БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет радиофизики и компьютерных технологий
Кафедра радиофизики и цифровых медиа технологий

Лабораторная работа по курсу
Статистическая радиофизика

Введение в машинное обучение. Решение задачи кластеризации

Минск 2023 г.

Цель работы: изучить теоретическую часть кластерного анализа и применить изученный материала на практике.

Общие сведения:

Кластерный анализ (Data clustering) – задача разбиения заданной выборки объектов (ситуаций) на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались. Задача кластеризации относится к широкому классу задач обучения без учителя.

Типы входных данных:

- Признаковое описание объектов. Каждый объект описывается набором своих характеристик, называемых признаками. Признаки могут быть числовыми или нечисловыми.
- Матрица расстояний между объектами. Каждый объект описывается расстояниями до всех остальных объектов обучающей выборки.

Матрица расстояний может быть вычислена по матрице признаковых описаний объектов бесконечным числом способов, в зависимости от того, как ввести функцию расстояния (метрику) между признаковыми описаниями. Часто используется евклидова метрика, однако этот выбор в большинстве случаев является эвристикой и обусловлен лишь соображениями удобства.

Обратная задача – восстановление признаковых описаний по матрице попарных расстояний между объектами – в общем случае не имеет решения, а приближённое решение не единственно и может иметь существенную погрешность. Эта задача решается методами многомерного шкалирования.

Таким образом, постановка задачи кластеризации по матрице расстояний является более общей. С другой стороны, при наличии признаковых описаний часто удаётся строить более эффективные методы кластеризации.

Цели кластеризации:

- Понимание данных путём выявления кластерной структуры. Разбиение выборки на группы схожих объектов позволяет упростить дальнейшую обработку данных и принятия решений, применяя к каждому кластеру свой метод анализа (стратегия «разделяй и властвуй»).
- Сжатие данных. Если исходная выборка избыточно большая, то можно сократить её, оставив по одному наиболее типичному представителю от каждого кластера.
- Обнаружение новизны (novelty detection). Выделяются нетипичные объекты, которые не удаётся присоединить ни к одному из кластеров.

В первом случае число кластеров стараются сделать поменьше. Во втором случае важнее обеспечить высокую (или фиксированную) степень сходства объектов внутри каждого кластера, а кластеров может быть сколько угодно. В третьем случае наибольший интерес представляют отдельные объекты, не вписывающиеся ни в один из кластеров.

Во всех этих случаях может применяться иерархическая кластеризация, когда крупные кластеры дробятся на более мелкие, те в свою очередь дробятся ещё мельче, и т. д. Такие задачи называются задачами **таксономии**.

Результатом таксономии является древообразная иерархическая структура. При этом каждый объект характеризуется перечислением всех кластеров, которым он принадлежит, обычно от крупного к мелкому. Визуально таксономия представляется в виде графика, называемого **дендрограммой**.

Классическим примером таксономии на основе сходства является биноминальная номенклатура живых существ, предложенная Карлом Линнеем в середине XVIII века. Аналогичные систематизации строятся во многих областях знания, чтобы упорядочить информацию о большом количестве объектов.

Функции расстояния:

- Метрика Хэмминга;
- Евклидова метрика;
- Взвешенная евклидова метрика;
- Метрика Минковского.

Методы кластеризации:

- Графовые алгоритмы кластеризации;
- Статистические алгоритмы кластеризации;
- Алгоритм k-средних (k-means);
- EM-алгоритм;
- Алгоритм ФОРЕЛЬ;
- Иерархическая кластеризация или таксономия;
- Нейронная сеть Кохонена;
- Ансамбль кластеризаторов.

Формальная постановка задачи кластеризации

Пусть X – множество объектов, Y - множество номеров (имён, меток) кластеров. Задана функция расстояния между объектами $\rho(x,x')$. Имеется конечная обучающая выборка объектов $X^m = \{x_1,\dots,x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ, а объекты разных кластеров существенно отличались. При этом $x_i \in X^m$ каждому объекту приписывается номер кластера y_i.

Алгоритм кластеризации – это функция $a: X \to Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$. Множество Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

Кластеризация (обучение без учителя) отличается от классификации (обучения с учителем) тем, что метки исходных объектов y_i изначально не заданы, и даже может быть неизвестно само множество Y.

Решение задачи кластеризации принципиально неоднозначно, и тому есть несколько причин:

- Не существует однозначно наилучшего критерия качества кластеризации. Известен целый ряд эвристических критериев, а также ряд алгоритмов, не имеющих чётко выраженного критерия, но осуществляющих достаточно разумную кластеризацию «по построению». Все они могут давать разные результаты;
- Число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным критерием;
- Результат кластеризации существенно зависит от метрики, выбор которой, как правило, также субъективен и определяется экспертом.

Пример решения задачи кластеризации: сгруппировать цветы по сходству

- Первоначально необходимо загрузить файл с данными в R-Studio

```
iris <- read.csv("Iris.csv")
```

- Преобразуем категориальный признак Species в фактор

```
iris$Species <- factor(iris$Species)
```

- Просмотрим структуру данных с помощью функций head()

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

- Загрузим библиотеку для цветной визуализации данных

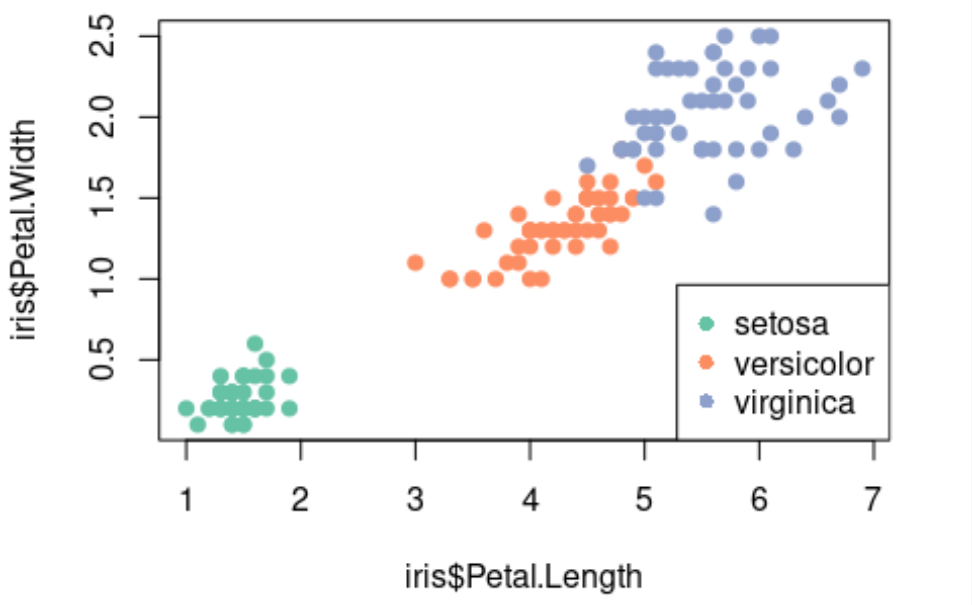
```
library(RColorBrewer)
```

- Создадим цветную палитру

```
palette <- brewer.pal(3, "Set2")
```

- Построим диаграмму рассеивания зависимости длины лепестка от ширины

```
plot(
  x = iris$Petal.Length,
  y = iris$Petal.Width,
  col = palette[as.numeric(iris$Species)],
  pch = 19)
legend(
  x = "bottomright",
  legend = levels(iris$Species),
  col = palette,
  pch = 16
  )
```



Как мы можем видеть, **Setosa** (Ирис щетинистый) может быть кластеризирован проще всего. Между тем, между **Versicolor** (Ирис разноцветный) и **Virginica** (Ирис виргинский) существует некоторая неопределенность.

Кластеризация методом К-средних

- Установим значение для воспроизведения одинаковой случайной последовательности каждый раз

```
set.seed(42)
```

- Проведем кластеризацию, используя функцию kmeans(), для этого необходимо задать количество кластеров (center)

```
irisCluster <- kmeans(iris[,1:4], center=3, nstart=20)
irisCluster

## K-means clustering with 3 clusters of sizes 50, 62, 38
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.006000    3.428000     1.462000    0.246000
## 2     5.901613    2.748387     4.393548    1.433871
## 3     6.850000    3.073684     5.742105    2.071053
##
## 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
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3
## [112] 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3
## [149] 3 2
##
## Within cluster sum of squares by cluster:
## [1] 15.15100 39.82097 23.87947
##  (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot
.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

- Сравним полученные результаты кластеризации с исходными данными (50 setosa; 50 versicolor; 50 virginica)

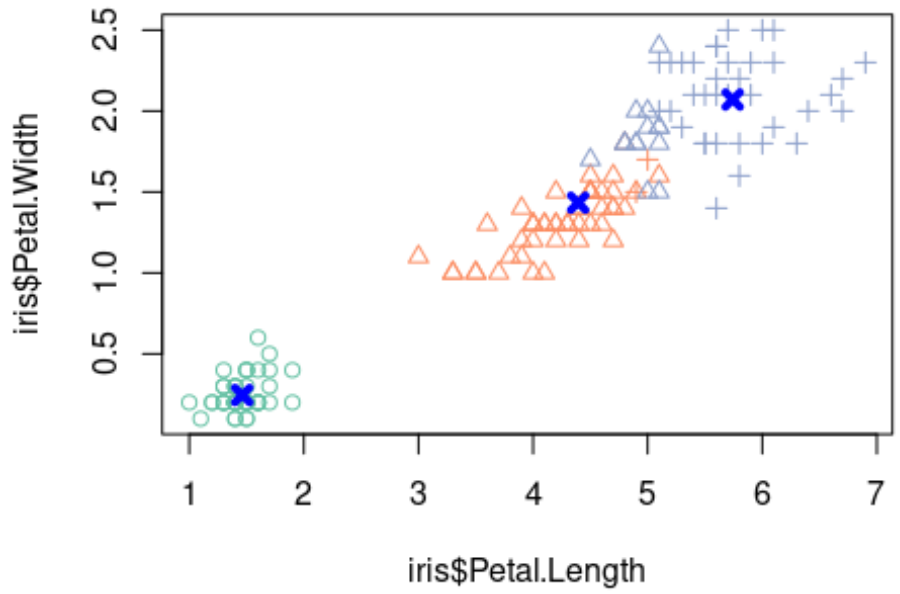
```
table(irisCluster$cluster, iris$Species)

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

- Отрисуем каждый кластер, как отдельную форму и отобразим центры кластеров

```
plot(x = iris$Petal.Length,
     y = iris$Petal.Width,
     col =  palette[as.numeric(iris$Species)],
```

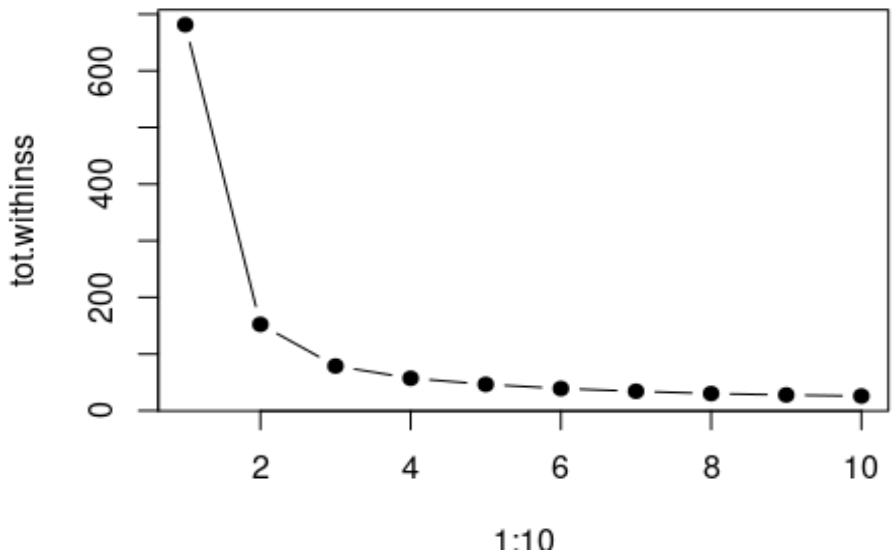
```
     pch = irisCluster$cluster)
points(x = irisCluster$centers[, "Petal.Length"],
       y = irisCluster$centers[, "Petal.Width"],
       pch = 4,
       lwd = 4,
       col = "blue")
```



- Отрисуем полученные кластеры с помощью clusplot()

Если бы мы не знали заранее точное количество кластеров, то для определения количества кластеров можно использовать метод локтя (elbow method). Суть данного метода в том, что мы обучаем модель, используя несколько вариантов количества кластеров, измеряем сумму квадратов внутрикластерных расстояний (tot.withinss) и выбираем тот вариант, при котором данное расстояние перестанет существенно уменьшаться.

```
tot.withinss <- vector(mode="character", length=10)
for (i in 1:10){
  irisCluster <- kmeans(iris[,1:4], center=i, nstart=20)
  tot.withinss[i] <- irisCluster$tot.withinss
}
plot(1:10, tot.withinss, type="b", pch=19)
```



Как можем видеть, после того как количество кластеров достигает трех, сумма квадратов внутрикластерных расстояний перестает существенно уменьшаться. Следовательно, три кластера и будет оптимальным значением.

Иерархическая кластеризация

- Вычислим матрицу расстояний

```
D <- dist(iris[ ,1:4])
```

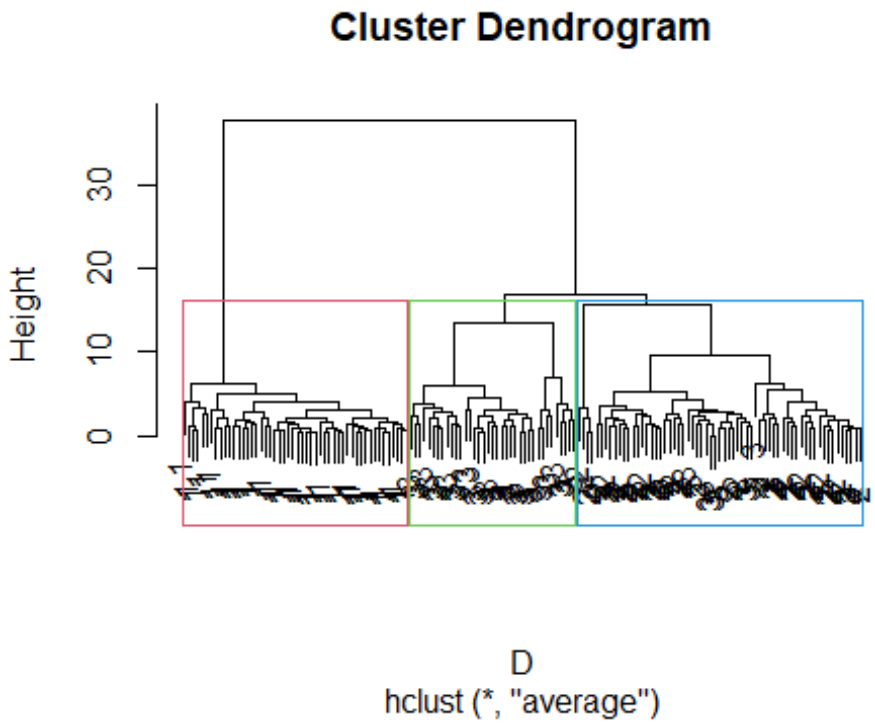
- Сгруппируем объекты в иерархическое дерево (дендрограмму)

```
hclusters <- hclust(D, method = "average")
```

Для выделения значимых кластеров можно задать фиксированное количество кластеров или задать некоторое пороговое значение T меры расстояний сходства. Число значимых кластеров определяется количеством пересечений линии порога T и связей иерархического дерева.

- Визуализируем дендрограмму кластеров и определим количество кластеров (через пороговое значение и методом задания фиксированного числа кластеров)

```
plot(x = hclusters, labels = as.numeric(iris$Species))
#rect.hclust(hclusters, h =16,border = 2:4) # Задание порогового значения
rect.hclust(hclusters, k=3, border = 2:4) #метод задания фиксированного числа кластеров
```

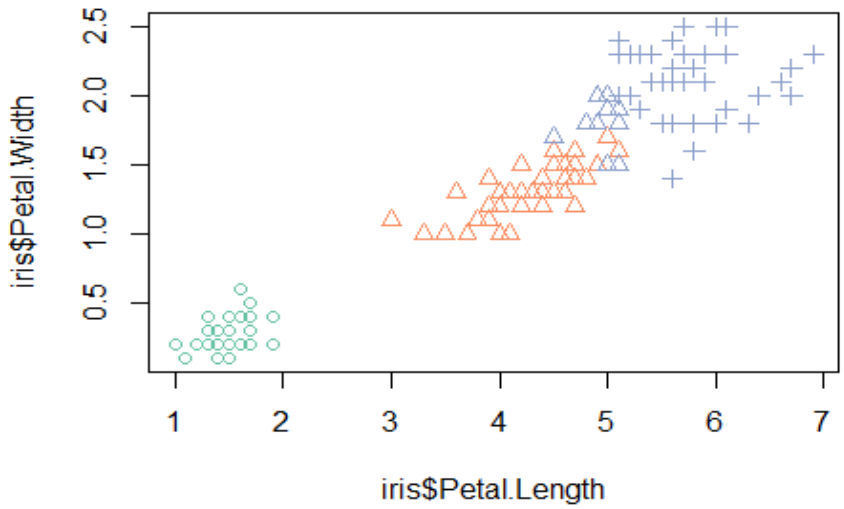


- Получим кластеры как вектор

```
cuts <- cutree(tree = hclusters, k = 3)
```

- Визуализируем полученные кластеры

```
plot(x = iris$Petal.Length,
     y = iris$Petal.Width,
     col = palette[as.numeric(iris$Species)],
     pch = cuts)
```



Задание: разделить клиентов на несколько групп

Создать новый проект: File → New Project → New Directory → New Project → Вводим имя новой директории R_Lab3 и указываем путь к папке, где будут храниться лабораторные работы. → Create project.

При выполнении данных команд у вас должен создаться новый проект. Далее создадим новый R Script, где непосредственно будут выполняться лабораторная работа. File → New File → R Script или комбинацией клавиш Ctrl + Shift + N. Для выполнения команды следует нажать на кнопку Run (Ctrl + Enter).

Результаты выполнения будут отражаться в поле **Console**, **Enviroment** или **Plots**.

1. Загрузить файл с данными **Customers.csv**;
2. Просмотреть структуру данных с помощью функций head();
3. Загрузить бибилиотеку **dplyr**
4. Преобразовать категориальный признак в фактор;
5. Удалить столбец **customer_id** из исходного дата фрейма и преобразовать столбец **Gender** в числовые значения.
   Например**, df %>% mutate(gender = as.numeric(gender)) %>% select(-customer_id)**
6. Просмотреть структуру данных после преобразования. **Почему данные необходимо преобразовать в числовые значения**?
7. Методом локтя определить оптимальное количество кластеров;
8. Для функции set.seed() установить аргумент, равный 42. **Объяснить для чего используется данная функция в работе**;
9. Провести кластеризацию методом k-средних;
10. Построить диаграмму рассеяния **spending_score от age**. В качестве параметров **col** и **pch** передать dfCluster$cluster;
11. Отобразить центры полученных кластеров;
12. Вычислить матрицу расстояний и провести иерархическую кластеризацию;
13. Визуализировать дендрограмму и отобразить найденные кластеры. **Объяснить выбранное количество кластеров**.
14. Представить кластеры как вектор;
15. Создать диаграмму рассеивание **spending_score от age**, окрашенную в цвет кластеров из пункта 12. В качестве параметров **col** и **pch** передать величину, реализованную в пункте 14.

Содержание отчёта

Файл отчёта должен содержать полный код программы с описанием производимых действий и ответами на вопросы.