Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

И.И. Ерохин

КЛАСТЕРНЫЙ АНАЛИЗ.

Методические указания к выполнению лабораторной работы

по курсу «Технологии анализа данных»

Калуга – 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Технологии анализа данных» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 4-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткую теоретическую часть, описывающую область применения кластерного анализа.

Методические указания составлены в расчете на всестороннее ознакомление студентов с основами работы кластерного анализа.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является формирование практических навыков решения задач кластерного анализа.

Основными задачами выполнения лабораторной работы являются:

1. Классификация объектов.

Результатами работы являются:

1. Графики, демонстрирующие классификацию объектов.
2. Подготовленный отчет.

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Общие сведения

Классификация — одна из важнейших задач, встречающихся при анализе данных. В зависимости от постановки можно различать следующие задачи: кластеризацию (классификацию в отсутствии обучающей выборки) и, собственно, классификацию (при наличии обучающей выборки), когда данные необходимо соотнести с уже известными классами.

Первая задача возникает, как правило, когда исследуются новые объекты или явления; в этом случае кластеризация позволяет выделить однородные группы объектов, и далее, планировать последующие изыскания. Классификация при наличии обучающей выборки предполагает, основываясь на уже имеющихся данных и их соответствии известным классам, определение класса для вновь поступающих данных. Распознавание текста, речи, аутентификация\ авторизация по отпечатку пальца — представляют задачи классификации этого типа.

Рассмотрим более подробно задачу кластеризации. Она предполагает разбиение массива данных на однородные группы, исходя из представлений о близости элементов между собой. При этом число кластеров может постулироваться заранее, или определяться в процессе кластерного анализа. В зависимости от того, каким образом определяются расстояния между кластеризуемыми объектами, результаты классификации будут различны. Если данные представляют собой наборы количественных признаков, вполне естественно рассматривать их как точки многомерного факторного пространства, а в качестве расстояния, например, использовать евклидову метрику. В случае трехмерного факторного пространства такой подход будет иметь ясную геометрическую интерпретацию и связь с реальным трехмерным миром (каждая точка данных - некоторая точка в трехмерном пространстве; расстояние между данными - это расстояние между такими точками).

Следует помнить, что количественные признаки могут иметь различную физическую интерпретацию: один признак может отождествляться с длиной, другой и массой объекта, поэтому понимание "расстояния" между элементами в этом случае весьма условно. Однако, если выбранные признаки полно характеризуют исследуемые объекты, то даже такие "бессмысленные" расстояния (несмотря на различие единиц измерения) имеют важное свойство: элементы, имеющие близкие значения количественных признаков, как следствие, характеризуются малыми расстояниями между друг другом, вполне ожидаемо похожи друг на друга. Таким образом, расстояния, построенные даже на базе разнородных признаков, выполняют свою главную роль — характеризуют сходство объектов.

В задаче кластеризации признаки не всегда могут иметь количественный характер. Например, наряду с различными морфометрическими характеристиками исследуемой совокупности объектов, может отмечаться наличие или отсутствие какой-либо их особенности. Два объекта могут иметь практически идентичные морфометрические признаки, но принципиально различаться ввиду этой особенности. Есть соблазн закодировать наличие особенности числом "1", а ее отсутствие, например, "0", и далее считать, что имеются только количественных признаки. Однако, в этом случае наличие или отсутствие особенности может легко маскироваться незначительными изменениями в морфометрических признаках. Грубо говоря, если ширина двух объектов отличается на 1 см, то это может быть эквивалентным тому, что объекты имеют или не имеют важную особенность. А это может быть критичным для помещения объекта в тот или иной кластер.

Метод k-means (k-средних)

Наиболее популярным алгоритмом кластеризации данных является метод k-средних. Это итеративный алгоритм кластеризации, основанный на минимизации суммарных квадратичных отклонений точек кластеров от центроидов (средних координат) этих кластеров.

Данный алгоритм очень легко представляется в виде простого псевдокода:

1. Выбрать количество кластеров k, которое нам кажется оптимальным для наших данных.
2. Высыпать случайным образом в пространство наших данных k точек (центроидов).
3. Для каждой точки нашего набора данных посчитать, к какому центроиду она ближе.
4. Переместить каждый центроид в центр выборки, которую мы отнесли к этому центроиду.
5. Повторять последние два шага фиксированное число раз, либо до тех пор, пока центроиды не "сойдутся" (обычно это значит, что их смещение относительно предыдущего положения не превышает какого-то заранее заданного небольшого значения).

Выбор числа кластеров для kMeans

В отличие от задачи классификации или регресии, в случае кластеризации сложнее выбрать критерий, с помощью которого было бы просто представить задачу кластеризации как задачу оптимизации. В случае kMeans распространен вот такой критерий – сумма квадратов расстояний от точек до центроидов кластеров, к которым они относятся.

$$J(C) = \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k||^2 \to \min_C,$$

здесь C – множество кластеров мощности K, μ_k– центроид кластера C_k .

Понятно, что здравый смысл в этом есть: мы хотим, чтобы точки располагались кучно возле центров своих кластеров. Но вот незадача: минимум такого функционала будет достигаться тогда, когда кластеров столько же, сколько и точек (то есть каждая точка – это кластер из одного элемента). Для решения этого вопроса (выбора числа кластеров) часто пользуются такой эвристикой: выбирают то число кластеров, начиная с которого описанный функционал $J(C)$ падает "уже не так быстро". Или более формально:

$$D(k) = \frac{|J(C_k) - J(C_{k+1})|}{|J(C_{k-1}) - J(C_k)|} \to \min_k$$

Рассмотрим пример.

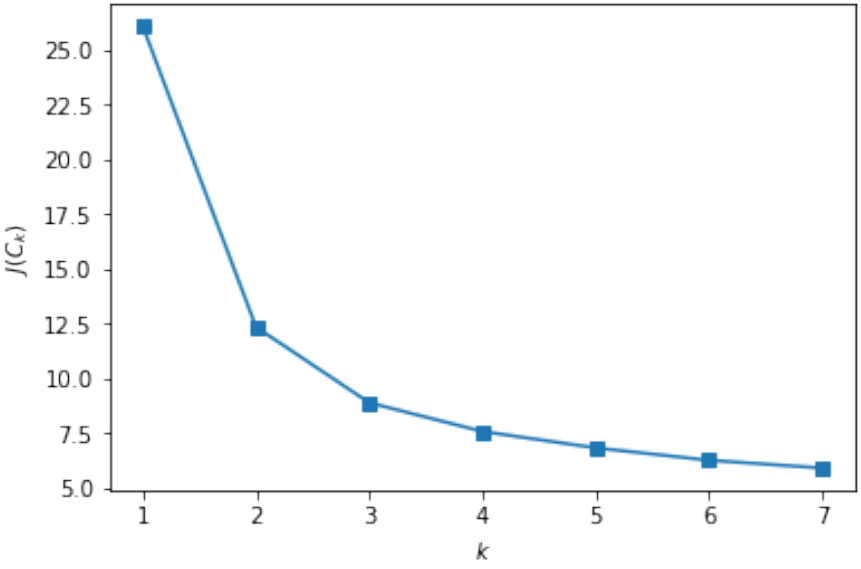

Рис. 1. Выбор числа кластеров

Видим, что $J(C_k)$ падает сильно при увеличении числа кластеров с 1 до 2 и с 2 до 3 и уже не так сильно – при изменении k с 3 до 4. Значит, в данной задаче оптимально задать 3 кластера.

Иерархическая кластеризация

Иерархическая кластеризация — совокупность алгоритмов упорядочивания данных, направленных на создание иерархии (дерева) вложенных кластеров. Выделяют два класса методов иерархической кластеризации:

- *Агломеративные методы* (англ. *agglomerative*): новые кластеры создаются путем объединения более мелких кластеров и, таким образом, дерево создается от листьев к стволу;
- *Дивизивные* или *дивизионные методы* (англ. *divisive*): новые кластеры создаются путем деления более крупных кластеров на более мелкие и, таким образом, дерево создается от ствола к листьям.

В данной лабораторной работе будет рассмотрен только агломеративная кластеризация.

Агломеративная кластеризация - наверное самый простой и понятный алгоритм кластеризации без фиксированного числа кластеров. Алгоритм очень прост:

1. Начинаем с того, что высыпаем на каждую точку свой кластер
2. Сортируем попарные расстояния между центрами кластеров по возрастанию
3. Берём пару ближайших кластеров, склеиваем их в один и пересчитываем центр кластера
4. Повторяем п. 2 и 3 до тех пор, пока все данные не склеятся в один кластер

Сам процесс поиска ближайших кластеров может происходить с использованием разных методов объединения точек:

1. Single linkage — минимум попарных расстояний между точками из двух кластеров

$$d(C_i, C_j) = min_{x_i \in C_i, x_j \in C_j} ||x_i - x_j||$$

2. Complete linkage — максимум попарных расстояний между точками из двух кластеров

$$d(C_i, C_j) = max_{x_i \in C_i, x_j \in C_j} ||x_i - x_j||$$

3. Average linkage — среднее попарных расстояний между точками из двух кластеров

$$d(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x_i \in C_i} \sum_{x_j \in C_j} ||x_i - x_j||$$

4. Centroid linkage — расстояние между центроидами двух кластеров

$$d(C_i, C_j) = ||\mu_i - \mu_j||$$

Выгода первых трёх подходов по сравнению с четвёртым в том, что для них не нужно будет пересчитывать расстояния каждый раз после склеивания, что сильно снижает вычислительную сложность алгоритма.

Результат иерархической кластеризации может быть представлен с помощью дендрограммы.

Сравнение метода k-средних с иерархической кластеризацией данных

- Иерархическая кластеризация хуже подходит для кластеризации больших объемов данных в сравнении с методом k-средних. Это объясняется тем, что временная

сложность алгоритма линейна для метода k-средних (O(n)) и квадратична для метода иерархической кластеризации ($O(n^2)$)

- В кластеризации при помощи метода k-средних алгоритм начинает построение с произвольного выбора начальных точек, поэтому, результаты, генерируемые при многократном запуске алгоритма, могут отличаться. В то же время в случае иерархической кластеризации результаты воспроизводимы.
- Из центроидной геометрии построения метода k-средних следует, что метод хорошо работает, когда форма кластеров является гиперсферической (например, круг в 2D или сфера в 3D).
- Метод k-средних более чувствителен к зашумленным данным, чем иерархический метод.

ОБРАЗЕЦ ВЫПОЛНЕНИЯ ЗАДАНИЯ

```
# Импортируем библиотеки для Kmeans
from sklearn import datasets
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import numpy as np
# Импортируем библиотеки для иерархической
кластеризации
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt
import pandas as pd

# Загружаем набор данных
iris_df = datasets.load_iris()
# Методы, доступные для набора данных
#print(dir(iris_df))
# Признаки
#print(iris_df.feature_names)
# Метки
#print(iris_df.target)
# Имена меток
#print(iris_df.target_names)
# Получение только числовых значений
# numeric_cols = iris_df._get_numeric_data().dropna(axis=1)

#Выбор числа кластеров для kMeans(Рис 1)
inertia = []
for k in range(1, 8):
    kmeans = KMeans(n_clusters=k,
random_state=1).fit(iris_df.data)
    inertia.append(np.sqrt(kmeans.inertia_))

plt.plot(range(1, 8), inertia, marker='s');
plt.xlabel('$k$')
plt.ylabel('$J(C_k)$');

#-----------K-means----------------
# Описываем модель
model = KMeans(n_clusters=3)
# Проводим моделирование и предсказание на всем наборе
данных к какому кластеру относится элемент
y_predict = model.fit_predict(iris_df.data)
```

```
# Центры кластеров
model.cluster_centers_


# Разделение набора данных
x_axis = iris_df.data[:, 0]  # Sepal Length
y_axis = iris_df.data[:, 1]  # Sepal Width

# Построение
plt.xlabel(iris_df.feature_names[0])
plt.ylabel(iris_df.feature_names[1])
plt.scatter(x_axis, y_axis, c=y_predict)
plt.scatter(model.cluster_centers_[:,0],
model.cluster_centers_[:,1], marker='*', c='r')
plt.show()

#-----------Иерархическая кластеризация----------------
# Создаем датафрейм
iris_df = pd.read_csv('iris_df.csv')
iris_df.columns = ['X1', 'X2', 'X3', 'X4', 'Y']
#print(iris_df)
# Исключаем информацию об образцах ириса, сохраняем для
дальнейшего использования
varieties = list(iris_df.pop('Y'))
samples = iris_df.values
#print(samples)
# Реализация иерархической кластеризации при помощи
функции linkage
mergings = linkage(samples, method='complete')
# Строим дендрограмму, указав параметры удобные для
отображения
dendrogram(mergings,
           labels=varieties,
           leaf_rotation=90,
           leaf_font_size=6,
           color_threshold=3.5
           )

plt.show()
```

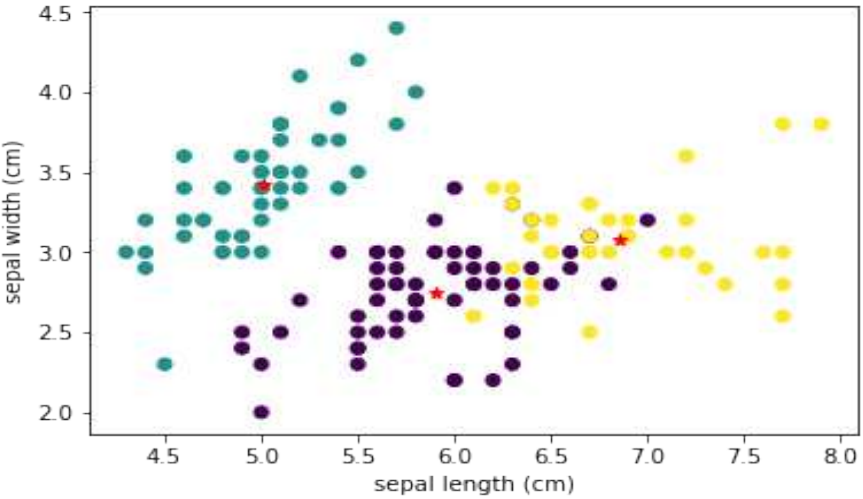

Рис. 2. Кластеризация методом k-means

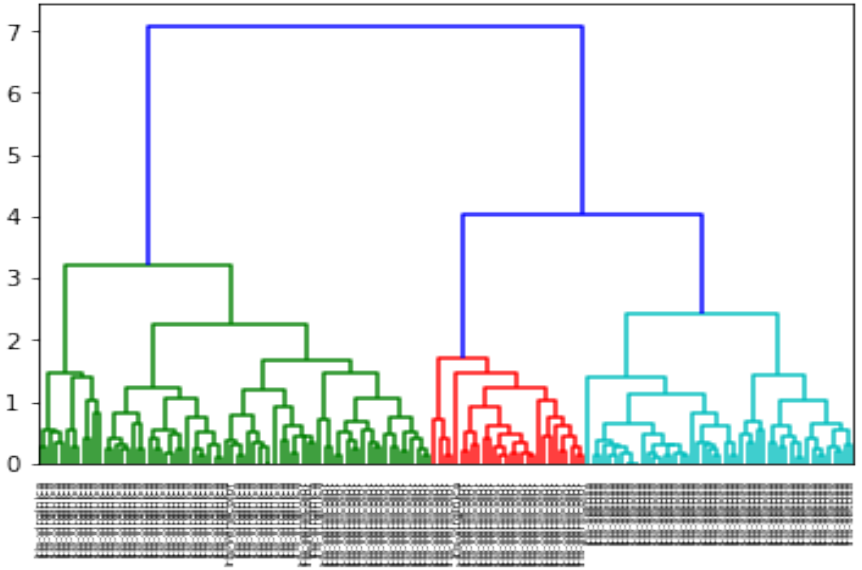

Рис. 3. Иерархическая кластеризация

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Провести классификацию объектов согласно варианту, полученному у преподавателя.

ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

В качестве результата работы необходимо построить график классификации объектов. По завершении готовится отчёт.

ВАРИАНТЫ ЗАДАНИЙ

Вариант 1

Считать данные из файла iris_df.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить график кластерзации для Petal Length, Petal width. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом single. Labels = вид ириса. Построить дендрограмму.

Вариант 2

Считать данные из файла seeds-less-rows.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить график кластерзации для area, perimeter. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом complete. Labels = grain_variety. Построить дендрограмму.

Вариант 3

Вручную сгенерировать три кластера точек и отобразить их. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом average. Построить дендрограмму.

Вариант 4

Считать данные из файла nba_2013.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить график кластерзации для pts, ast. С помощью k-means построить их

центроиды. Провести иерархическую кластеризацию методом average. Labels = pts. Построить дендрограмму.

Вариант 5

Считать данные из файла iris_df.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить график кластерзации для Sepal Length, Petal width. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом average. Labels = Sepal Length. Построить дендрограмму.

Вариант 6

Считать данные из файла seeds-less-rows.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить график кластерзации для compactness, asymmetry_coefficient С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом single. Labels = area. Построить дендрограмму.

Вариант 7

Вручную сгенерировать четыре кластера точек и отобразить их. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом complete. Построить дендрограмму.

Вариант 8

Считать данные из файла nba_2013.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить график кластерзации для age, mp. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом single. Labels = age. Построить дендрограмму.

Вариант 9

Считать данные из файла iris_df.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить

график кластерзации для Petal Length, Sepal width. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом complete. Labels = Petal width. Построить дендрограмму.

Вариант 10

Считать данные из файла seeds-less-rows.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить график кластерзации для length, width. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом average. Labels = length. Построить дендрограмму.

Вариант 11

Вручную сгенерировать три кластера точек и отобразить их. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом single. Построить дендрограмму.

Вариант 12

Считать данные из файла nba_2013.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить график кластерзации для age, pts. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом single. Labels = age. Построить дендрограмму.

Вариант 13

Считать данные из файла iris_df.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров. Построить график кластерзации для Sepal length, Sepal width. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом single. Labels = вид ириса. Построить дендрограмму.

Вариант 14

Считать данные из файла seeds-less-rows.csv в структуру DataFrame. Построить график показывающий выбор числа кластеров.

Построить график кластерзации для groove_length, width. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом complete. Labels = width. Построить дендрограмму.

Вариант 15

Вручную сгенерировать четыре кластера точек и отобразить их. С помощью k-means построить их центроиды. Провести иерархическую кластеризацию методом average. Построить дендрограмму.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Дайте определение классификации.
2. Назовите отличие кластеризации от классификации.
3. Опишите шаги алгоритма k-means.
4. Как правильно выбрать число кластеров.
5. Дайте определение иерархической кластеризации.
6. Виды иерархической кластеризации.
7. Алгоритм агломеративной кластеризации.
8. Методов объединения точек для агломеративной кластеризации.
9. Сравнение методов k-means и иерархической кластеризации.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 1 занятие (2 академических часа: 1 час на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, описание процесса выполнения лабораторной работы, результаты выполнения работы, выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Маккинли, Уэс Python и анализ данных / Пер. с англ. Слинкин А.А. - М.: ДМК Пресс, 2015. - 482 с.:ил.
2. Грас, Дж. Data Science. Наука о данных с нуля / Пер. с англ. - СПБ.: БХВ -Петербург, 2017. - 336с.: ил.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

3. Henley, A.J. Learn Data Analysis with Python: Lessons in Coding / A.J. Henley, Dave Wolf ISBN 978-1-4842-3486-0

Электронные ресурсы:

4. Научная электронная библиотека http://eLIBRARY.RU
5. Электронно-библиотечная система http://e.lanbook.com