Лекция 8

Кластеризация и логистическая регрессия

Машинное обучение · 2026

Лектор

Елизавета Власова

Практики

Кирилл Захаров
Михаил Подсытник
Игорь Толстокулаков

Виды обучения



В ансамблях были признаки x и известный ответ y.

В кластеризации целевой метки нет. Ищем группы похожих объектов по их признакам.

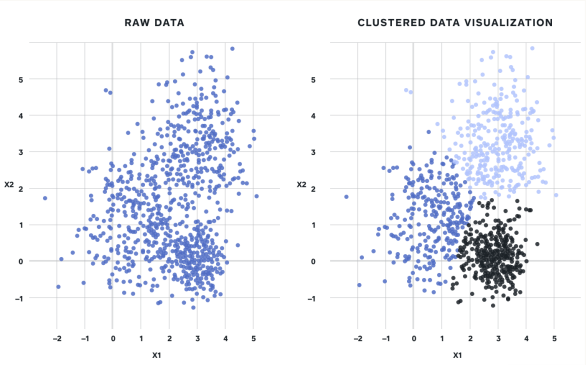
Номер кластера присваивает алгоритм. Он не обязан совпадать с классом объекта.

Кластеризация

Группы объектов без целевой метки

1

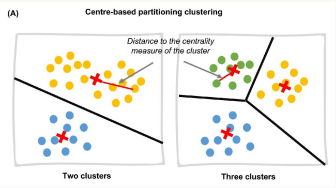
Что такое кластер



Близость зависит от признаков, масштаба и выбранного расстояния.

Подходы к кластеризации

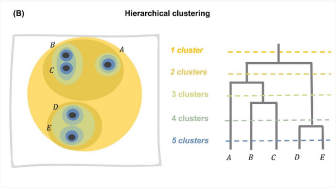
Центроиды



Близость к центру группы.

K-means

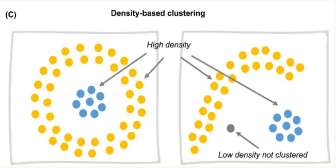
Иерархия



Вложенные группы и дерево объединений.

Агломерация

Плотность



Связные плотные области и шум.

DBSCAN, HDBSCAN

Центроид кластера



Центроид μ_C — среднее векторов объектов кластера C:

$$\mu_C = \frac{1}{|C|} \sum_{x_i \in C} x_i.$$

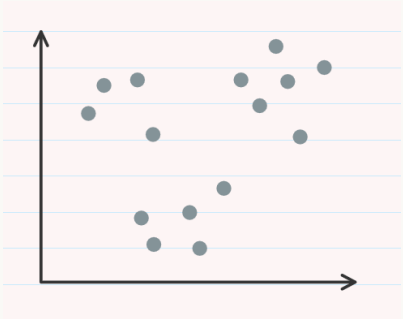
Он минимизирует сумму **квадратов евклидовых расстояний**:

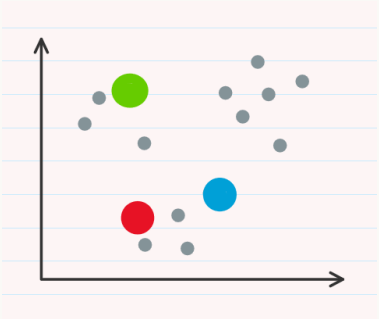
$$\mu_C = \arg\min_{\mu} \sum_{x_i \in C} \|x_i - \mu\|_2^2.$$

Центроид может не совпадать ни с одной точкой данных.

K-means: инициализация

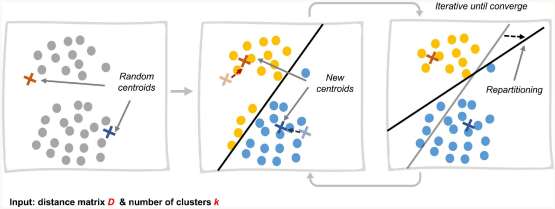
Задаём число кластеров K и выбираем начальные центроиды.





Серые точки — объекты. Крупные цветные круги — начальные центроиды.

K-means: назначение и пересчёт



1. Назначение

Каждая точка получает ближайший центр:

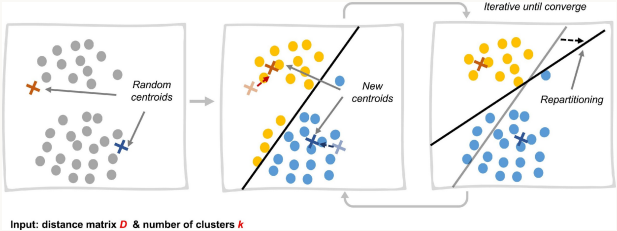
$$c_i = \arg\min_k \|x_i - \mu_k\|_2^2.$$

2. Пересчёт

Новый центр — среднее точек своей группы:

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i.$$

K-means: сходимость

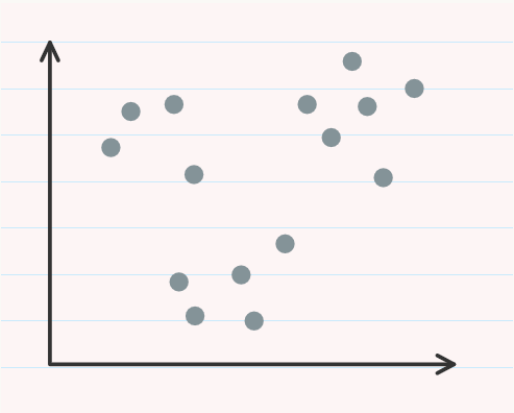


Повторяем назначение точек и пересчёт центров, пока изменения малы.

$$J = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|_2^2 \longrightarrow \min .$$

Каждый шаг не увеличивает J, но итог зависит от начальных центров.

K-means: выбор начальных центров

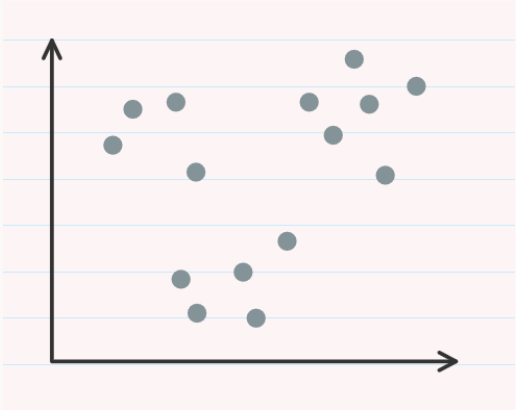


Если несколько начальных центров попадут в одну плотную группу, другой группе может не достаться центра.

Вопрос: что изменится, если запустить алгоритм с другими начальными центрами?

Почему полезно разнести центры ещё до первой итерации?

K-means++



Первый центр выбираем случайно.

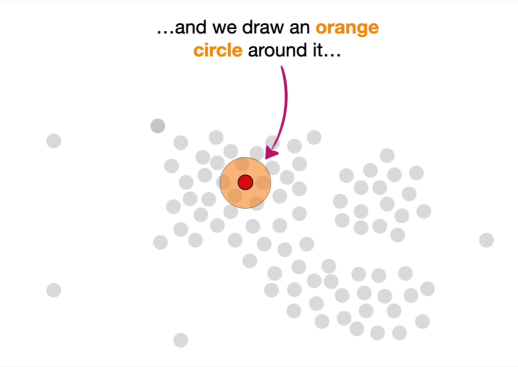
Каждый следующий выбираем с вероятностью

$$P(x_i) = \frac{D(x_i)^2}{\sum_j D(x_j)^2},$$

где $D(x_i)$ — расстояние до ближайшего уже выбранного центра.

Далёкие точки получают больший шанс. Затем запускаем обычный K-means.

DBSCAN: окрестность точки



Соседи точки x находятся на расстоянии не больше ε:

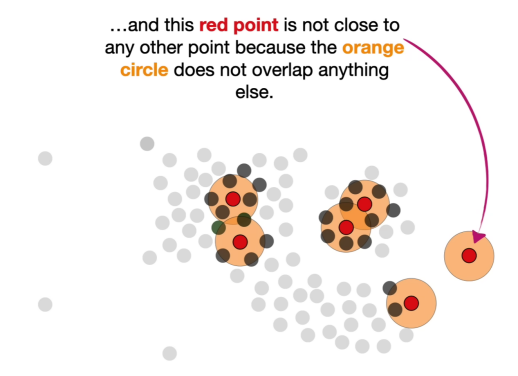
$$N_\varepsilon(x) = \{x_j : d(x, x_j) \le \varepsilon\}.$$

ε — радиус круга на рисунке.

`min_samples` задаёт, сколько точек нужно для плотной окрестности.

В дальнейшем считаем саму точку в её окрестности.

DBSCAN: изолированная точка

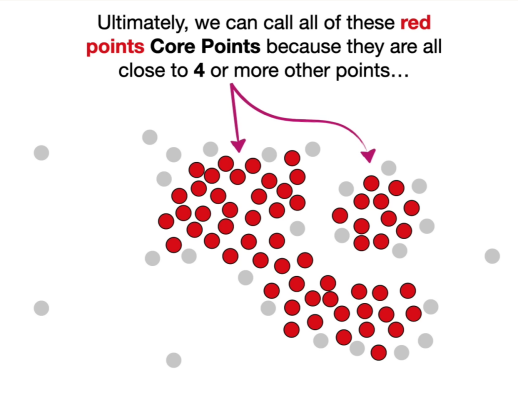


В ε-окрестности правой точки нет других объектов.

При `min_samples>1` она не будет ядром.

До неё не доходит ни один плотный кластер, поэтому она останется **шумом**.

DBSCAN: ядро и граница



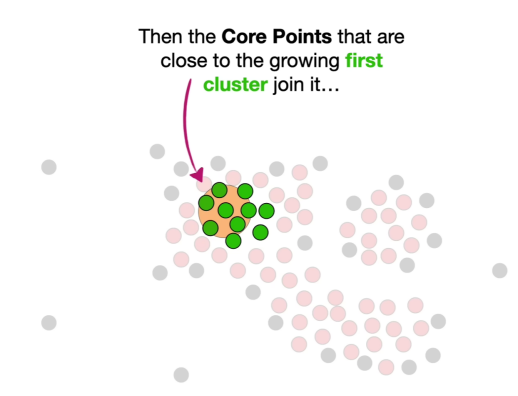
Ядро (core point): в окрестности не меньше `min_samples` точек.

Граничная точка: сама не ядро, но лежит в окрестности ядра.

Шум: точка не относится ни к ядрам, ни к их границам.

Красные точки на схеме — ядра.

DBSCAN: рост кластера



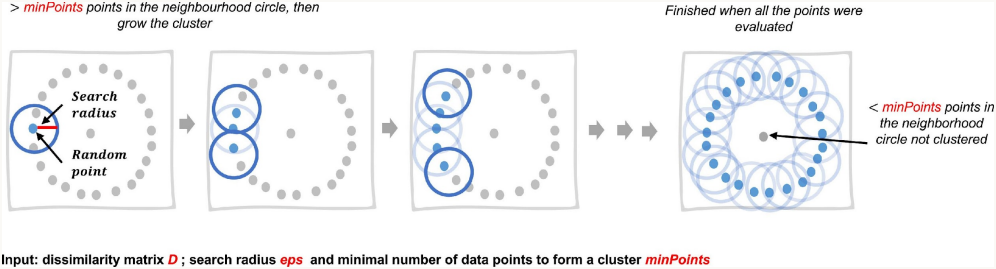
Начинаем с ядра и включаем его соседей.

Через соседние **ядра** продолжаем расширение.

Граничные точки добавляем, но через них рост не продолжаем.

Зелёные точки — уже найденная часть кластера.

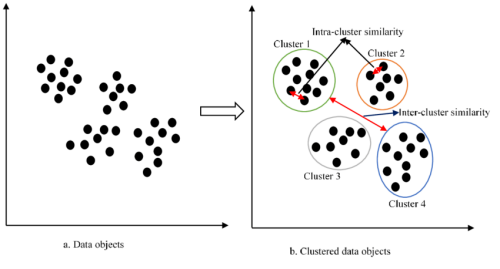
DBSCAN: результат



Цепочки ядер образуют кластеры произвольной формы. Изолированные точки остаются шумом.

Проверка: может ли одна граничная точка соединить две цепочки ядер?

Расстояния внутри и между кластерами



Intra-cluster: расстояния внутри одной группы.

Inter-cluster: расстояния между разными группами.

Для компактных групп хотим малые intra и большие inter.

Компактность и разделённость

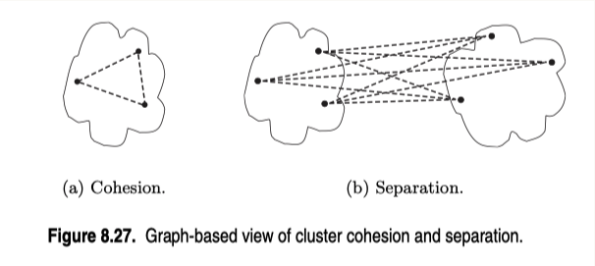


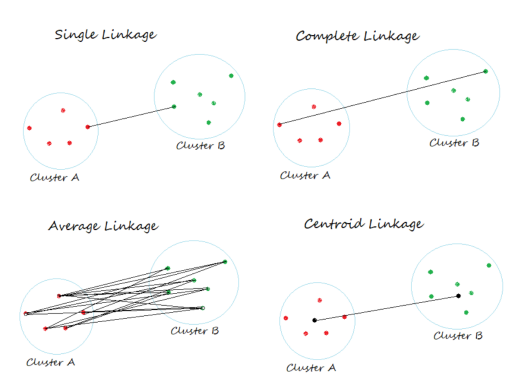
Figure 8.27. Graph-based view of cluster cohesion and separation.

Cohesion — насколько тесно собраны объекты внутри группы.

Separation — насколько группа отделена от остальных.

Конкретный способ измерения задаёт метрика качества.

Расстояние между кластерами



Single linkage: ближайшая пара точек из разных кластеров.

Complete linkage: самая далёкая пара.

Average linkage: среднее по всем парам.

Centroid linkage: расстояние между центроидами.

Один набор точек может дать разные деревья объединений.

Ward linkage

Объединяем пару кластеров A, B, которая меньше всего увеличивает сумму квадратов расстояний внутри групп:

$$\Delta(A, B) = \frac{|A|\,|B|}{|A| + |B|} \|\mu_A - \mu_B\|_2^2.$$

$|A|, |B|$ — размеры групп, μ_A, μ_B — их центроиды.

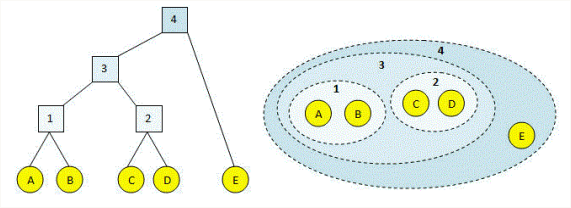
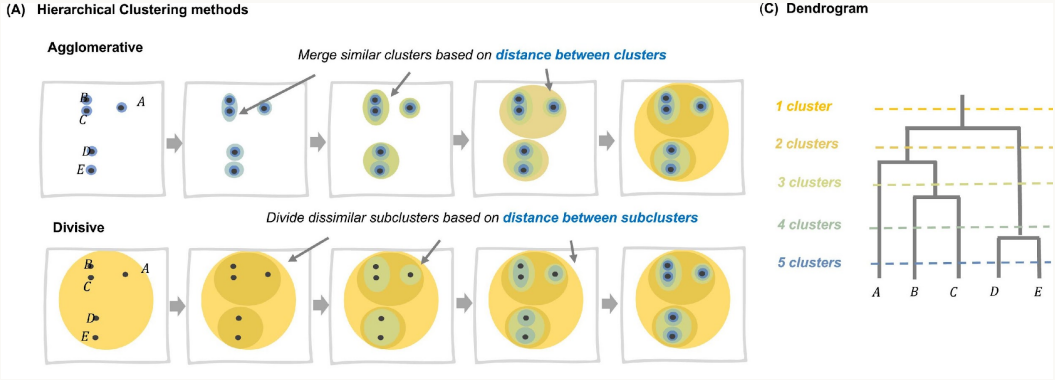


Схема показывает порядок объединений. Ward выбирает его по приросту ошибки.

Иерархическая кластеризация

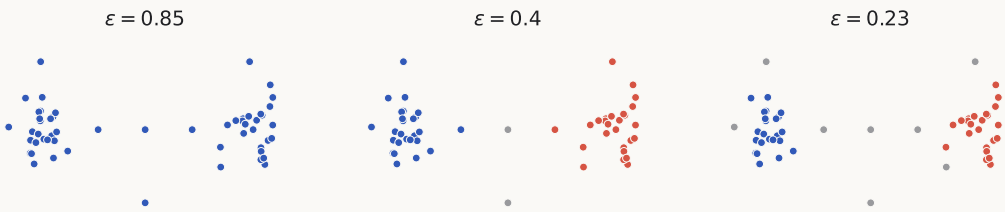


Agglomerative: объединяем точки. **Divisive:** делим общую группу.

Срез дерева на выбранной высоте задаёт итоговые кластеры.

HDBSCAN: разные уровни ε

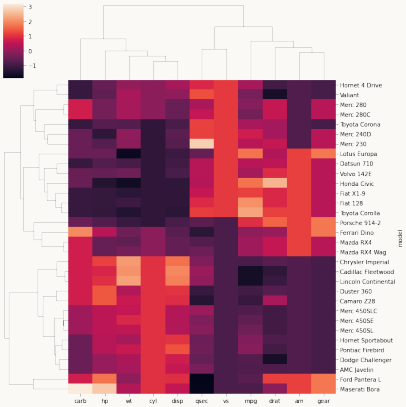
DBSCAN при разных ε на тех же точках. Серые точки — шум.



Идея: проследить группы при разных ε и выбрать те, которые долго сохраняются.

HDBSCAN строит иерархию плотности и выбирает **устойчивые кластеры**. Один общий ε задавать не нужно.

Кластеризация и тепловая карта

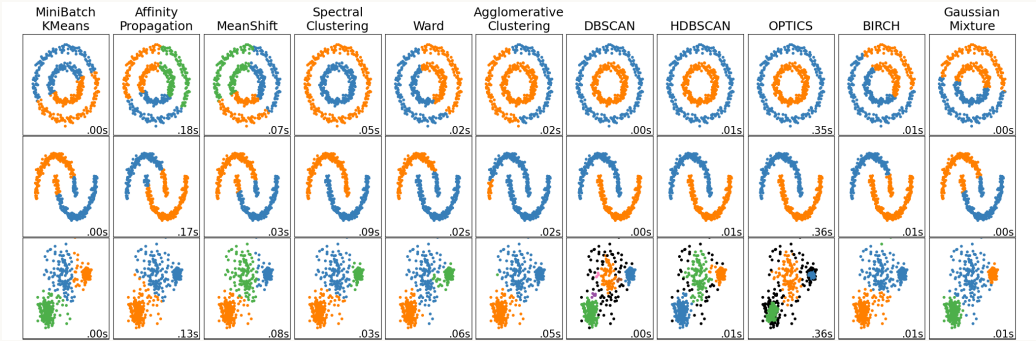


Деревья задают порядок строк и столбцов: похожие объекты оказываются рядом.

Цвет ячейки — значение признака. Это **не номер кластера**.

Число групп выбирают срезом дерева. Масштаб признаков меняет расстояния.

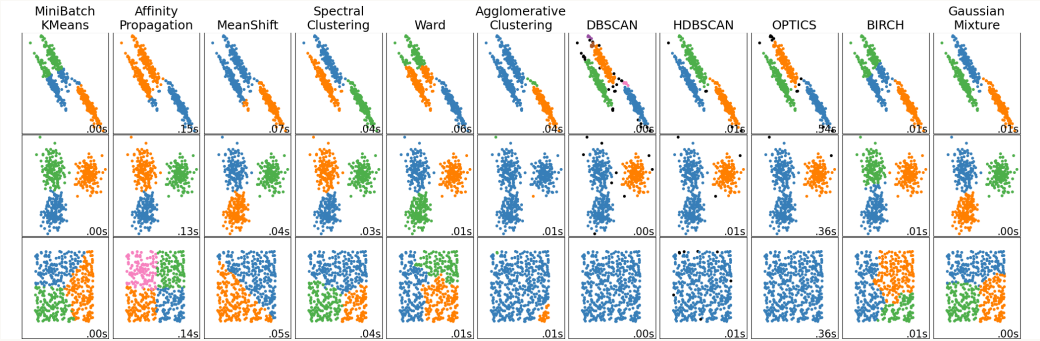
Сравнение: форма и плотность



Строки — наборы данных, столбцы — алгоритмы. Цвета — найденные группы, чёрные точки — шум.

Обратите внимание на кольца, дуги и группы разной плотности.

Сравнение: масштаб и структура



Те же алгоритмы: вытянутые группы, разный разброс и данные без явных кластеров.

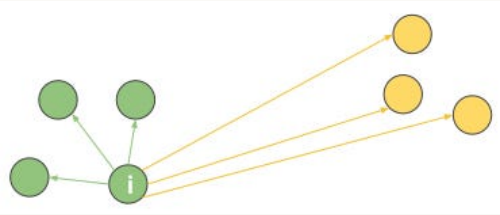
Вопрос: всегда ли разбиение цветом означает, что группы действительно есть?

2

Качество кластеризации

Как оценить группы, если истинных меток нет?

Silhouette score



$a(i)$ — среднее расстояние от точки i до остальных точек своего кластера.

$b(i)$ — минимум среднего расстояния до точек другого кластера.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

$-1 \leq s(i) \leq 1$.

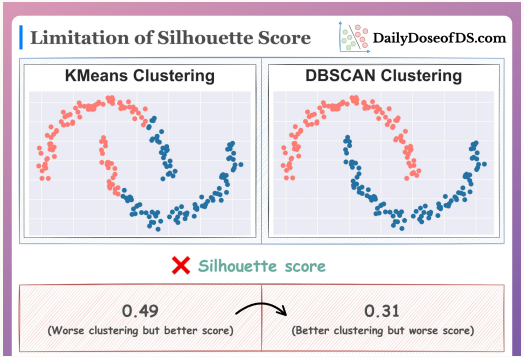
$s \approx 1$: своя группа ближе.

$s \approx 0$: точка между группами.

$s < 0$: другая группа в среднем ближе.

Итог — среднее $s(i)$ по точкам. Истинные классы не нужны.

Ограничение silhouette



Silhouette предпочитает компактные группы.

На двух дугах K-means получает 0.49, а DBSCAN — 0.31.

Если нужны именно связные дуги, больший silhouette выбирает неподходящее разбиение.

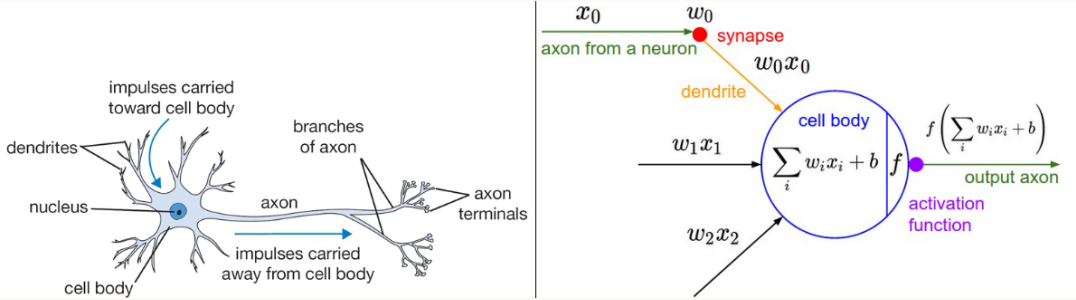
Сначала определяем, какие группы нужны задаче.

3

Логистическая регрессия

Один нейрон для бинарной классификации

Устройство нейрона



Математический нейрон: **взвесить входы** → **сложить** → **применить активацию**.

x_i — входы, w_i — сила их влияния, b — смещение; f преобразует сумму в выход.

Взвешенная сумма

$$z = w^{\mathsf{T}} x + b = \sum_{j=1}^{d} w_j x_j + b.$$

x — d признаков объекта, w — обучаемые веса, b — смещение.

Один проход через линейную часть:

$$x = (2, 1), \qquad w = (1, -1), \qquad b = 0,$$

$$z = 1 \cdot 2 + (-1) \cdot 1 + 0 = 1.$$

Граница классификации $z = 0$ — гиперплоскость.

Дальше преобразуем z в вероятность.

Линейная бинарная классификация

По знаку z можно сразу выбрать класс:

$$\widehat{y} = \begin{cases} +1, & z \geq 0, \\ -1, & z < 0. \end{cases}$$

Здесь используем метки -1 и $+1$.

При $z = 0.01$ и $z = 10$ ответ одинаковый, хотя степень уверенности различается.

Число неверных ответов меняется скачками. Его неудобно минимизировать градиентным спуском.

Вероятность класса

Хотим получить не только метку, но и оценку вероятности:

$$p = P(y = +1 \mid x) = \sigma(z), \qquad z = w^{\mathsf{T}} x + b.$$

Сигмоида переводит любое число z в интервал $(0, 1)$.

Прогноз класса: выбираем $+1$, если $p \geq t$.

При стандартном пороге $t = 0.5$ граница совпадает с $z = 0$.

Порог можно менять под цену ошибок, не переобучая модель.

Сигмоида

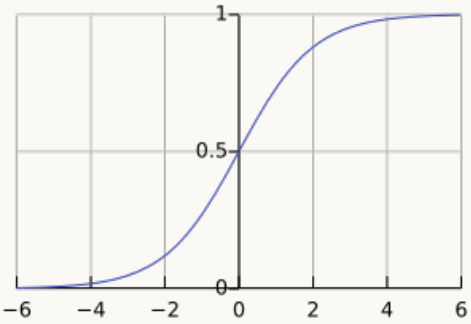
$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

Продолжаем пример:

$$z = 1 \quad \Rightarrow \quad p \approx 0.731.$$

Модель оценивает вероятность класса $+1$ примерно в 73%.

Проверка: чему равно p при $z = 0$?



Горизонтальная ось — z.
Вертикальная — вероятность p.

Вероятности двух классов

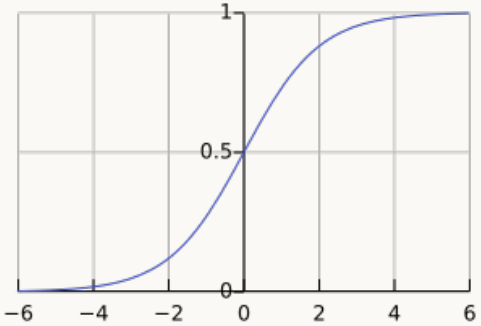
При $z = 0$ получаем $p = 0.5$.

$$P(y = -1 \mid x) = 1 - p = \sigma(-z).$$

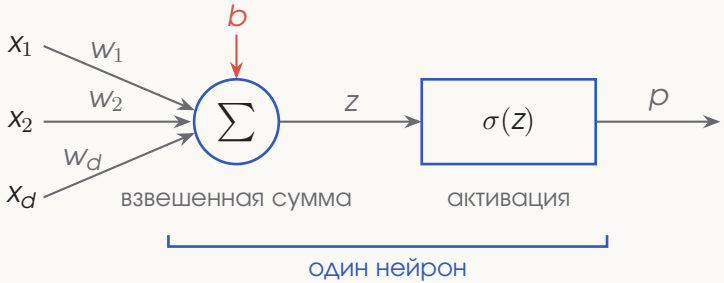
Для меток $y \in \{-1, +1\}$ обе формулы записываются вместе:

$$P(y \mid x) = \sigma(yz).$$

Если знак z совпадает с y, то $yz > 0$ и правильному классу назначена вероятность больше 0.5.



Логрег — один нейрон



Логрег использует ровно этот блок: $p = \sigma(w^T x + b)$.

z может быть любым числом. Сигмоида даёт вероятность $p \in (0, 1)$.

При пороге $p = 0.5$ граница всё ещё линейная: $w^T x + b = 0$.

Обучение: правдоподобие

В обучающей выборке известны пары (x_i, y_i). Хотим назначать правильным классам высокие вероятности:

$$\mathcal{L}(w, b) = \prod_{i=1}^{N} \sigma(y_i z_i) \longrightarrow \max, \qquad z_i = w^{\mathsf{T}} x_i + b.$$

Произведение относится к **всем объектам** при условной независимости наблюдений.

Удобнее перейти к сумме логарифмов и минимизации:

$$-\log \mathcal{L} = \sum_{i=1}^{N} \log(1 + e^{-y_i z_i}) \longrightarrow \min.$$

Это логистическая функция потерь.

LogLoss и обучение весов

Перекодируем метки: $-1 \mapsto 0$, $+1 \mapsto 1$. Теперь $y_i \in \{0, 1\}$, а $p_i = P(y_i = 1 \mid x_i)$.

$$L = -\frac{1}{N} \sum_{i=1}^{N} \Big[y_i \log p_i + (1 - y_i) \log(1 - p_i) \Big].$$

Если $y = 1$**:** при $p = 0.9$ ошибка $-\log 0.9 \approx 0.105$, а при $p = 0.1$ — 2.303.

Обновляем веса и смещение градиентным спуском:

$$w \leftarrow w - \eta \nabla_w L, \qquad b \leftarrow b - \eta \frac{\partial L}{\partial b}.$$

η — шаг обучения. Веса меняются, чтобы уменьшать ошибку.

Нелинейность между слоями

Без нелинейной активации два слоя сворачиваются в один:

$$h = W_1 x + b_1, \qquad z = W_2 h + b_2$$

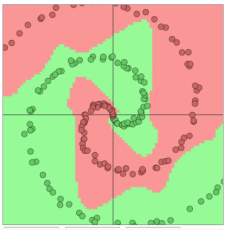
$$z = (W_2 W_1)x + (W_2 b_1 + b_2) = Wx + b.$$

W_i, b_i — веса и смещения слоёв; h — промежуточный выход.

Сколько таких слоёв ни добавь, модель останется линейной.

Нелинейная активация φ между слоями позволяет строить границы сложной формы:

$$h = \varphi(W_1 x + b_1).$$



Такие классы одной прямой не разделить.

Теоретический минимум

- Кластеризация
- Центроид
- K-means и K-means++
- DBSCAN: ядро, граница, шум
- Иерархическая кластеризация
- Linkage и дендрограмма
- HDBSCAN
- Silhouette score
- Нейрон и веса
- Смещение
- Функция активации
- Логистическая регрессия
- Сигмоида
- LogLoss
- Нелинейность между слоями

Ко(т)нец презентации



В следующей лекции

Объединяем нейроны в сеть.

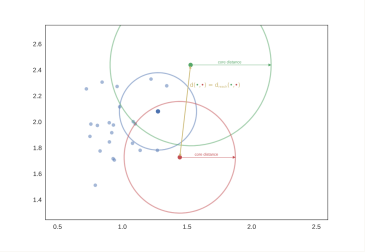
Слои, нелинейные активации и обучение нейросети.

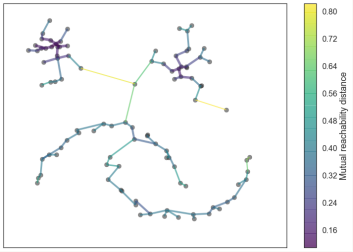
Дополнительные материалы

HDBSCAN: устройство алгоритма · Метрики и выбор параметров



HDBSCAN: расстояние с учётом плотности





$c_k(x)$ — расстояние до k-го соседа. Чем оно больше, тем разреженнее окрестность.

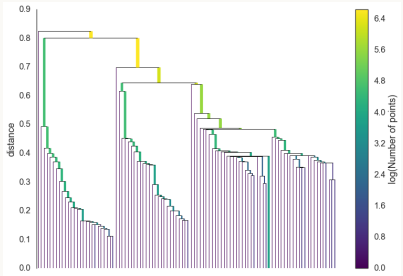
Соединяем точки в остовное дерево. Цвет ребра показывает новое расстояние.

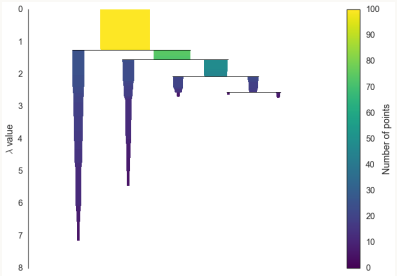
$$d_{\mathrm{mr}}(x, y) = \max\{c_k(x), c_k(y), d(x, y)\}.$$

$d(x, y)$ — исходное расстояние. Разреженные «мосты» получают больший вес.

HDBSCAN: иерархия плотных групп

DBSCAN использует один радиус ε. HDBSCAN рассматривает разные уровни плотности и ищет устойчивые группы.





Слева — объединения по расстоянию. Справа — сжатое дерево; $\lambda = 1/d$ растёт в сторону более плотных групп. Ширина ветви связана с числом точек.

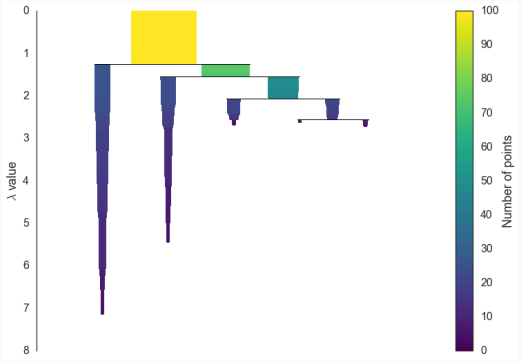
HDBSCAN: устойчивость и параметры

Устойчивый кластер сохраняется при изменении уровня плотности.

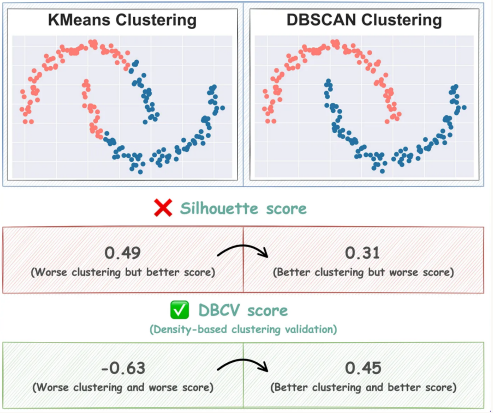
`min_cluster_size` — минимальный размер группы.

`min_samples` влияет на оценку плотности и долю шума.

Построение дерева и иерархии добавляет работу. Время зависит от данных и реализации.



DBCV: качество плотностных кластеров



Сравниваем плотностное разделение групп и разреженные места внутри них.

$$V_i = \frac{\mathrm{sep}_i - \mathrm{spar}_i}{\max(\mathrm{sep}_i, \mathrm{spar}_i)}.$$

sep_i — минимальное плотностное расстояние до другого кластера.

spar_i — самое слабое соединение внутри кластера.

DBCV усредняет V_i с весами размеров групп.

Больше — лучше.

Выбор ε по k-distance

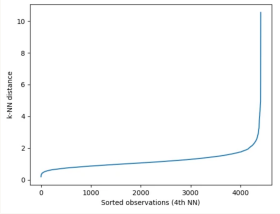
Фиксируем `min_samples`$= m$.

Для каждой точки считаем расстояние до $(m-1)$-го **другого** соседа.

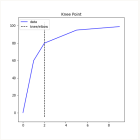
Сортируем расстояния по возрастанию.

Значение около перегиба — кандидат на ε.

Это эвристика. При разной плотности одного удобного перегиба может не быть.

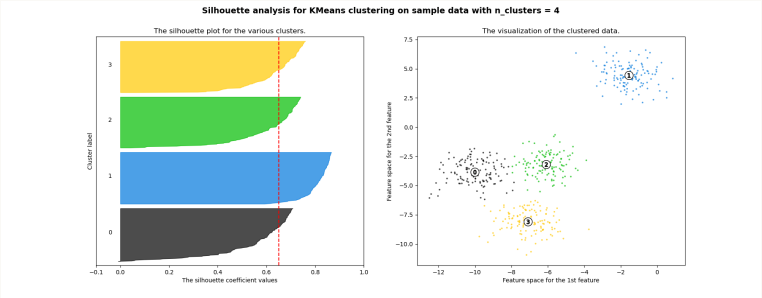


Сверху — отсортированные расстояния.



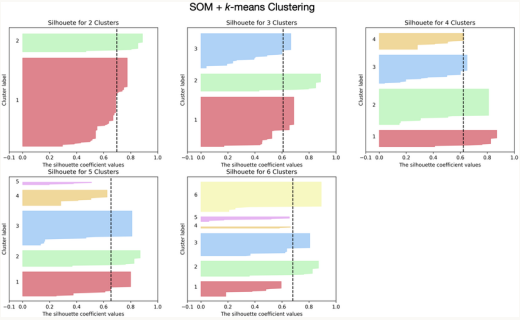
Снизу — общая иллюстрация перегиба.

Как читать silhouette plot



По горизонтали — $s(i)$, по вертикали — точки, сгруппированные по кластерам. Высота полосы показывает размер группы. Пунктир — среднее по всем точкам. Смотрим на отрицательные значения и группы с низким silhouette.

Silhouette при разном числе кластеров



Сравниваем среднее значение, распределение внутри групп и отрицательные хвосты.

Размеры кластеров могут различаться. Равенство размеров не является обязательным критерием.

Вопрос: достаточно ли одного среднего числа, чтобы выбрать K?

Purity: известны истинные классы

C_k — кластер, y_i — истинный класс объекта. Находим самый частый класс c:

$$\text{purity}(C_k) = \frac{1}{|C_k|} \max_c \big|\{i \in C_k : y_i = c\}\big|.$$

Если все N объектов распределены по кластерам, общая purity:

$$\text{Purity} = \frac{1}{N} \sum_k |C_k| \, \text{purity}(C_k).$$

Пример: 8 объектов класса A и 2 класса B дают purity $= 8/10 = 0.8$.

Один объект в каждом кластере даст purity $= 1$. Поэтому одной purity недостаточно.

Retention: сколько объектов сохранили

При пороге чистоты τ считаем долю всех объектов в достаточно чистых кластерах:

$$\mathrm{Retention}(\tau) = \frac{1}{N} \sum_{k:\ \mathrm{purity}(C_k) \geq \tau} |C_k|.$$

N включает все объекты, в том числе шум. τ — требуемая чистота.

Пример: из 100 объектов 70 находятся в кластерах с purity ≥ 0.9. Тогда retention$(0.9) = 0.70$.

Для фиксированного разбиения рост τ не увеличивает retention.