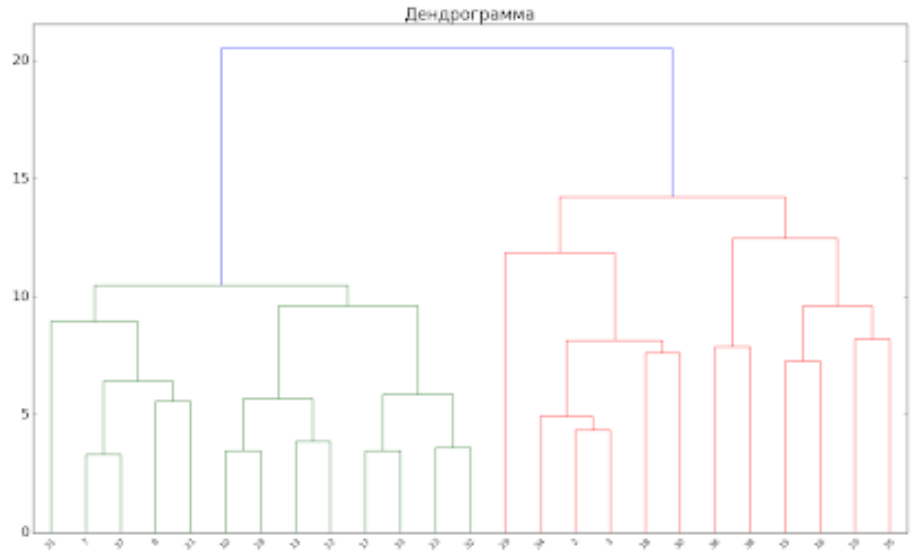
Лабораторная работа №5.1

Иерархический кластерный анализ



Алгоритм иерархического кластерного анализа

При иерархическом кластерном анализе заранее неизвестно число кластеров (групп, на которые разбивается набор объектов).

Шаг 0. Каждое наблюдение – отдельный кластер

Шаг 1. Два соседних кластера объединяются в один

и т.д.

Этот процесс продолжается до тех пор, пока не останутся только два кластера.

Алгоритм работает всегда! Даже если кластеров нет, они все равно найдутся.
Определить есть ли кластеры и сколько их позволяет *дендрограмма*.

А что делает аналитик?

1. Осуществляет отбор переменных
2. Определяет метод стандартизации (если это необходимо)
3. Определяет каким методом вычислять расстояние между кластерами
4. Определяет каким методом вычислять расстояние между объектами
5. Интерпретирует результат

Рассмотрим на примере

Задача сегментации потребителей безалкогольных напитков.

Компания провела опрос с целью выявить, какие напитки предпочитают респонденты. Опрошенные указывали, какие напитки из предложенного списка они пьют регулярно.

В списке присутствовали:

- Coca-Cola
- диетическая Coca-Cola
- Pepsi-Cola
- диетическая Pepsi-Cola
- 7-Up
- диетический 7-Up
- Спрайт
- минеральная вода

Этап 1. Отбор переменных

Какие переменные будут использоваться при анализе?

Очевидно, в решении данной задачи будут участвовать все переменные.

Но задачи и данные бывают разные.

Например, влияет ли цвет глаз покупателя на средний объем выпиваемого им пива?

Возможно, влияет. Чтобы ответить на этот вопрос наверняка, необходимо исследовать данные.

Критерием при отборе переменных для анализа является в первую очередь ясность интерпретации полученного результата, во вторую – интуиция исследователя.

Этап 2. Стандартизация данных

Стандартизация набора данных подразумевает масштабирование данных, при котором

1. Максимальное значение в выборке равно 1, минимальное равно 0 (или -1)

2. Среднее равно 0, выборочная дисперсия равна 1

Зачем нужна стандартизация данных?

Представим себе, что значения переменной *x* находятся в диапазоне от 100 до 700, а значения переменной *y* – в диапазоне от 0 до 1. Тогда, при расчете величины расстояния между точками, отражающими положение объектов в пространстве, имеющая большие значения, т.е. переменная *x*, будет практически полностью доминировать над переменной с малыми значениями, т.е. переменной *y*. Таким образом из-за неоднородности единиц измерения признаков становится невозможно корректно рассчитать расстояния между точками. Эта проблема решается при помощи предварительной *стандартизации* переменных.

Этап 3. Расстояние между объектами

Расстояние между объектами определяет их «похожесть»

- *Евклидово расстояние*

(x1, x2, x3)

(y1, y2, y3)

$$d_{xy} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

- Квадрат Евклидова расстояния

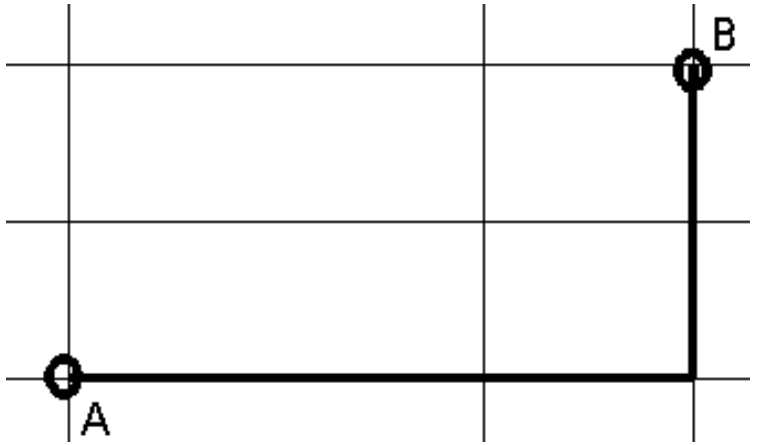
Формально, квадрат расстояния расстоянием не является. Но в некоторых случаях его использовать удобнее.

- Расстояние городских кварталов (Манхеттен, сити-блок)

$$X = (x_1, x_2, \ldots, x_k)$$

$$Y = (y_1, y_2, \ldots, y_k)$$

$$d_{XY} = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_k - y_k|$$



Что выбрать?

На слайде 6 представлены самые популярные меры расстояний. Есть и другие.

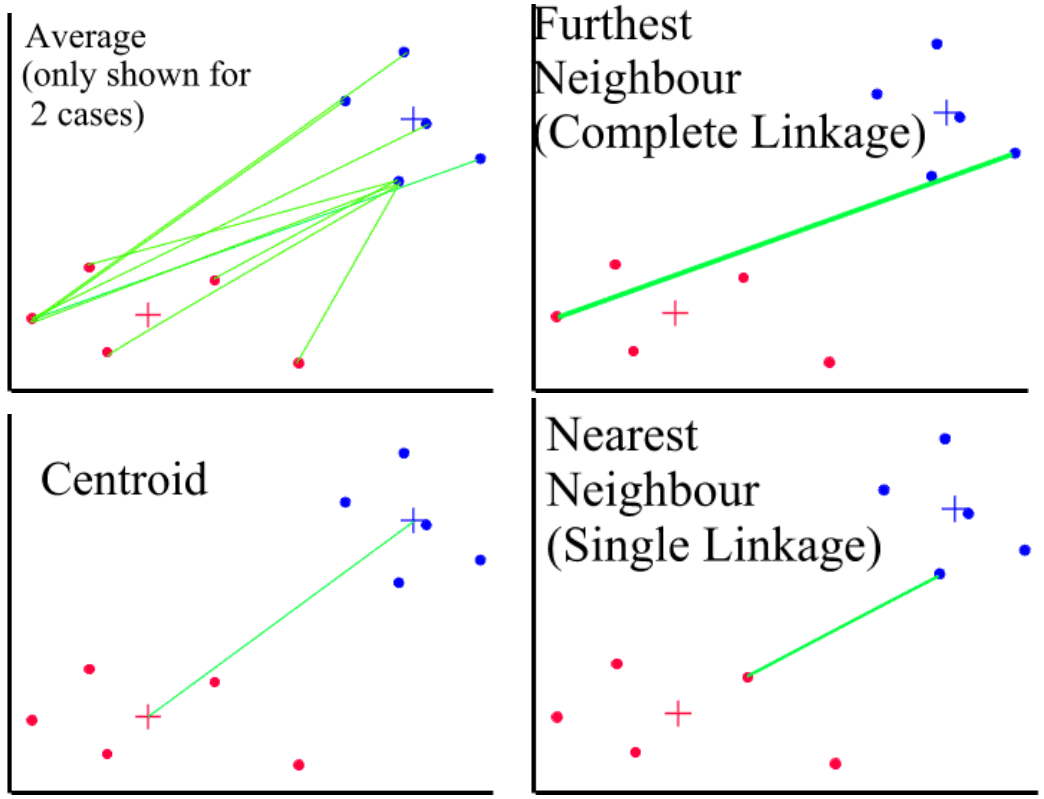
Евклидово расстояние – самое популярное.

Квадрат Евклидова расстояния применяется для придания большего веса более отдаленным друг от друга объектам.

Манхеттен чаще используется в случаях, когда в данных есть выбросы. Для этой меры влияние отдельных больших разностей (выбросов) уменьшается т.к. они не возводятся в квадрат.

Этап 4. Расстояние между кластерами

- Среднее невзвешенное расстояние (Average linkage clustering).
- Центроидный метод (Centroid Method).
- Метод дальнего соседа, максимального расстояния (Complete linkage clustering).
- Метод ближайшего соседа (Single linkage clustering).
- Метод Варда (Ward's method).



Метод Варда - в обоих кластерах для всех имеющихся наблюдений производится расчет средних значений отдельных переменных. Затем вычисляются квадраты евклидовых расстояний от отдельных наблюдений каждого кластера до этого кластерного среднего значения. Эти дистанции суммируются. Потом в один новый кластер объединяются те кластеры, которые дают наименьший прирост общей суммы дистанций.

Вернемся к задаче

- *Загрузите данные*

```
import pandas as pd
df = pd.read_csv("beverage_r.csv", sep=";", index_col='numb.obs')
```

```
df.head()
```

numb.obs	COKE	D_COKE	D_PEPSI	D_7UP	PEPSI	SPRITE	TAB	SEVENUP
1	1	0	0	0	1	1	0	1
2	1	0	0	0	1	0	0	0
3	1	0	0	0	1	0	0	0

- *Применим алгоритм иерархической кластеризации*

Вычисления производятся функцией linkage из пакета scipy. Аргументы функции: таблица данных, метод вычисления расстояний между кластерами (`ward` , `single` , `complete` , `average`) и метод вычисления расстояний между объектами (`euclidean` (default), `cityblock` (Manhattan), `hamming` , `cosine`).

```
# Импортируем необходимые функции
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

# Объект, в котором будет хранится информация о последовательном слиянии кластеров
link = linkage(df, 'ward', 'euclidean')
```

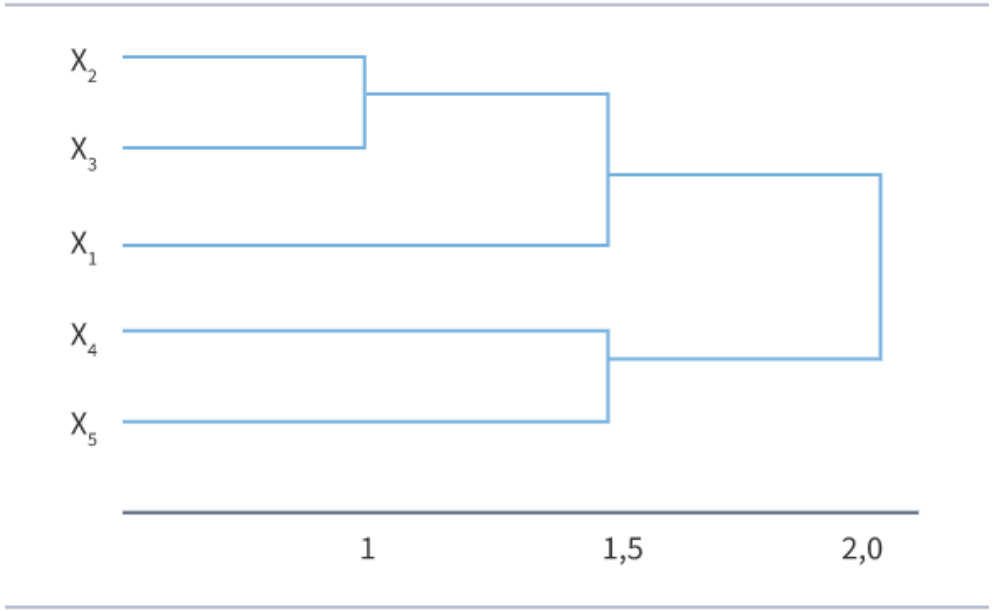
См. документацию

Объект `link` представляет собой матрицу `(n-1) x 4` , где `n` - число наблюдений. Каждая строка матрицы представляет собой результат слияния очередной пары кластеров с номерами `link[i, 0]` и `link[i, 1]` . Новому кластеру присваивается номер `n + i` . `link[i, 2]` содержит расстояние между объединяемыми кластерами, а `link[i, 3]` - размер нового кластера.

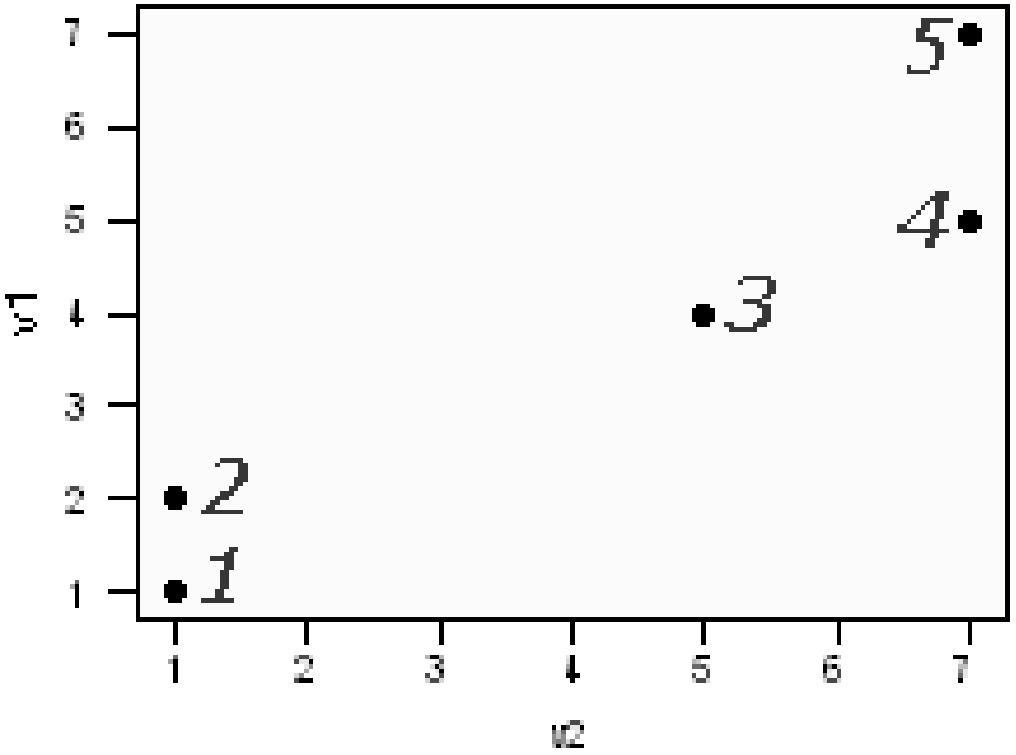
Построение дендрограммы

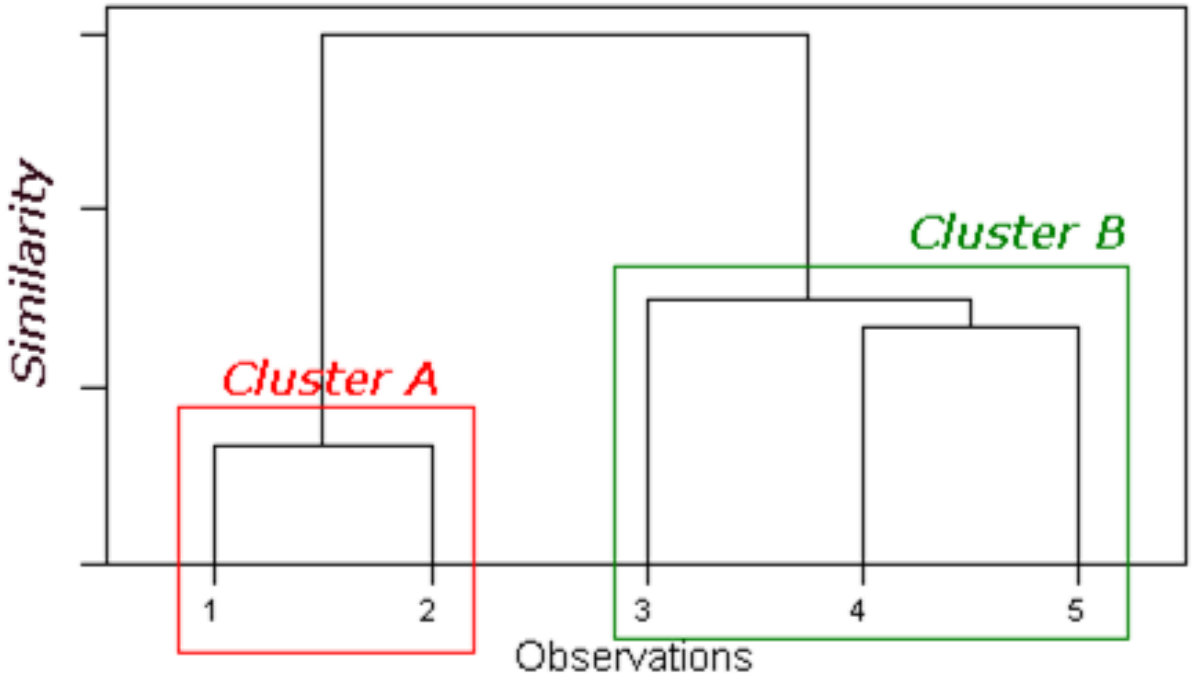
Дендрограмма – это визуализатор, используемый для представления результатов иерархической кластеризации. Она показывает степень близости отдельных объектов и кластеров, а также наглядно демонстрирует в графическом виде последовательность их объединения или разделения. Количество уровней дендрограммы соответствует числу шагов слияния или разделения кластеров.

В дендрограмме, представленной на рисунке, на первом шаге группируются объекты x_2 и x_3, образуя кластер (x_2,x_3) с минимальным расстоянием (например, Евклидовым) между объектами, примерно равным 1. Затем объекты x_4 и x_5 группируются в другой кластер (x_4,x_5) с расстоянием между ними, равным 1,5. Расстояние между кластерами (x_2,x_3) и (x_1) также оказывается равным 1,5, что позволяет сгруппировать их на том же уровне, что и (x_4,x_5). И наконец, два кластера (x_1,x_2,x_3) и (x_4,x_5) группируются на самом высоком уровне иерархии кластеров с расстоянием 2.



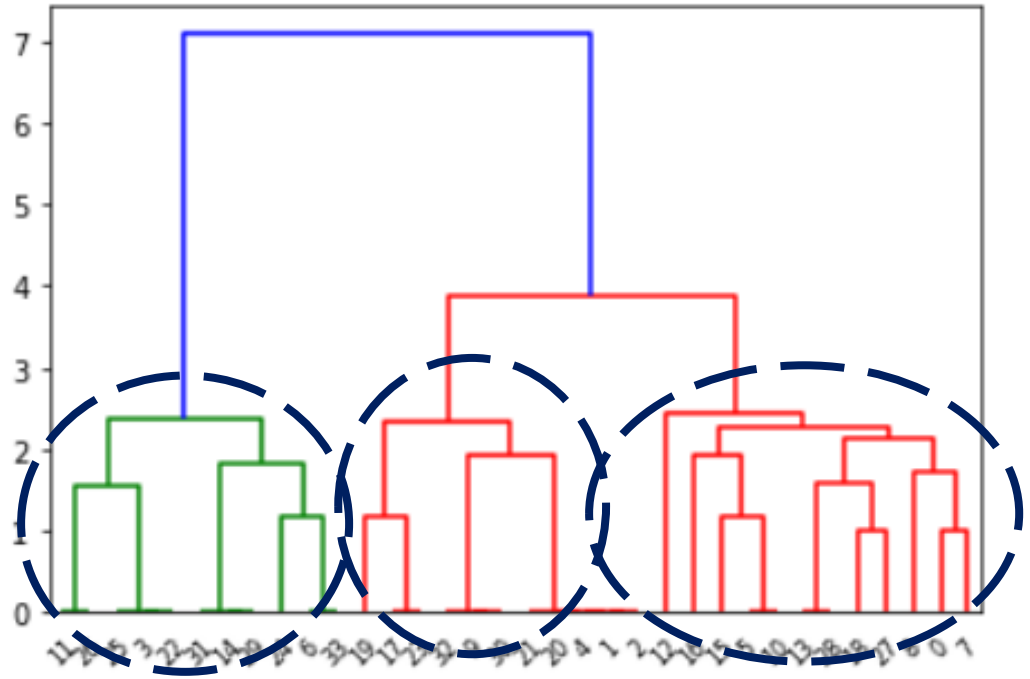
Пример


v1
v2
1
2
3
4
5
6
7
1
2
3
4
5


Similarity
Cluster A
Cluster B
1
2
3
4
5
Observations

Вернемся к задаче

```
# Функция для построения дендрограммы
dn = dendrogram(link)
```



Инорируем раскраску кластеров!

На самом деле здесь 3 кластера!

Добавим к данным колонку `cluster`, в которую запишем номер кластера, в которую попал объект. Это сделаем с помощью функции `fcluster`. В качестве первого аргумента передаём объект `linkage`, полученный выше. Третьим аргументом `criterion` передаём критерий для разбиения на кластеры (`distance` для остановки разбиения по расстоянию между кластерами и `maxclust` для разбиения по заданному числу кластеров). Вторым же аргументом является либо пороговое значение для расстояния между кластерами, либо количество кластеров, в зависимости от аргумента `criterion`.

```
df['cluster'] = fcluster(link, 3, criterion='distance')
```

Проанализируйте полученный результат:

- Сколько респондентов оказалось в каждой группе?
- Как распределились респонденты по группам?
- Какие можно сделать выводы?