**ЗАДАНИЕ 11 - Решение задачи кластеризации методом k средних**

Выполнил студент 2 курса

группы 09-715(1)

Санамян Артак Размикович

**Текст задания:**

1. Примените метод k средних для решения задачи кластеризации (в качестве исходных данных Вы можете использовать те данные, для которых ранее решалась задача классификации, удалив из них столбец со значениями переменной отклика).   
  
2. Если число признаков m = 2, визуализуйте данные (постройте облако точек, "раскрасив" точки в цвета, соответствующие номерам кластеров) и сравните полученные кластеры с облаками точек в соответствующей задаче классификации.

3. Исследуйте работу метода, варьируя значения параметра.

4. Проанализируйте полученные результаты, оформите отчёт.

Источник числовых данных, использованный для выполнения задания: <http://archive.ics.uci.edu/ml/datasets/Flags>

**Описание выполненной работы:**

Используем те данные, для которых ранее решалась задача классификации, удалив из них столбец со значениями переменной отклика.

Данные представляют собой информацию о странах на 1990 год. Необходимо определить религию страны:

0 – Католики;

1 – Остальные христиане;

2 – Мусульмане;

3 – Буддисты;

4 – Индуисты;

5 – Язычники;

6 – Марксисты;

7 – Другие.

Факторами являются следующие признаки:

1. Название страны (не числовой);
2. Материк (1-6);
3. Зона: географический квадрант, основанный на Гринвиче и экваторе(1-4)
4. Площадь: в тысячах квадратных километров;
5. Население: в миллионах;
6. Язык: 1 = английский, 2 = испанский, 3 = французский, 4 = немецкий, 5 = славянский, 6 = другой индоевропейский, 7 = китайский, 8 = арабский, 9 = японский / турецкий / финский / мадьяр, 10 = Другие;
7. Религия: 0 = католик, 1 = другой христианин, 2 = мусульманин, 3 = буддист, 4 = индус, 5 = этнический, 6 = марксист, 7 = другие;
8. Полосы: количество вертикальных полос на флаге;
9. Полосы: количество горизонтальных полос на флаге;
10. Цвета: количество различных цветов на флаге;
11. Красный: 0 – нет, 1 – цвет есть на флаге;
12. Зеленый: 0 – нет, 1 – цвет есть на флаге;
13. Синий: 0 – нет, 1 – цвет есть на флаге;
14. Золотой: 0 – нет, 1 – цвет есть на флаге (также желтый);
15. Белый: 0 – нет, 1 – цвет есть на флаге;
16. Черный: 0 – нет, 1 – цвет есть на флаге;
17. Оранжевый: 0 – нет, 1 – цвет есть на флаге (также коричневый);
18. Основной оттенок (не числовой);
19. Количество кругов;
20. Количество крестов;
21. Количество Андреевских крестов
22. Количество частей флага;
23. Количество символов солнца или звезд;
24. Полумесяц: 1, если присутствует символ полумесяца, иначе 0
25. Треугольник: 1, если присутствуют какие-либо треугольники, 0 в противном случае
26. 1, если присутствует неодушевленное изображение (например, лодка), в противном случае 0
27. 1, если присутствует одушевленное изображение (например, орел, дерево, человеческая рука), 0 в противном случае
28. Текст: 1 если какие-либо буквы или письма на флаге (например, девиз или слоган), 0 в противном случае
29. Цвет в верхнем левом углу;
30. Цвет в левом нижнем углу.

Применим нормализацию данных по формуле:

Положим и вызовем метод k – средних. Результат кластеризации:

> # Вызовем метод к средних

> flag\_clusters <- kmeans(flag, 2)

> flag\_clusters

K-means clustering with 2 clusters of sizes 86, 108

Cluster means:

V2 V3 V4 V5 V6 V8 V9 V10

1 0.4813953 0.4922481 0.03303951 0.02295589 0.3953488 0.11162791 0.08471761 0.4568106

2 0.5407407 0.3333333 0.02982373 0.02318489 0.5514403 0.07407407 0.13161376 0.2685185

V11 V12 V13 V14 V15 V16 V17 V19

1 0.8488372 0.6162791 0.4883721 0.98837209 0.5116279 0.4186047 0.18604651 0.04651163

2 0.7407407 0.3518519 0.5277778 0.05555556 0.9444444 0.1481481 0.09259259 0.03935185

V20 V21 V22 V23 V24 V25 V26

1 0.08139535 0.13953488 0.03488372 0.02534884 0.02325581 0.19767442 0.45348837

2 0.06944444 0.05555556 0.03935185 0.02962963 0.08333333 0.09259259 0.09259259

V27 V28

1 0.38372093 0.16279070

2 0.05555556 0.01851852

Clustering vector:

[1] 1 1 2 1 1 1 2 1 2 2 2 2 1 2 2 1 1 1 2 1 2 1 2 1 1 1 1 1 2 2 1 2 1 1 1 1 2 1 1 2 1

[42] 2 2 2 1 2 2 2 1 2 1 1 2 2 1 2 1 1 2 2 2 1 1 2 1 1 1 1 2 2 1 1 2 1 1 1 2 2 1 2 2 2

[83] 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 1 1 2 2 2 1 1 2 1 2 2 2 2

[124] 2 2 2 2 1 2 2 2 2 2 2 1 1 2 2 2 1 2 2 1 1 2 1 2 1 2 2 2 1 2 2 2 2 1 1 1 1 1 1 2 1

[165] 1 1 2 2 2 1 2 1 2 2 2 2 1 1 2 1 2 2 1 2 1 1 1 1 1 2 2 1 1 1

Within cluster sum of squares by cluster:

[1] 244.039 207.861

(between\_SS / total\_SS = 14.5 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss"

[6] "betweenss" "size" "iter" "ifault"

Размеры кластеров – 86 и 108.

Среднее значение каждого фактора в кластере:

Cluster means:

V2 V3 V4 V5 V6 V8 V9 V10

1 0.4813953 0.4922481 0.03303951 0.02295589 0.3953488 0.11162791 0.08471761 0.4568106

2 0.5407407 0.3333333 0.02982373 0.02318489 0.5514403 0.07407407 0.13161376 0.2685185

V11 V12 V13 V14 V15 V16 V17 V19

1 0.8488372 0.6162791 0.4883721 0.98837209 0.5116279 0.4186047 0.18604651 0.04651163

2 0.7407407 0.3518519 0.5277778 0.05555556 0.9444444 0.1481481 0.09259259 0.03935185

V20 V21 V22 V23 V24 V25 V26

1 0.08139535 0.13953488 0.03488372 0.02534884 0.02325581 0.19767442 0.45348837

2 0.06944444 0.05555556 0.03935185 0.02962963 0.08333333 0.09259259 0.09259259

V27 V28

1 0.38372093 0.16279070

2 0.05555556 0.01851852

Вызовем метод k – средних при Результат кластеризации:

|  |
| --- |
| > flag\_clusters <- kmeans(flag, 3)  > flag\_clusters  K-means clustering with 3 clusters of sizes 74, 56, 64  Cluster means:  V2 V3 V4 V5 V6 V8 V9 V10 V11 V12 V13 V14 V15  1 0.6540541 0.09009009 0.02040298 0.02595453 0.7342342 0.06756757 0.1245174 0.2722008 0.7567568 0.4189189 0.4189189 0.06756757 0.9729730  2 0.5321429 0.39285714 0.04140016 0.03192319 0.5496032 0.13571429 0.0994898 0.3367347 0.8571429 0.6071429 0.1607143 0.91071429 0.2321429  3 0.3375000 0.77604167 0.03490830 0.01202877 0.1319444 0.07812500 0.1049107 0.4575893 0.7656250 0.4062500 0.9218750 0.54687500 0.9531250  V16 V17 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28  1 0.1756757 0.08108108 0.05405405 0.047297297 0.01351351 0.020270270 0.01621622 0.12162162 0.09459459 0.1351351 0.04054054 0.02702703  2 0.4821429 0.07142857 0.02232143 0.008928571 0.01785714 0.004464286 0.01500000 0.03571429 0.23214286 0.2678571 0.25000000 0.05357143  3 0.1875000 0.25000000 0.04687500 0.164062500 0.25000000 0.085937500 0.05218750 0.00000000 0.10937500 0.3750000 0.34375000 0.17187500  Clustering vector:  [1] 2 2 1 3 2 2 3 3 3 3 3 1 2 1 1 3 2 3 2 3 1 2 1 3 3 2 3 2 1 1 2 3 2 3 1 2 3 2 3 1 2 3 3 3 2 1 1 1 3 3 3 2 3 1 2 3 3 3 1 1 3 3 2 3 2 2 2 3  [69] 1 1 2 3 3 2 2 2 2 3 3 1 3 1 1 1 1 3 1 1 1 2 1 1 2 1 3 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 2 2 3 1 1 2 3 2 2 1 1 1 3 3 3 1 1 3 1 1 1 1 1 3 2 3  [137] 1 1 1 3 3 1 3 2 1 2 1 2 1 3 1 3 1 3 1 1 2 2 3 2 3 3 1 2 3 2 1 1 1 2 1 2 1 3 1 1 3 3 1 2 3 3 3 3 2 2 2 3 2 3 1 2 2 2  Within cluster sum of squares by cluster:  [1] 123.5018 126.2506 164.1852  (between\_SS / total\_SS = 21.6 %)  Available components:  [1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" "size" "iter" "ifault" |
|  |
| |  | | --- | |  | |

Вызовем метод k – средних при Результат кластеризации:

|  |
| --- |
| > flag\_clusters <- kmeans(flag, 4)  > flag\_clusters  K-means clustering with 4 clusters of sizes 53, 55, 21, 65  Cluster means:  V2 V3 V4 V5 V6 V8 V9 V10 V11 V12 V13 V14 V15  1 0.6452830 0.1320755 0.023985392 0.031558850 0.6687631 0.09433962 0.09299191 0.2803235 0.8301887 0.71698113 0.1132075 0.03773585 0.9056604  2 0.5272727 0.4303030 0.041208171 0.031818182 0.5313131 0.14909091 0.10259740 0.3584416 0.8545455 0.54545455 0.2000000 0.96363636 0.2727273  3 0.3428571 0.6666667 0.003853823 0.003070673 0.1164021 0.07619048 0.03741497 0.7210884 0.9523810 0.90476190 1.0000000 1.00000000 1.0000000  4 0.4523077 0.5179487 0.037596231 0.015247253 0.4068376 0.04307692 0.15604396 0.2857143 0.6461538 0.06153846 0.9384615 0.23076923 0.9538462  V16 V17 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28  1 0.20754717 0.15094340 0.05188679 0.04716981 0.03773585 0.009433962 0.01132075 0.13207547 0.07547170 0.13207547 0.05660377 0.03773585  2 0.52727273 0.09090909 0.02727273 0.00000000 0.01818182 0.000000000 0.01418182 0.03636364 0.23636364 0.34545455 0.25454545 0.05454545  3 0.28571429 0.52380952 0.09523810 0.26190476 0.42857143 0.107142857 0.02476190 0.00000000 0.04761905 0.95238095 0.90476190 0.47619048  4 0.09230769 0.03076923 0.03076923 0.10000000 0.09230769 0.069230769 0.05353846 0.03076923 0.13846154 0.04615385 0.04615385 0.01538462  Clustering vector:  [1] 2 2 1 3 2 2 4 4 4 4 4 1 2 1 1 2 2 3 1 3 1 2 4 4 3 2 3 2 4 1 2 4 2 3 1 2 4 2 4 1 2 4 4 4 2 4 1 1 3 4 4 2 4 1 2 4 3 3 4 4 4 2 2 4 2 2 2 2  [69] 4 4 2 3 4 2 2 2 2 4 3 1 4 1 1 1 1 1 4 1 1 2 1 1 2 1 3 1 4 1 1 4 1 2 4 1 1 4 1 2 1 4 2 2 1 4 1 2 3 1 2 4 4 4 4 4 4 1 1 4 4 1 4 1 1 4 2 3  [137] 4 4 1 3 4 1 3 2 4 2 1 2 1 4 1 4 4 1 4 4 2 2 3 2 4 3 1 2 2 4 1 1 4 2 4 1 1 4 1 1 3 4 1 2 4 4 3 4 2 2 2 3 2 4 4 2 2 2  Within cluster sum of squares by cluster:  [1] 85.26405 128.83855 38.86075 109.26025  (between\_SS / total\_SS = 31.4 %)  Available components:  [1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss" "size" "iter" "ifault" |
|  |
| |  | | --- | |  | |

**Код программы:**

# Назначим рабочий директорий

setwd("C://Users/pc/Documents/Магистратура/2 семестр/R/task9 KMid")

getwd()

#очищаем рабочее пространство

rm (list=ls())

# Прочтём данные о флагах

flagdata <- read.csv("flagdata.txt", sep=",", head=FALSE)

# Удалим столбец с классами

# flagdata = flagdata[-7]

# flagdata

# Формула нормализации данных

normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x)))}

# Нормализуем данные и исключим нечисловые данные

flag <- as.data.frame(lapply(flagdata[c(2:6, 8:17, 19:28)], normalize))

m = dim(flag)[2] # количество факторов # 25

n = dim(flag)[1] # объём выборки # 194

# Добавим названия столбцов

# colnames(flag)<- c('V2','V3','V4','V5','V6','V8','V9','V10','V11','V12','V13','V14','V15','V16','V17','V19','V20','V21','V22','V23','V24','V25','V26','V27','V28','Type')

flag

# Вызовем метод к средних

flag\_clusters <- kmeans(flag, 2)

flag\_clusters

flag\_clusters <- kmeans(flag, 3)

flag\_clusters

flag\_clusters <- kmeans(flag, 4)

flag\_clusters