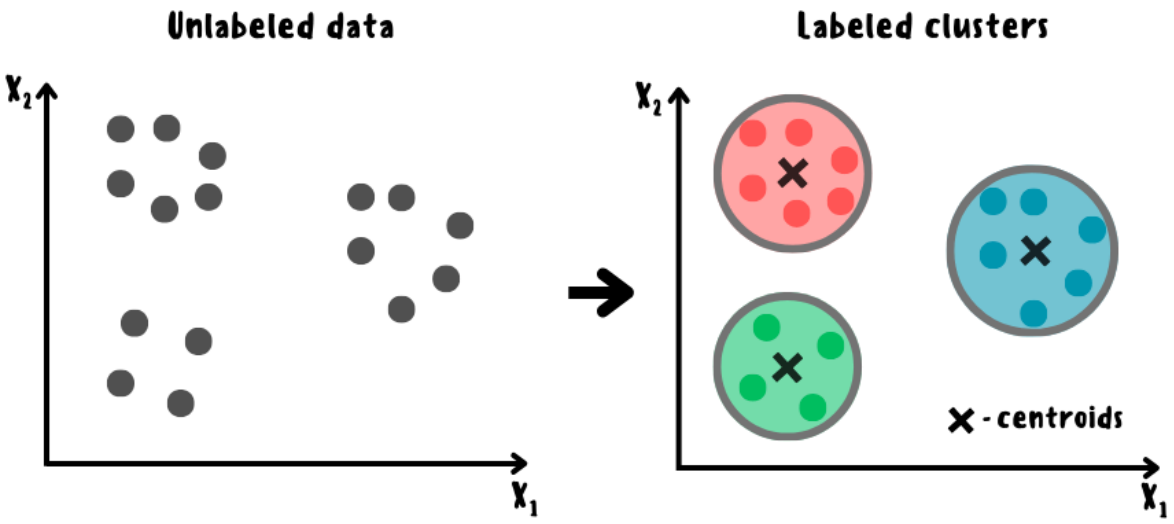
33. Метод K – средних в кластерном анализе. Положительные и отрицательные стороны этого метода.

1.4.1. Метод k-средних



Примечание: хорошо про k-means еще описано [здесь](#)

В отличие от иерархических процедур метод k-средних не требует вычисления и хранения матрицы расстояний между объектами. Алгоритм данного метода предполагает использование только исходных значения переменных и минимизирует дисперсию внутри каждого кластера. При этом возможны две модификации метода k-средних:

1. первая предполагает пересчет центра тяжести кластера после каждого изменения его состава;
2. вторая предполагает пересчет центра тяжести кластера после того, как будет завершен просмотр всех данных.

Пусть требуется классифицировать n объектов $N_1, N_2, \ldots, N_n$. Для применения метода k-средних заранее задается число кластеров k и случайно выбранные k объектов, которые будут служить эталонами E_m^0 $(m = 1, k)$, т.е. центрами кластеров. При этом выбор начальных условий играет важную роль: они влияют на длительность процесса классификации и на его результаты. Эталонные объекты могут отбираться случайным образом или задаваться исследователем исходя из каких-либо априорных соображений. Например, в качестве параметров исходного разбиения можно использовать результаты, полученные одним из методов иерархического кластерного анализа.

Затем определяются расстояния от остальных $n - k$ объектов до этих центров. Для этого используется одна из метрик, например, евклидово расстояние. Наблюдение N_i $(i = 1, n)$ относится к тому кластеру, расстояние до которого было минимальным. Если встречаются два или более минимальных расстояния, то i-й объект присоединяют к центру с наименьшим порядковым номером.

После этого для кластера, в котором увеличилось число наблюдений на единицу, рассчитывается новый центр тяжести (как среднее по каждому показателю) по всем включенным в кластер наблюдениям. На следующем шаге выбираем точку N_{i+1} и для нее повторяются все процедуры. Таким образом, через $(n - k)$ шагов все точки (объекты) совокупности окажутся отнесенными к одному из k кластеров.

Смысл описываемого алгоритма — в последовательном уточнении эталонных точек $E_m^v = \{e_1^v, e_2^v, \ldots, e_p^v\}$ (m - номер кластера, $m = 1, k$, v - номер итерации, $\nu = 0, 1, 2, \ldots; p$ - количество признаков, характеризующих объекты) с соответствующим пересчетом приписываемых им весов W_m^v (равных количеству объектов, входящих в данный кластер). При этом $W_m^0 = 1$.

При достаточно большом числе итераций или при достаточно больших объемах классифицируемых совокупностей дальнейший пересчет эталонных точек практически не приводит к их изменению, т.е. имеет место сходимость (в определенном смысле) E_m^v $(m = 1, k)$ к некоторому пределу при $\nu \to \infty$.

Если же в какой-то конкретной задаче исследователь не успел добраться до стадии практически устойчивых (по ν) значений эталонных точек, то пользуются одним из двух вспомогательных приемов:

1. «зацикливают» алгоритм, «прогоняя» его после рассмотрения последней точки $N_n = N_{k+(n-k)}$ снова через точку N_1, затем N_2 и т.д. Процедура продолжается до тех пор, пока последующее разбиение не даст такой же результат, как и предыдущее разбиение (Каждое разбиение включает в себя количество итераций, равное числу многомерных наблюдений.);

2. производят многократное повторение алгоритма, используя в качестве начальных эталонов E_m^0 различные комбинации из k точек исследуемой совокупности и выбирая для дальнейшего исследования наиболее повторяющиеся (в некотором смысле) финальные эталоны E_m^{n-k}.

После этого вычисляются центры тяжести полученных кластеров (по исходным данным) и строится окончательное разбиение: каждая многомерная точка относится к тому кластеру, центр которого ближе всего к этой точке. Полученное разбиение должно подтвердить разбиение, наблюдаемое при использовании одного из вспомогательных приемов получения практических устойчивых значений эталонных точек.

Алгоритм метода k-средних можно представить в виде последовательности процедур:

- **Шаг 1.** Выбор числа кластеров, на которые должна быть разбита совокупность, задание первоначального положения центров тяжести кластеров.
- **Шаг 2.** В соответствии с выбранными мерами сходства определение нового состава каждого кластера.
- **Шаг 3.** Пересчет центра тяжести кластера после каждого изменения его состава или после полного просмотра всех данных.
- **Шаг 4.** Процедуры 2 и 3 повторяются до тех пор, пока последующее разбиение не даст такой же состав кластеров, что и предыдущее разбиение.

Пример 3.

Пусть имеются шесть объектов N_i ($i = 1, 2, \ldots, 6$), которые необходимо разбить на три класса (кластера) при помощи метода k -средних. Каждый из объектов описывается тремя признаками X_1, X_2, X_3 (см. табл. 4):

№ объекта	X_1	X_2	X_3
1	0.10	10	5.0
2	0.80	14	2.0
3	0.40	12	3.0
4	0.18	11	4.0
5	0.25	13	3.2
6	0.67	15	2.4

Для упрощения числовых выкладок, так же, как и в предыдущих примерах данные не будем стандартизовать.

Решение. Нулевая итерация. В качестве эталонов возьмем первые три объекта ($k = 3$). Согласно выбранному правилу классификации запишем исходные значения эталонов и весов:

$$E_1^0 = N_1 = (0, 10; 10; 5, 0); \quad W_1^0 = 1;$$

$$E_2^0 = N_2 = (0, 80; 14; 2, 0); \quad W_2^0 = 1;$$

$$E_3^0 = N_3 = (0, 40; 12; 3, 0); \quad W_3^0 = 1;$$

Обозначения:

1. E_m^v **— эталон (центроид) кластера**
   - E — обозначение для **эталона** (центра кластера).
   - m — номер кластера ($m = 1, 2, \ldots, k$).
   - v — номер итерации алгоритма (например, $v = 0$ — начальная итерация).
   - Пример:

     $$E_1^0 = (0.10; 10; 5.0)$$

     — это **центр первого кластера на нулевой итерации**.
2. N_i **— объект (точка данных)**
   - N — обозначение для **объекта** (наблюдения).
   - i — номер объекта ($i = 1, 2, \ldots, n$).

3. W_m^v **— вес кластера**
   - W — обозначение для **веса** кластера, т.е. количества объектов, входящих в кластер.
   - m — номер кластера.
   - v — номер итерации.
   - Пример:

$$W_1^0 = 1$$

   — означает, что на нулевой итерации в **первом кластере** находится **1 объект** (сам эталон E_1^0).

1. На первом шаге берем четвертый объект и определяем его расстояние до каждого из эталонов по евклидовой метрике (4):

$$d_{41} = \sqrt{(0,18-0,1)^2 + (11-10)^2 + (4-5)^2} = 1,416;$$

$$d_{42} = \sqrt{(0,18-0,8)^2 + (11-14)^2 + (4-2)^2} = 3,658;$$

$$d_{43} = \sqrt{(0,18-0,4)^2 + (11-12)^2 + (4-3)^2} = 1,431;$$

Минимальным является расстояние до первого эталона. Значит объект N_4 присоединяем к 1 эталону. Эталон E_1 пересчитываем, а эталоны E_2, E_3 оставляем без изменений.

$$E_1^1 = \frac{W_1^0 E_1^0 + N_4}{W_1^1}; \quad W_1^1 = W_1^0 + 1 = 2;$$

$$E_2^1 = E_2^0; \quad W_2^1 = W_2^0;$$

$$E_3^1 = E_3^0; \quad W_3^1 = W_3^0;$$

$$E_1^1 = \left(\frac{0,1+0,18}{2}; \frac{10+11}{2}; \frac{5+4}{2}\right) = (0,14; 10,5; 4,5);$$

2. На втором шаге берем пятый объект и определяем его расстояние до каждого из эталонов по евклидовой метрике:

$$d_{51} = \sqrt{(0,25-0,14)^2 + (13-10,5)^2 + (3,2-4,5)^2} = 2,820;$$

$$d_{52} = \sqrt{(0,25-0,80)^2 + (13-14)^2 + (3,2-2,0)^2} = 1,656;$$

$$d_{53} = \sqrt{(0,25-0,40)^2 + (13-12)^2 + (3,2-3,0)^2} = 1,031;$$

Минимальным оказалось расстояние до третьего эталона. Присоединяем к нему пятый объект. При этом третий эталон будет пересчитан и его вес увеличится на единицу, а эталоны E_1, E_2 останутся прежними:

$$E_3^2 = \left(\frac{0,4+0,25}{2}; \frac{13+12}{2}; \frac{3,0+3,2}{2}\right) = (0,325; 12,5; 3,1);$$

$$W_3^2 = W_3^1 + 1 = 2;$$

$$E_1^2 = E_1^1; \quad W_1^2 = W_1^1;$$

$$E_2^2 = E_2^1; \quad W_2^2 = W_2^1;$$

3. На третьем шаге берем шестой объект и определяем его расстояние до каждого из эталонов по евклидовой метрике:

$$d_{61} = \sqrt{(0,67-0,14)^2 + (15-10,5)^2 + (2,4-4,5)^2} = 4,994;$$

$$d_{62} = \sqrt{(0,67-0,80)^2 + (15-14)^2 + (2,4-2,0)^2} = 1,085;$$

$$d_{63} = \sqrt{(0,67-0,325)^2 + (15-12,5)^2 + (2,4-3,1)^2} = 2,619;$$

Следовательно, шестой объект должен быть присоединен ко второму эталону. При этом второй эталон будет пересчитан и его вес увеличится на единицу:

$$E_2^3 = \left(\frac{0,8+0,67}{2}; \frac{14+15}{2}; \frac{2,0+2,4}{2}\right) = (0,735; 14,5; 2,2);$$

$$W_2^3 = W_2^2 + 1 = 2;$$

$$E_1^3 = E_1^2; \quad W_1^3 = W_1^2;$$

$$E_3^3 = E_3^2; \quad W_3^3 = W_3^2.$$

Для того чтобы добиться практически устойчивых (по v) значений эталонных точек, процесс «зацикливаем», т.е. по тому же правилу осуществляется просмотр и присоединение к соответствующему эталону каждого из шести объектов. При этом происходит пересчет эталонов и продолжается наращивание их весов. Проводимые расчеты (здесь не приводятся) показывают, что итерации 16-21 дают такой же результат, как и итерации 10-15.

В результате образованы три кластера $S_1 = \{1,4\}$, $S_2 = \{2,6\}$, $S_3 = \{3,5\}$. Вычислим центры тяжести полученных кластеров:

$$C_1 = (0.14; 10,5; 4,5); \quad \text{центр 1 кластера}$$

$$C_2 = (0.735; 14,5; 2,2); \quad \text{центр 2 кластера}$$

$$C_3 = (0.325; 12,5; 3,1); \quad \text{центр 3 кластера}$$

После этого строится окончательное разбиение: каждая многомерная точка относится к тому кластеру, центр которого ближе всех к центру кластера. Для нашего примера определяем поочередно расстояния от всех точек (N_1, N_2, N_3, N_4, N_5, N_6) до центров трех кластеров (см. табл.5).

Таблица 5

Расстояния до центров классов

Центры кластеров	Объекты
	N_1
C_1	0,708
C_2	5,338
C_3	3,148

Как видно из табл.5, подтверждается полученное разбиение на три кластера: $S_1 = \{N_1, N_4\}$; $S_2 = \{N_2, N_6\}$; $S_3 = \{N_3, N_5\}$. На этом решение задачи классификации методом k-средних завершается.

Влияние различных факторов на результат классификации. На характеристики кластерной структуры оказывают влияние:

- набор признаков, по которым осуществляется классификация;
- тип выбранного алгоритма классификации (например, иерархические и итеративные методы приводят к образованию различного числа кластеров);
- выбор меры сходства (например, если используется метод k-средних, то задаваемые начальные условия в значительной степени определяют конечный результат разбиения).

Простейший прием, позволяющий судить о качестве разбиения - сравнение средних значений признаков по кластерам со средними значениями, рассчитанными по всей совокупности объектов. Если различие существенно, то это является признаком хорошего разбиения. При этом неинформативные признаки могут быть исключены из рассмотрения на определенном этапе решения задачи разбиения объектов на классы.

Оценка результата классификации производится с помощью функционала (критерия) качества, который также называют *целевой функцией* - статистическое выражение, зависящее от параметров исследуемой системы. Наилучшим разбиением считается то, при котором целевая функция достигает экстремума (максимального или минимального значения). Чаще всего определенный алгоритм классификации обеспечивает получение экстремального значения определенного функционала (т.е. алгоритм и целевая функция связаны между собой). Например, использование метода Уорда приводит к получению кластеров с минимальной дисперсией.

Наиболее распространенными функционалами качества являются:

1. **Сумма квадратов расстояний до центров классов.** При использовании этого критерия стремятся получить такое разбиение совокупности объектов на k кластеров, при котором значение этого функционала было бы минимальным, т.е.:

$$F_1 = \sum_{l=1}^{k} \sum_{i \in S_l} d^2(X_i, \overline{X}_l) \to \min,$$

где l номер класса; $\overline{X}_l$ - центр l-го класса; X_i - вектор значений переменных для i-го объекта, входящего в кластер S_l; $d(X_i, \overline{X}_l)$ - расстояние между i-м объектом и центром класса S_l.

2. Сумма внутриклассовых расстояний между объектами.

Наилучшим следует считать такое разбиение, при котором достигается минимальное значение этого функционала, т.е. получены компактные кластеры.

$$F_2 = \sum_{l=1}^{k} \sum_{i,j \in S_l} d_{ij}^2 \to \min,$$

где d_{ij} - расстояние между i-м и j-м объектами, входящих в кластер S_l.

3. Сумма внутриклассовых дисперсий.

Оптимальным следует считать такое разбиение, при котором достигается минимальное значение этого функционала, т.е. получены однородные кластеры:

$$F_3 = \sum_{l=1}^{k} \sum_{j=1}^{p} \sigma_{lj}^2 \to \min,$$

где σ_{lj}^2 - дисперсия j-й переменной в кластере S_l.

На принципе минимизации внутриклассовой дисперсии основаны алгоритмы метода k-средних и метода Уорда.

4. Средние межклассовые расстояния.

Оптимальным следует считать такое разбиение, при котором достигается максимальное значение этого функционала, т.е. чем дальше кластеры друг от друга, тем лучше:

$$F_4 = \frac{\sum_{l=1}^{k-1} \sum_{q=l+1}^{k} \sum_{i \in S_l} \sum_{j \in S_q} d_{ij}}{\sum_{l=1}^{k-1} \sum_{q=l+1}^{k} n_l n_q} \to \max.$$

Положительные и отрицательные стороны метода K-Means

✅ Положительные стороны (Преимущества)

1. **Простота реализации и понимания**
   • Алгоритм интуитивно понятен и легко реализуется.
   • Логика (выбор центроидов → присвоение точек → пересчёт центров) проста для восприятия.
2. **Высокая скорость работы**
   • Вычислительная сложность линейна относительно числа объектов: $O(n \cdot k \cdot d \cdot i)$,
   где n — число объектов, k — число кластеров, d — размерность, i — число итераций.
   • Подходит для работы с большими наборами данных (big data).
3. **Масштабируемость**
   • Эффективно работает с большими объёмами данных.
   • Может быть распараллелен.
4. **Гарантированная сходимость**
   • Алгоритм гарантированно сходится к локальному минимуму внутриклассовой дисперсии. (алгоритм обязательно завершит работу (не зациклится бесконечно) и найдёт решение, которое лучше всех "соседних", но не обязательно лучшее из всех возможных)
   • Количество итераций обычно невелико.
5. **Универсальность применения**
   • Широко используется в различных областях:
   - Сегментация клиентов
   - Сжатие изображений
   - Анализ текстов
   - Биоинформатика
6. **Интерпретируемость результатов**
   • Каждый кластер представлен своим центроидом — «средним» объектом.
   • Легко анализировать характеристики кластеров через центроиды.

❌ Отрицательные стороны (Недостатки)

1. **Необходимость заранее задавать число кластеров (k)**
   • На практике оптимальное k часто неизвестно.
   • Неправильный выбор k приводит к плохому разбиению.
   • Требует дополнительных методов (например, elbow method, silhouette analysis).
2. **Чувствительность к начальным центроидам**
   • Разные начальные центры → разные конечные результаты.
   • Может сходиться к локальному, а не глобальному оптимуму.
   • Требует многократного запуска с разными инициализациями.
3. **Чувствительность к выбросам**
   • Выбросы сильно влияют на положение центроидов.
   • Могут создавать отдельные кластеры или искажать границы.
4. **Работа только с числовыми данными**
   • Требует вычисления средних значений → не работает с категориальными признаками.

```
# Пример данных о клиентах
клиенты = [
    {"возраст": 25, "город": "Москва", "пол": "М"},    # как вычислить среднее?
    {"возраст": 30, "город": "Питер",  "пол": "Ж"},
    {"возраст": 35, "город": "Москва", "пол": "М"}
]

# Что такое "средний город"?
# (Москва + Питер + Москва) / 3 = ?  ← Бессмысленно!

# Решение:
# 1) One-Hot Encoding для категориальных признаков
# Преобразуем "город" в числовой формат
город_Москва = [1, 0]  # если 1-й элемент = Москва
город_Питер  = [0, 1]  # если 2-й элемент = Питер

# 2) Использование специальных расстояний (например, расстояния Хэмминга)
# 3) Выбор других алгоритмов (K-Modes, K-Prototypes)
```

• Для смешанных данных нужна предварительная обработка.

5. **Предположение о сферической форме кластеров**
   • Алгоритм минимизирует сумму квадратов расстояний до центроидов.
   • Плохо работает с кластерами сложной формы (например, вытянутыми, изогнутыми).
   Пример плохого случая:

```
# Данные в форме двух концентрических окружностей
 # Внутренний круг: радиус 1, 100 точек
 # Внешний круг:  радиус 3, 100 точек

 # K-Means разделит на 2 кластера по диаметру!
 # Правильное решение: один кластер = внутренний круг, другой = внешний
```

6. **Равный вес всех признаков**
   • Не учитывает различную важность признаков.
   • Требует предварительной нормализации данных.
7. **Проблема «пустых» кластеров**
   • В процессе работы некоторые кластеры могут остаться без объектов.
   • Требует специальных стратегий переинициализации.
   Как возникает:
   - Начальный центр попал в "пустынную" область данных
   - Все точки оказались ближе к другим центрам
   - Кластер остаётся без точек
8. **Фиксированный размер кластеров**
   • Стремится создавать кластеры примерно одинакового размера.
   • Плохо работает, когда кластеры сильно различаются по размеру.

⚖️ Когда использовать K-Means?

Хорошо подходит, когда:

- Число кластеров известно или может быть оценено
- Кластеры имеют сферическую или выпуклую форму
- Размеры кластеров примерно одинаковы
- Данные числовые и нормализованные
- Требуется быстрая обработка больших данных
- Нужна простая интерпретация результатов

Лучше выбрать другой метод, когда:

- Кластеры имеют сложную форму
- Размеры кластеров сильно различаются
- Данные содержат много выбросов
- Присутствуют категориальные признаки
- Неизвестно число кластеров
- Требуется учёт различной важности признаков