МАШИННОЕ ОБУЧЕНИЕ
ЛАБОРАТОРНЫЙ ПРАКТИКУМ
(предварительный вариант)

Ю.В. Горюнов, Н.Ю. Золотых, А.Н. Половинкин

4 декабря 2007 г.

0.1. Кластеризация

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

Аргументы:

`x` — численная матрица, содержащая объекты;

`centers` — или число кластеров, или множество исходных центров кластеров. Если аргумент представляет собой число, то выбирается случайное множество центро 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.

Возвращаемое 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: Сгенерируем случайные данные, которые разбиваются на 2 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)
```

```
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 пере-

менной. 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 друних 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: Фрейм данных `ruspini` из пакета `cluster` содержит координаты 75 точек на плоскости. Разобьем их на 4 кластера:

```
> library(cluster)
> cl <- clara(ruspini, 4)
> plot(ruspini, col = cl$clustering, xlab = "x", ylab = "y")
```

Результат см. на рис. 1.

Теперь разобьем то же множество точек на 5 кластеров:

```
> cl <- clara(ruspini, 5)
> plot(ruspini, col = cl$clustering, xlab = "x", ylab = "y")
```

Результат см. на рис. 2.

```
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

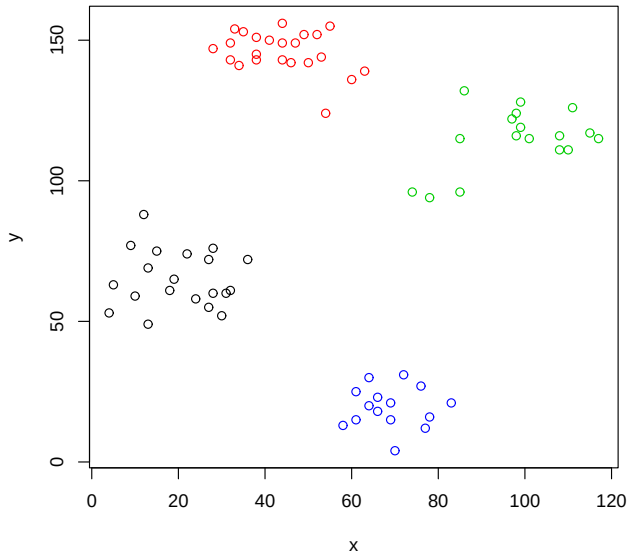


Рис. 1. Множество точек, разбитое на 4 кластера

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

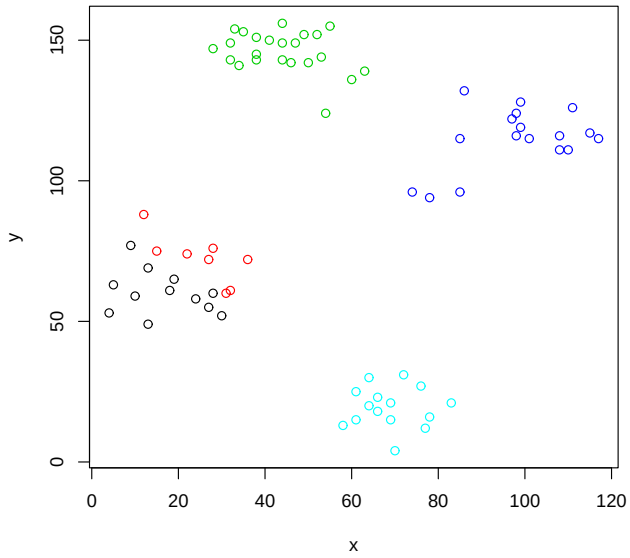


Рис. 2. То же множество точек, разбитое на 5 кластеров

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 усредняющмй 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 объектов, 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 различие of the merger на финальном шаге алгоритма. `ac` – это среднее среди всех `1-m(i)`. Его также можно рассматривать, как среднюю ширину (или процент заполняемости) of the banner plot. Т.к. `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)`$\times$`2`, 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.

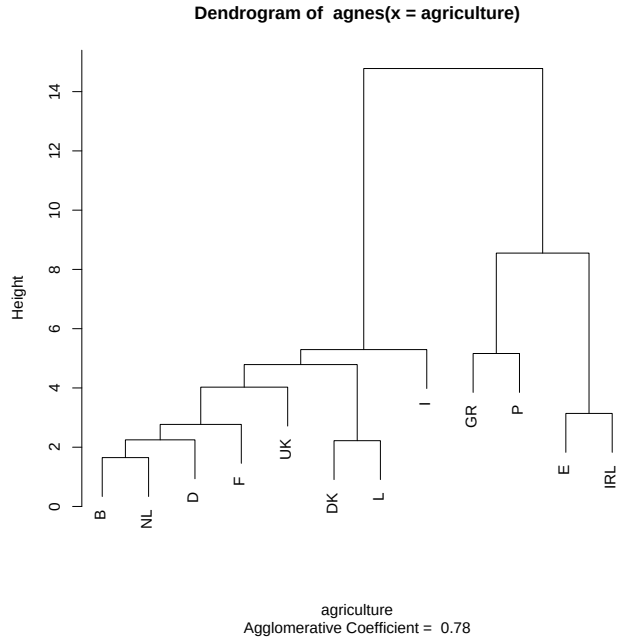
`diss` — объект класса `dissimilarity`, представляющий матрицу различий для исходных данных.

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

Пример: Набор данных `agriculture` из пакета `cluster` содержит данные о валовом национальном продукте на душу населения и проценте населения, работающего в сельском хозяйстве. Построим дендрограмму:

```
> data(agriculture)
> plot(agnes(agriculture))
```

Как видно из рис. 0.1, в один кластер объединены страны с большой долей сельского хозяйства (Греция, Португалия, Испания и Ирландия).



0.1.1. Задания

1) Разбейте множество объектов из набора данных `pluton` на 3 кластера методом центров тяжести (kmeans). Сравните качество разбиения в зависимости от макисмального 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` (число голосов, поданных за республиканцев на выборах с 1856 по 1976 год). Проинтерпретируйте полученный результат.