*Лабораторная работа №6*

*(часть 1)*

**Задачи классификации и кластеризации**

## Цель работы:

Закрепить знания, об алгоритмах классификации и кластеризации данных, ознакомиться с некоторыми функциями языка R, осуществляющими этот вид анализа, принципами их работы. Научиться визуализировать результаты работы функций кластерного анализа и классификаторов, интерпретировать полученные результаты.

## Общие сведения

6.2.1. Классификация

Под***классификацией*** будем понимать отнесение объектов (наблюдений, событий) к одному из ***заранее известных классов***.

Классификация - это закономерность, позволяющая делать вывод относительно определения характеристик конкретной группы. Таким образом, для проведения классификации должны присутствовать признаки, характеризующие группу, к которой принадлежит то или иное событие или объект (обычно при этом на основании анализа уже классифицированных событий формулируются правила причисление объекта к группе).

Классификация относится ***к стратегии*** ***обучения с учителем*** (supervised learning), которое также именуют контролируемым или управляемым обучением.

Классификация может быть ***одномерной*** (по одному признаку) и ***многомерной*** (по двум и более признакам).

6.2.2. Кластеризация

Задача ***кластеризации*** сходна с задачей классификации, но ее отличие в том, что классы изучаемого набора данных заранее не предопределены и формируются автоматически, поэтому синонимами термина "кластеризация" являются "автоматическая классификация", "***обучение без учителя***" и "таксономия".

Кластеризация предназначена для разбиения совокупности объектов на однородные группы (кластеры или классы). Если данные выборки представить, как точки в признаковом пространстве, то задача кластеризации сводится к определению "сгустков точек".

Цель кластеризации - поиск таких сгустков.

Кластеризация является описательной процедурой, она не делает никаких статистических выводов, но дает возможность провести **разведочный анализ** и изучить "структуру данных".

Само понятие "кластер" определено неоднозначно: каждый алгоритм (или даже каждое исследование одного и того же алгоритма) формирует свои "кластеры". Переводится понятие кластер (cluster) как "скопление", "гроздь".

|  |  |  |
| --- | --- | --- |
| Таблица 6.1. Сравнение классификации и кластеризации | | |
| **Характеристика** | **Классификация** | **Кластеризация** |
| Контролируемость обучения | Контролируемое обучение | Неконтролируемое обучение |
| Стратегия | Обучение с учителем | Обучение без учителя |
| Наличие метки класса | Обучающее множество сопровождается меткой, указывающей класс, к которому относится наблюдение | Метки класса обучающего множества неизвестны |
| Основание для классификации | Новые данные классифицируются на основании обучающего множества | Дано множество данных с целью установления существования классов или кластеров данных |

## Некоторые функции для классификации в R

**Шаг 1.**

**(Сначала проведите дескриптивный анализ и анализ распределения независимых параметров)**

**Шаг 2.**

Поскольку в большинстве задач количество классов заранее неизвестно, то для выделения предполагаемого количества групп (классов), проведем иерархический кластерный анализ, результатом которого является дендрограмма (рис.6.1), позволяющая сформировать представление исследователя о возможном разделении выборки на классы.

*Шаг 2.1. Чтение данных, отбросить столбец, классифицирующий ирисы*

data(iris)

fix(iris)

labels\_iris<-iris[,5] # Этот столбец позже используем для подписей на дендрограмме

iris\_C<-iris[,-5]

*Шаг 2.2. Удаление пропущенных значений*

# В данной задаче пропущенных значений нет.

# Удалять нечего.

*Шаг 2.3. Стандартизация переменных.*

# В данной задаче переменные существенно различны (риc 6.1.).

# Стандартизировать надо.

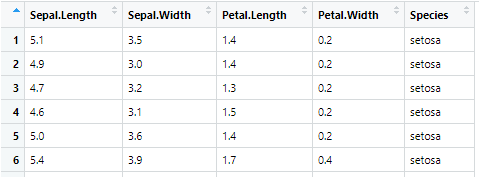


Рис. 6.1. Фрагмен таблицы iris.

maxs <- apply(iris\_C, 2, max)

mins <- apply(iris\_C, 2, min)

iris\_C <- scale(iris\_C, center = mins, scale = maxs - mins)

*Шаг 2.3.*  Создаем матрицу попарных расстояний (по умолчанию - Евклидово расстояние)

dist.iris <- dist(iris\_C)

*Шаг 2.4.*  Проводим кластерный анализ, результаты записываем в список clust.iris

# hclust ожидает матрицу расстояния, а не исходные данные.

clust.iris <- hclust(dist.iris, "ward.D")

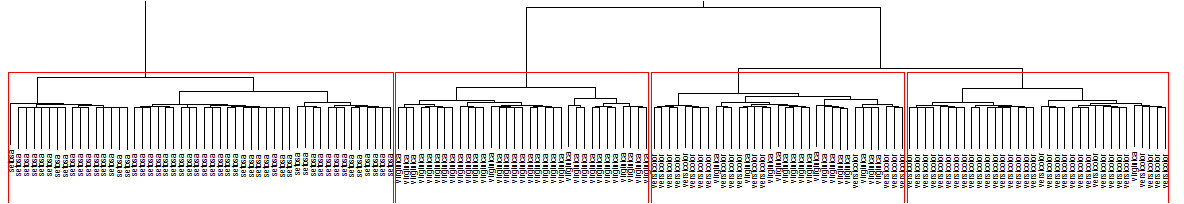
*Шаг 2.5. Построение дендрограммы*

plot(clust.iris)

Увидев дендрограмму (рис.6.2), понимаем, что наиболее целесообразно разделить ее на 3 либо на 4 кластера. Попробуем разделить на 4 кластера.

plot(clust.iris, labels\_iris, cex=0.5)

rect.hclust(clust.iris, k=4, border="red")



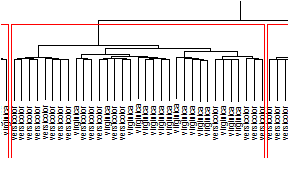


Рис. 6.2. Дендрограмма ирисов, деление на 4 кластера

Присмотревшись внимательнее ко второму справа кластеру, обнаруживаем, что он состоит из смеси двух сортов ирисов – versicolor и verginica, и, согласно дендрограмме, его лучше присоединить к первому справа кластеру. Поэтому дальнейший анализ проводим для 3-х кластеров.

*Шаг 2.6.*  Разделим выборку на 3 кластера

# Вектор groups содержит номер кластера, в который попал классифицируемый объект

groups <- cutree(clust.iris, k=3)

# Для каждой группы определяем средние значения характеристик и строим датафрейм.

# в 1-ом кластере

g1<-colMeans(iris[groups==1, 1:4])

# во 2-ом кластере

g2<-colMeans(iris[groups==2, 1:4])

# в 3-ем кластере

g3<-colMeans(iris[groups==3, 1:4])

df<-data.frame(g1,g2,g3)

df1<-t(df)

df<-t(df1)

barplot(df, col=c("red","green","blue","yellow")) # построим график

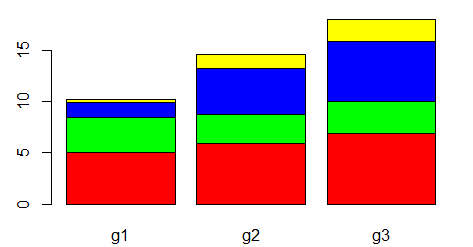


Рис. 6.3.Группы ирисов

Поработав немного над графиком, получим те же данные, несколько в другом виде (рис.6.4.)

barplot(df, ylim=c(0,12),

main = "Groups of iris", axes = FALSE, col=c("red","green","blue","yellow"),

beside=TRUE)

axis(2, at = 0:5, labels = 0:5)

legend("top", legend = rownames(df), col=c("red","green","blue","yellow"), lwd=10, bty = "n")

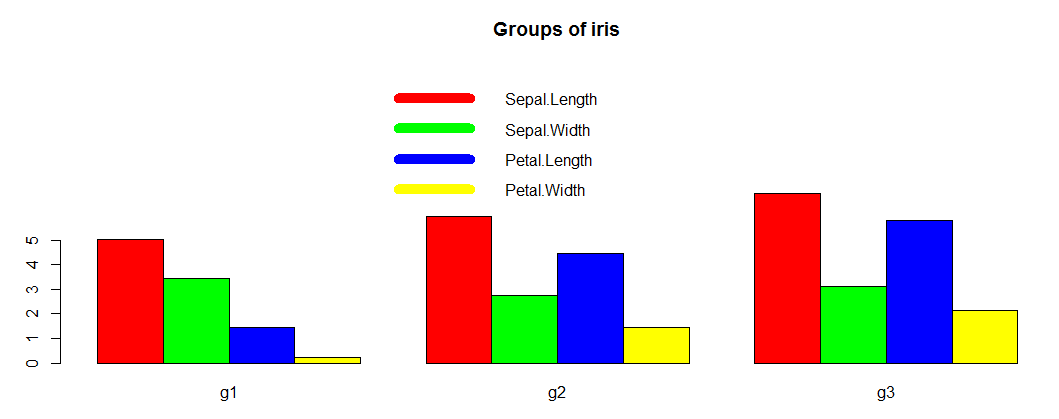


Рис. 6.4. Группы ирисов с легендой

Строим диаграмму каменная осыпь:

plot(1:149, clust.iris$height, type='b')

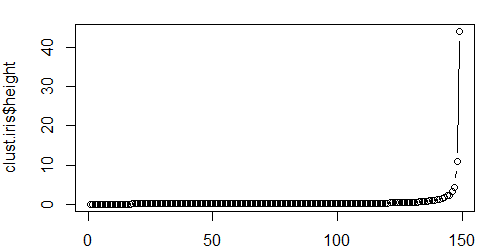


Рис. 6.5. Диаграмма "Каменная осыпь"

Первичное представление о многомерной классификации можно также получить с помощью библиотеки lattice (решетка).

Функция R **xyplot** () этой библиотеки используется для создания двумерных диаграмм рассеяния или графиков временных рядов. Упрощенный формат выглядит следующим образом:

library (lattice)

xyplot(y ~ x, data)

Итак, чтобы создать диаграмму рассеяния, используем функцию xyplot (формула, данные). Чтобы построить решетчатый график, нужно указать как минимум два аргумента:

* **формула:** как правило, представляется в виде у ~ х | Z. Это означает, что мы хотим создать график зависимости ***y*** от ***x*** с условием ***z*** . Другими словами, создаем график для каждого уникального значения ***z*** . Каждая из переменных в формуле должна быть столбцом во фрейме данных, который мы указываем в аргументе данных.
* **данные:** фрейм данных, который содержит все столбцы, указанные в аргументе формулы.

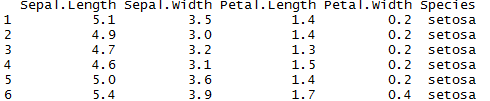
Снова возьмем для экспериментов датасет ИРИСЫ:

my\_data <- iris

Наша классификация будет содержать 3 вида (кдасса) цветов:

"setosa" "versicolor" "virginica"

head(my\_data)



и выведем график рассеяния с минимальным количеством параметров:

xyplot(Sepal.Length ~ Petal.Length, data = my\_data)

получим распределение элементов нашей выборки в зависимости: длина чашелистика от длины лепестка (рис. 6.5 (а))

|  |  |
| --- | --- |
|  |  |
| а) без выделения сортов | б) с выделением сортов |
| Рис. 6.5. Зависимость длины чашелистика от длины лепестка в наборе данных "Ирисы" | |

Для варианта б) нужно добавить параметр group = Species:

xyplot(Sepal.Length ~ Petal.Length, group = Species, data = my\_data, auto.key = TRUE)

очевидно, что этот параметр поможет нам визуально отличить группы друг от друга (параметр auto.key = TRUE добавит легенду цветов сверху от графика).

**Как видим, здесь уже понадобилось знание о возможных кластерах или классах.**

**Шаг 3.**

Построим боксплот, который поможет нам убедиться, что мы действительно имеем 3 класса существенно отличных друг от друга:

boxplot(Sepal.Length ~ Species, data = iris, ylab = "Sepal.Length",

+ frame = FALSE, col = "lightgray")

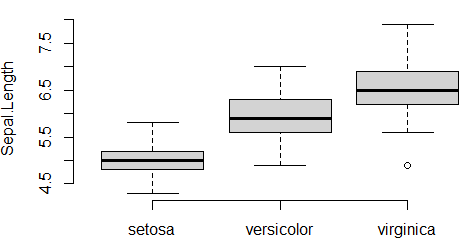


Рис 6.6. Боксплот, отражающий характеристики классов цветов

**Шаг 4.**

***Задача:***

Определить принадлежность цветка ирис к одному из трех видов (классов) по геометрическим параметрам чашелистиков и лепестков:

|  |  |
| --- | --- |
| Sepal—чашелистик, Petal—лепесток, Species—вид. |  |
| Рис 6.7. График, классифицирующий ирисы согласно их геометрическим параметрам. | |

Для получения такой классификации модифицируем параметры функции еще раз:

xyplot(Sepal.Length~Petal.Length+Petal.Width|Species,data=iris, grid = T, auto.key=TRUE)

Мы изменили формулу, указав, что длина лепестков, как мы считаем, зависит от длины и ширины чашелистиков и видов цветов.

**\*) Синтаксис функции для группировки данных: y ~ x | group.**

**Добавим линию сглаживания:**

xyplot(

Sepal.Length ~ Petal.Length | Species,

layout = c(3, 1), # panel with ncol = 3 and nrow = 1

group = Species, data = iris,

type = c("p", "smooth"), # Show points and smoothed line

scales = "free" # Make panels axis scales independent

)

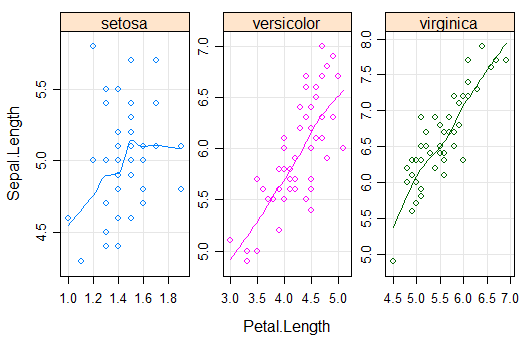


Рис 6.8. График, классифицирующий ирисы согласно их геометрическим параметрам с линией сглаживания

***Шаг 5. Построим трехмерный график наших классов***

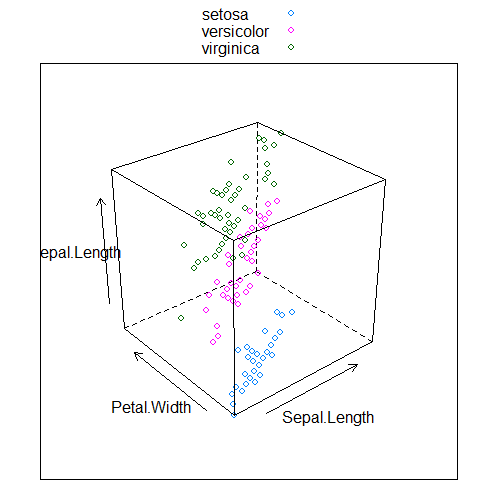


Рис 6.9. Трехмерный график, классифицирующий ирисы согласно их геометрическим параметрам.

cloud(Sepal.Length ~ Sepal.Length \* Petal.Width, group = Species, data = iris, auto.key = TRUE)

Можно так:

>install.packages("scatterplot3d")

library("scatterplot3d")

Трехмерная классификация ирисов:

> colors <- c("#999999", "#E69F00", "#56B4E9")

> colors <- colors[as.numeric(iris$Species)]

> scatterplot3d(iris[,1:3], pch = 16, color=colors)

|  |  |
| --- | --- |
|  |  |
| а) без легенды | б) с легендой |

Рис 6.10. Трехмерный график, классифицирующий ирисы согласно их геометрическим параметрам (scatterplot3d).

Трехмерная классификация ирисов с легендой:

s3d <- scatterplot3d(iris[,1:3], pch = shapes, color=colors)

legend("bottom", legend = levels(iris$Species),

col = c("#999999", "#E69F00", "#56B4E9"),

pch = c(16, 17, 18),

inset = -0.25, xpd = TRUE, horiz = TRUE)

## **6.4. Формулировка задания:**

За наборами данных обращаться к преподавателю.

Общая схема выполнения ЛР 6:

Часть первая - (ЛР 6.1) - предназначена для разведочного анализа данных и определения количества результирующих выходных групп.

Часть вторая - (ЛР 6.2) - использует предположение о количестве выходных классов Вашего набора данных, классификация строится из этого предположения.

1. Выполнить иерархическую кластеризацию Вашего набора данных, построив дендрограмму. Подробно обосновать Ваш выбор числа групп.
2. Построить диаграмму "Каменная осыпь"
3. Визуализировать результат
4. В целом: выполнить шаги 1-5 анализа для своего набора данных (если какие-то из шагов нерелевантны вашему набору данных, объяснить почему).

Вариант берется согласно вашего номера из списка группы.

|  |  |  |
| --- | --- | --- |
| № | Задание | Параметры |
|  | Экономика городов |  |
|  | Занятость в странах Европы |  |
|  | French\_Food – расходы на питание во Франции |  |
|  | Исследование Пищевых Предпочтений 2019 Года |  |
|  | Страны\_по\_уровню жизни |  |
|  | Классификация филиалов сети магазинов |  |
|  | Туристские поездки |  |
|  | Титаник |  |
|  | Темпы роста стран |  |
|  | Covid\_Russia |  |
|  | Шоколад |  |
|  | Меню ресторана |  |
|  | GooglePlay |  |
|  | Выборы в США |  |
|  | Football |  |
|  | Крушение самолетов |  |
|  | Школьники |  |
|  | ЗП\_в \_Америке |  |