите данные из файла seeds\_dataset.txt, который содержит описание зерен трех сортов пшеницы: Kama, Rosa and Canadian. Признаки: 1. область A, 2. периметр P, 3. компактность C = 4\*pi\*A/P^2, 4. длина зерна, 5. ширина зерна, 6. коэффициент ассиметрии, 7. длина колоска.