Обучение без учителя. Кластеризация

Санкт-Петербургский государственный университет
Кафедра статистического моделирования
Семинар «Статистическое и машинное обучение»

Санкт-Петербург, 2025

Обучение без учителя

Обучение без учителя (unsupervised learning) — раздел машинного обучения, изучающий класс задач обработки данных, в которых известны только описания множества объектов (признаки объектов) из обучающей выборки, и требуется обнаружить внутренние зависимости, существующие между объектами. В отличие от обучения с учителем, правильные «ответы» или «метки» для объектов неизвестны.

Задачи обучения без учителя

- **Кластеризация.** Разделение объектов на группы (кластеры) на основании их сходства.
- **Поиск ассоциативных правил.** Выявление связей между объектами. Цель — найти закономерности вида «Если встречается A, то с высокой вероятностью встречается и B», где A, B — некоторые не пересекающиеся наборы признаков.
- **Понижение размерности.** Уменьшение количества признаков при сохранении максимально возможной информации из исходных данных.
- **Заполнение пропусков.** Восстановление отсутствующих данных на основе закономерностей, найденных в имеющихся данных.

Постановка задачи кластеризации

Дано:

$\boldsymbol{X}$ — пространство объектов;

$\boldsymbol{X}^n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ — выборка из $\boldsymbol{X}$, где $\boldsymbol{x}_i$ — i-й объект;

$\rho : \boldsymbol{X} \times \boldsymbol{X} \to [0, \infty)$ — функция расстояния между объектами.

Найти:

Y — множество кластеров;

$a : \boldsymbol{X} \to Y$ — алгоритм кластеризации.

Причем Y и a такие, что

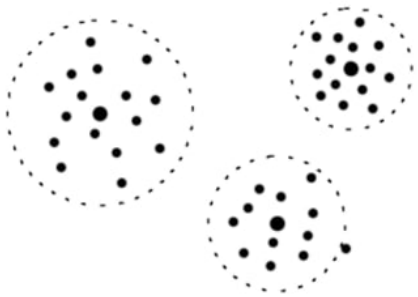
- каждый кластер состоит из близких объектов (относительно ρ);
- объекты разных кластеров существенно различаются.

Некорректность задачи кластеризации

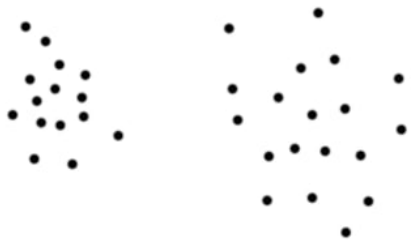
Решение задачи кластеризации неоднозначно:

- точной постановки задачи кластеризации нет;
- существует много критериев качества кластеризации;
- число кластеров, как правило, неизвестно заранее;
- результат кластеризации сильно зависит от метрики ρ, выбор которой также не однозначен.

Типы кластерных структур



кластеры с центрами



внутрикластерные расстояния
меньше межкластерных

ленточные кластеры

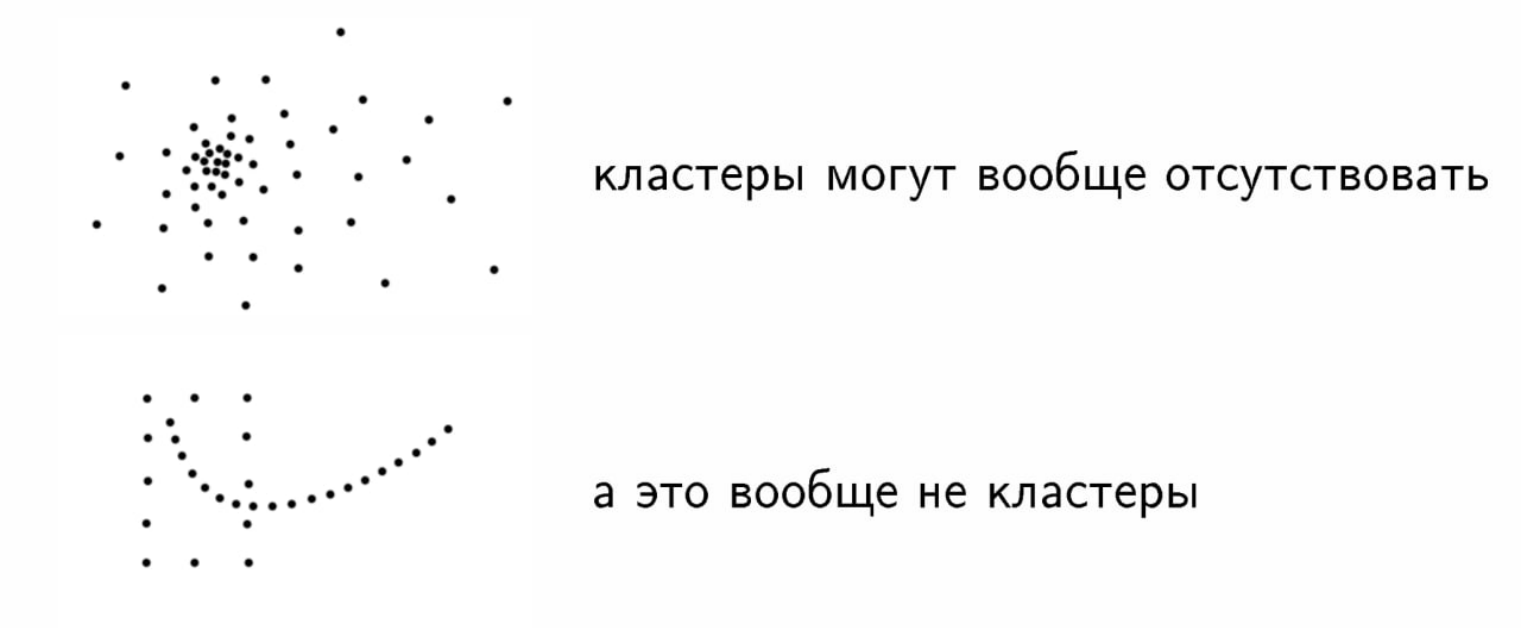
Типы кластерных структур

перемычки между кластерами

разреженный фон
из нетипичных объектов

перекрывающиеся кластеры

Типы кластерных структур



- Каждый метод кластеризации имеет свои ограничения и выделяет кластеры лишь некоторых типов.
- Понятие «тип кластерной структуры» зависит от метода и также не имеет формального определения.

Классификация алгоритмов

В большинстве источников выделяют пять групп алгоритмов:

- **Основанные на центроидах (centroid based):** k-means, k-modes, k-medoids, **Meanshift**, Affinity Propagation
- **Иерархические (hierarchical):** агломеративные (Ward/single/average/complete linkage), BIRCH, на основе теории графов (выделение связных компонент и минимальное остовное дерево), Spectral Clustering
- **Основанные на плотности (density based):** **DBSCAN**, **HDBSCAN**, **OPTICS**
- **Сеточные (grid based):** STING, Wave cluster, **CLIQUE**, OptiGrid, MAFIA
- **Основанные на модели данных (model based):** **Expectation Maximization (EM)**, COBWEB

EM-алгоритм для model-based подхода

Model-based clustering — подход с четко поставленной задачей: предполагаем какую-то статистическую модель данных и в ней находим параметры.

Выборка $\boldsymbol{X}^n$ — случайна, независима и взята из смеси распределений, плотность которой в точке $\boldsymbol{x} \in \boldsymbol{X}^n$ представима в виде:

$$p(\boldsymbol{x}) = \sum_{j=1}^{k} w_j p_j(\boldsymbol{x}; \boldsymbol{\theta}_j), \text{ при этом}$$

- $\sum\limits_{j=1}^{k} w_j = 1$, $w_j \geq 0$ — априорные вероятности кластеров.
- $p_j(\boldsymbol{x}; \boldsymbol{\theta}_j)$ — плотность распределения j-го кластера с параметрами $\boldsymbol{\theta}_j$.

EM-алгоритм для model-based подхода

Предполагается, зная число кластеров k и вид плотностей p_j, оценить параметры $w_j, \boldsymbol{\theta}_j$, максимизируя логарифм функции правдоподобия

$$\ln L(\{\boldsymbol{x}_i\}; \{w_j\}; \{\boldsymbol{\theta}_j\}) = \sum_{i=1}^{n} \ln \sum_{j=1}^{k} w_j p_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) \to \max_{\{w_j\}, \{\boldsymbol{\theta}_j\}}.$$

Для решения данной задачи применяется **EM-алгоритм**.

EM-алгоритм. Шаг E (Expectation)

Для текущих оценок параметров вычисляем вероятность принадлежности каждой точки $\boldsymbol{x}_i$ к каждой компоненте смеси с параметрами $\boldsymbol{\theta}_j$ по формуле Байеса:

$$g_{ij} = \frac{w_j p_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)}{\sum\limits_{s=1}^{k} w_s p_s(\boldsymbol{x}_i; \boldsymbol{\theta}_s)} \quad - \text{ скрытые переменные.}$$

EM-алгоритм. Шаг M (Maximization)

Обновляем параметры, используя скрытые переменные, найденные на предыдущем шаге. Максимизируем логарифм функции правдоподобия методом Лагранжа:

$$\mathcal{L}(\{\boldsymbol{x}_i\};\{w_j\};\{\boldsymbol{\theta}_j\}) = \sum_{i=1}^{n} \ln \sum_{j=1}^{k} w_j p_j(\boldsymbol{x}_i;\boldsymbol{\theta}_j) - \lambda(\sum_{j=1}^{k} w_j - 1).$$

Из равенства нулю производной по w_j следует:

$$w_j = \frac{1}{n} \sum_{i=1}^{n} g_{ij}, \quad j = 1, \ldots, k.$$

Из равенства нулю производной по $\boldsymbol{\theta}_j$ следует:

$$\boldsymbol{\theta}_j = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} g_{ij} \ln p(\boldsymbol{x_i};\boldsymbol{\theta}), \quad j = 1, \ldots, k.$$

Таким образом, параметры будут уточняться на каждом шаге.

EM-алгоритм. Случай нормальных плотностей

Предположим, что компоненты смеси имеют нормальные распределения со средними $\boldsymbol{\mu}_j$ и матрицами ковариаций Σ_j, тогда имеем следующие оценки параметров:

$$\boldsymbol{\mu}_j = \frac{1}{nw_j} \sum_{i=1}^{n} g_{ij} \boldsymbol{x}_i,$$

$$\Sigma_j = \frac{1}{nw_j} \sum_{i=1}^{n} g_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}.$$

EM-алгоритм для model-based. Плюсы и минусы

Преимущества:

- Имеет формально поставленную задачу;
- Полученное разбиение на кластеры интерпретируемо со статистической точки зрения.

Недостатки:

- Число кластеров k является гиперпараметром;
- Параметры должны быть оценены, что требует большего количества точек данных в каждой компоненте;
- Алгоритм чувствителен к начальным данным.

k-means и его модификации

- k-means — итеративно группирует данные вокруг k-центров кластеров, пересчитывая их положение как среднее точек кластера до сходимости.
- Mini-batch k-means — модификация классического k-means, использующая случайные подвыборки данных на каждой итерации для обучения. Хорошо подходит для больших датасетов.
- k-medoids — вариант k-means, который в качестве центроидов выбирает реальные точки (медоиды) из данных, а не их средние значения, что повышает устойчивость к выбросам.
- k-modes — вариант алгоритма k-means для работы с категориальными данными, который выбирает один из объектов в кластере в качестве моды и минимизирует сумму расстояний Хэмминга между модой и объектами в кластере. Расстояние Хэмминга представляет из себя количество позиций, в которых значения векторов не совпадают.

Mean Shift

Mean Shift — это алгоритм кластеризации, который реализует локальный градиентный подъём по оценке плотности данных (KDE). Сдвиг — это направление наибольшего возрастания плотности, а итоговые «центры» кластеров — локальные максимумы плотности (моды). Точки, которые сходятся к одному максимуму, считаются принадлежащими одному кластеру.

В отличие от популярного алгоритма k-means, Mean Shift не требует предварительного указания количества кластеров, но требует задать параметр окна для KDE.

Mean Shift. Определения

- Ядерная оценка плотности в $\mathbb{R}^p$ (p — кол-во признаков) с окном ширины h (bandwidth) и радиально-симметричным ядром:

$$f_h(\boldsymbol{x}) = \frac{1}{nh^p} \sum_{i=1}^{n} k\left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right\|^2 \right), \text{ где } k : [0, \infty] \to \mathbb{R} \text{ — профиль ядра.}$$

- Средневзвешенное значение плотности в окне:

$$m(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\boldsymbol{x} - \boldsymbol{x}_i}{h} \right\|^2 \right)}, \text{ где } g(u) = -k'(u).$$

  $m(\boldsymbol{x})$ пропорционален градиенту функции $f_h(\boldsymbol{x})$.

- Вектор сдвига (mean shift) равен $m(\boldsymbol{x}) - \boldsymbol{x}$ и направлен в сторону максимального увеличения плотности.

Mean Shift. Алгоритм

1. Инициализация — каждая точка данных становится потенциальным центром кластера.
2. Для каждой точки создается скользящее окно с фиксированным радиусом (bandwidth).
3. Вычисляется центроид $m(\boldsymbol{x})$ всех точек в пределах окна как взвешенное среднее.
4. Центр окна перемещается к вычисленному центроиду.
5. Процесс повторяется до тех пор, пока окна не перестанут существенно смещаться (достигнута сходимость).
6. Пересекающиеся окна объединяются, выбирается окно с наибольшим количеством точек.
7. Точки данных назначаются кластерам в соответствии с окном, в котором они находятся.

Mean Shift. Плюсы и минусы

Преимущества:

- Не требует предварительного задания количества кластеров k;
- Может находить кластеры произвольной формы;
- Теоретически обоснован как метод поиска мод плотности.

Недостатки:

- Чувствителен к параметру bandwidth h: малое h приводит к большому количеству мелких кластеров, может создать отдельный кластер для каждой точки; большое h приводит к малому количеству крупных кластеров, может объединить все точки в один кластер;
- Высокая вычислительная сложность для больших наборов данных;
- Нормализация/стандартизация данных может существенно изменить расклад.

DBSCAN

DBSCAN (Density-based spatial clustering of applications with noise) — это эвристический алгоритм кластеризации, основанный на плотности. Алгоритм группирует вместе те объекты, которые тесно расположены, помечая как выбросы объекты, которые находятся в областях с малой плотностью.

Помимо того, что DBSCAN может обнаруживать кластеры произвольной формы и выбросы в данных, его главная особенность заключается в самостоятельном определении необходимого количества кластеров, что избавляет от необходимости в их подборе.

Алгоритм достаточно прост, наряду с k-means один из самых популярных

DBSCAN. Типы объектов

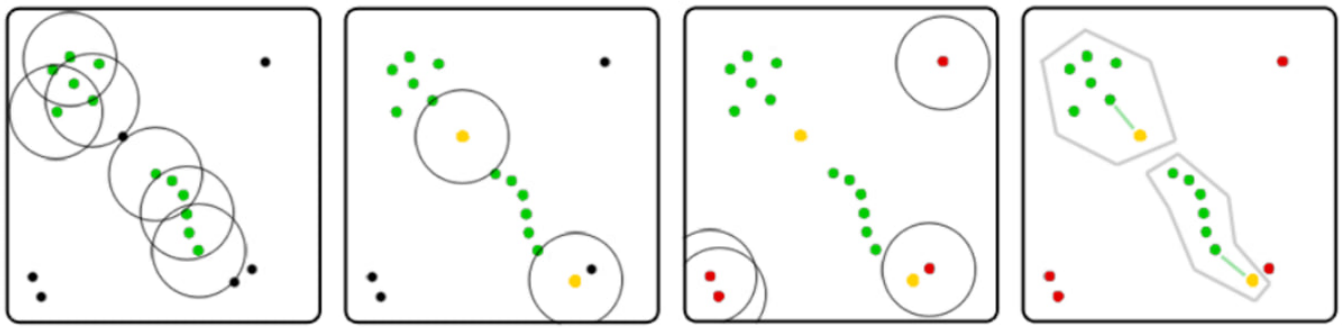
Для $\boldsymbol{x} \in \boldsymbol{X}$ его ε-окрестность $U_\varepsilon(\boldsymbol{x}) = \{\boldsymbol{u} \in \boldsymbol{X} : \rho(\boldsymbol{x}, \boldsymbol{u}) \leqslant \varepsilon\}$.

Алгоритм имеет 2 гиперпараметра:

- величина окрестности ε;
- минимальное количество объектов в окрестности m.

Каждый объект может быть одного из трёх типов:

- **корневой** (core): в ε-окрестности не менее m точек;
- **граничный** (border): в ε-окрестности меньше m точек, но среди них есть как минимум одна корневая;
- **шумовой** (noise): не корневой и не граничный.



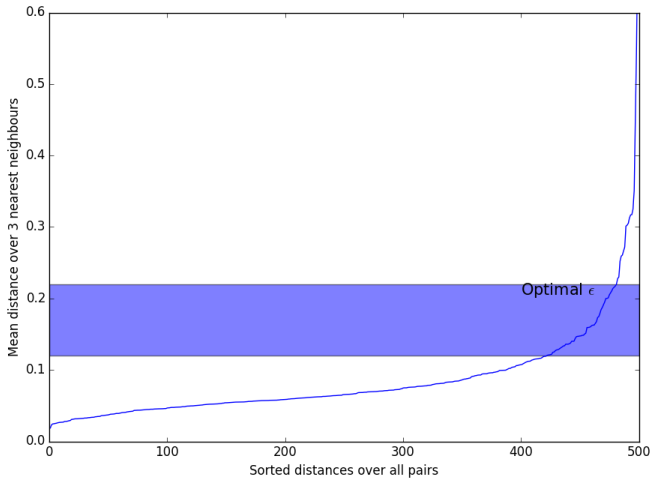
DBSCAN. Алгоритм

Вход: выборка $\boldsymbol{X}^n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, параметры ε и m;
Выход: разбиение выборки на кластеры и шумовые выбросы;

1. $U = \boldsymbol{X}^n$ — мн-во еще не обработанных объектов, $a = 0$;
2. **Пока** есть некластеризованные точки, т.е. $U \neq \varnothing$;
3. взять случайную точку $\boldsymbol{x} \in U$;
4. **если** $|U_\varepsilon(\boldsymbol{x})| < m$, **то**
5. пометить $\boldsymbol{x}$ как шумовой;
6. **иначе**
7. создать новый кластер: $K = U_\varepsilon(\boldsymbol{x})$; $a = a + 1$;
8. **для всех** $\boldsymbol{x}' \in K$
9. **если** $|U_\varepsilon(\boldsymbol{x}')| \geq m$ **то** $K = K \cup U_\varepsilon(\boldsymbol{x}')$;
10. **иначе** пометить $\boldsymbol{x}'$ как граничный элемент K;
11. соотнести объект классу a для всех $\boldsymbol{x}' \in K$;
12. $U = U \setminus K$

DBSCAN. Подбор параметров

- Значение параметра m предлагается выбирать как $m \geqslant p + 1$, где p — количество признаков (размерность данных). Также встречается $m = 2 \cdot p$.
- Чтобы подобрать ε, используют следующий алгоритм:
  - Строится график, где по оси y для каждой точки будет среднее расстояние по m ближайшим соседям, а по оси x — точки, отсортированные в порядке возрастания этого расстояния.
  - Следует взять ε где-нибудь в полосе, где происходит самый сильный перегиб.



DBSCAN. Плюсы и минусы

Преимущества:

- Относительно быстро работает на больших данных;
- Устойчив к выбросам;
- Не требуется заранее указывать количество кластеров;
- Способен находить кластеры произвольной формы, а также шумовые точки в данных;
- Хорошо поддаётся модифицированию.

Недостатки:

- Чувствителен к выбору параметров ε и m;
- Проблемы с высокоразмерными данными (проклятие размерности);
- Плохо работает с кластерами разной плотности. Не способен соединять кластеры через проёмы, и, наоборот, способен связывать явно различные кластеры через плотно населённые перемычки.

OPTICS

OPTICS (Ordering points to identify the clustering structure) — модификация DBSCAN для решения его ключевой проблемы: неспособности эффективно обрабатывать данные с кластерами различной плотности.

Основная идея OPTICS заключается в линейном упорядочивании точек таким образом, чтобы пространственно близкие точки становились соседними в этом упорядочивании. Этот порядок затем можно использовать для извлечения кластеров с любыми параметрами плотности.

OPTICS. Определения

- **Основное расстояние** (Core Distance) d_{core} — расстояние от точки до ее m-го ближайшего соседа. То есть это такое минимальное расстояние $\varepsilon' \leqslant \varepsilon$, при котором точка все еще остается корневой (основной).

  Если такого расстояния $\leqslant \varepsilon$ не существует, d_{core} считается неопределённой.
- **Достижимое расстояние** (Reachability Distance) — мера того, насколько легко точка может быть достигнута из других точек в наборе данных:

$$d_{reach}(\boldsymbol{a}, \boldsymbol{b}) = \max(d_{core}(\boldsymbol{b}), \rho(\boldsymbol{a}, \boldsymbol{b})), \text{ где } \rho(\boldsymbol{a}, \boldsymbol{b}) = \mathsf{dist}(\boldsymbol{a}, \boldsymbol{b}).$$

- Как основное, так и достижимое расстояния не определены, если нет достаточно плотного кластера (применительно к ε).

OPTICS. Алгоритм

Алгоритм строит упорядочивание точек $\pi = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ такое, что последовательность достижимых расстояний $r_i = d_{reach}(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i)$ отражает переходы между плотными областями данных.

1. $d_{reach}(\boldsymbol{x}) = \infty, \quad \forall \boldsymbol{x} \in \boldsymbol{X}^n$.
2. Для каждой непосещённой точки $\boldsymbol{x} \in \boldsymbol{X}^n$:
   - Вычислить $d_{core}(\boldsymbol{x})$.
   - Добавить $\boldsymbol{x}$ в упорядочение π.
   - Если $d_{core}(\boldsymbol{x})$ определена, то для всех $\boldsymbol{y} \in U_\varepsilon(\boldsymbol{x})$, которые ещё не включены в упорядочение, пересчитывается их достижимость:

$$d_{reach}(\boldsymbol{y}) = \min(d_{reach}(\boldsymbol{y}), \max(d_{core}(\boldsymbol{x}), \rho(\boldsymbol{x}, \boldsymbol{y}))).$$

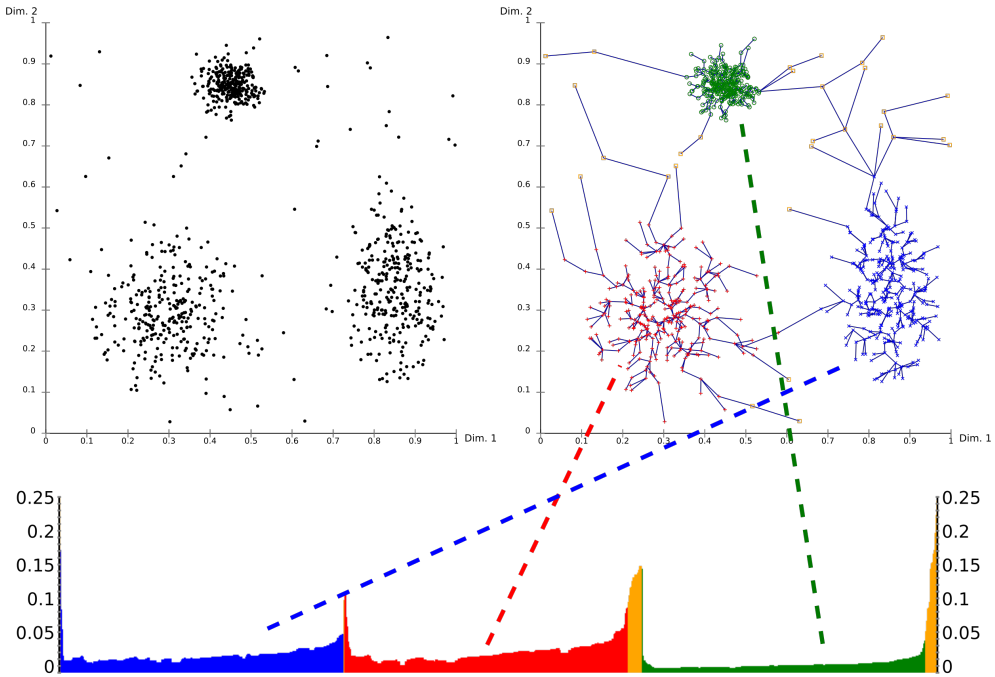
   - Следующей выбирается точка

$$p' = \arg\min_{\boldsymbol{y} \notin \pi} d_{reach}(\boldsymbol{y})$$

   и процесс повторяется до тех пор, пока все точки не будут упорядочены.

OPTICS. Извлечение кластеров

- Результатом работы алгоритма является **график достижимости** — двумерный график, где по оси x откладываются точки в порядке их обработки алгоритмом, а по оси y — достижимое расстояние.
- Поскольку точки, принадлежащие кластеру, имеют небольшое достижимое расстояние до ближайшего соседа, **кластеры** выглядят как **долины** на графике достижимости. Чем глубже долина, тем плотнее кластер.
- **Пики** на графике представляют расстояния между кластерами или переходы от кластера к шуму.



HDBSCAN

HDBSCAN (Hierarchical DBSCAN) — модификации DBSCAN, которая автоматически находит подходящее значение ε для каждого кластера, используя иерархический подход, что позволяет определять кластеры с разной плотностью.

По сравнению с DBSCAN для него требуется большее количество вычислений, что увеличивает время работы алгоритма.

HDBSCAN. Определения

- **Основное расстояние** (Core Distance) d_{core} — расстояние от точки до ее m-го ближайшего соседа. Это величина, показывающая, насколько плотна область вокруг точки.

- **Взаимное достижимое расстояние** (Mutual Reachability Distance) — специальная метрика:

$$d_{mreach}(\boldsymbol{a}, \boldsymbol{b}) = \max(d_{core}(\boldsymbol{a}), d_{core}(\boldsymbol{b}), \rho(\boldsymbol{a}, \boldsymbol{b})), \text{ где } \rho(\boldsymbol{a}, \boldsymbol{b}) = \text{dist}(\boldsymbol{a}, \boldsymbol{b}).$$

Расстояния между точками в областях высокой плотности (с малым d_{core}) остаются неизменными, а расстояния между точками в областях низкой плотности (с большим d_{core}) увеличиваются. Таким образом, общий эффект использования MRD в качестве метрики расстояния заключается в том, что точки в областях с низкой плотностью отдаляются от точек в областях с высокой плотностью.

Это делает алгоритм устойчивым к разной плотности.

HDBSCAN. Алгоритм

1. Для каждой точки рассчитывается d_{core}.
2. Вычисляем MRD между всеми парами точек, строим полный взвешенный граф.
3. На графе взаимной достижимости строим минимальное остовное дерево (MST).
4. Строим иерархию кластеров (преобразовываем MST в дендрограмму):
   - Сортируем рёбра MST по весу в порядке возрастания;
   - Начиная с самого маленького расстояния, рёбра последовательно добавляются, соединяя точки и формируя кластеры.

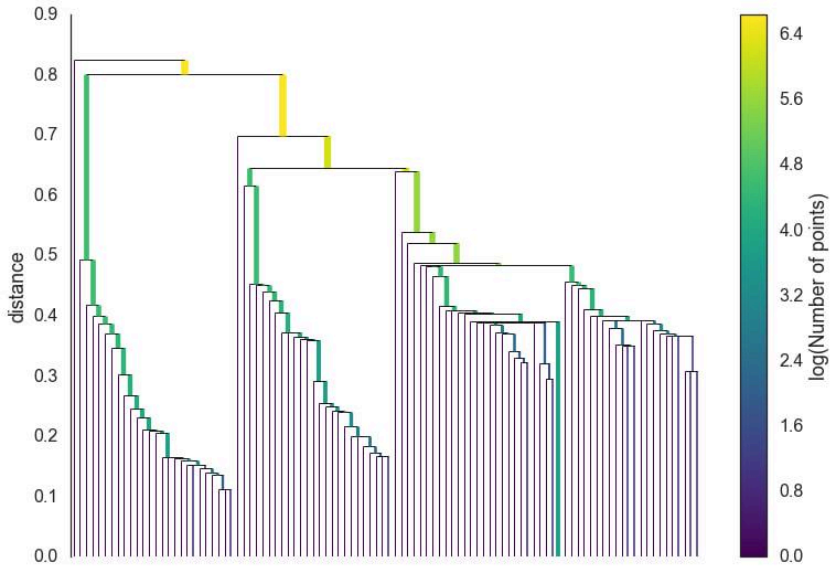
HDBSCAN. Алгоритм

5. Введём параметр $\hat{m}$ — минимальное число точек в кластере и новую шкалу для дендрограммы $\lambda = 1/d_{mreach}$.
6. Будем рассматривать дендрограмму снизу вверх по мере убывания λ (возрастания d_{mreach}). Мы скажем, что множество узлов C_i, получающееся рассмотрением связного поддерева в дендрограмме на высоте λ, является кластером, если оно содержит хотя бы $\hat{m}$ вершин. Максимальное такое λ назовем $\lambda_{i,\text{death}}$.
7. Минимальную величину λ, на которой поддерево C_t распадается на два кластера C_i и C_j, назовём $\lambda_{i,\text{birth}} = \lambda_{j,\text{birth}} = \lambda_{t,\text{death}}$.
8. Разбиваем точки на кластеры так, чтобы

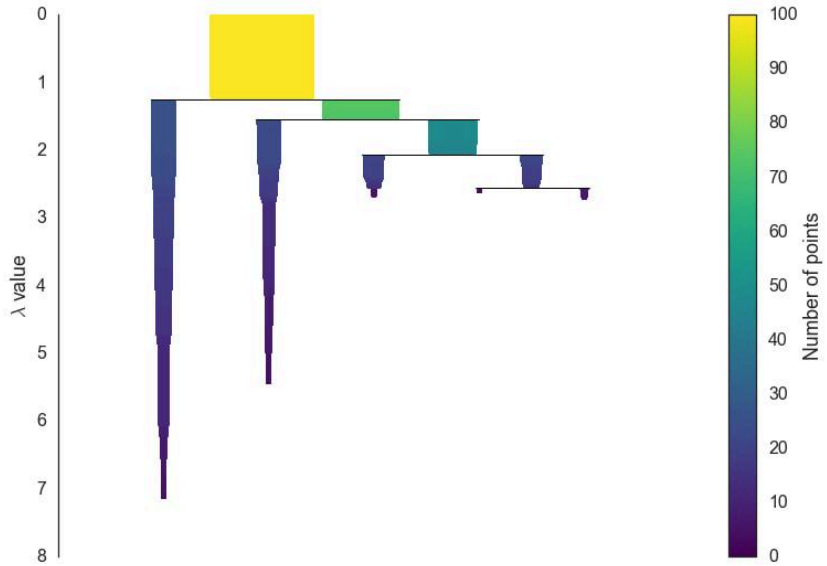
$$\max \sum_{i \in C} \int_{\lambda_{i,\text{birth}}}^{\lambda_{i,\text{death}}} p_{i,\lambda} d\lambda,$$

   где $p_{i,\lambda}$ — это число точек в кластере C_i на уровне λ.
9. Не попавшие в разбиение точки объявляем выбросами.

HDBSCAN. Иллюстрация



(a) Дендрограмма



(b) Сжатая дендрограмма

CLIQUE

CLIQUE (CLustering In QUEst) — это алгоритм, который комбинирует в себе идеи сеточного подхода к кластеризации и подхода, основанного на подпространствах, для обнаружения плотных кластеров в подпространствах максимальной размерности.

Мотивация: В данных с большим количеством признаков «проклятие размерности» приводит к тому, что данные становятся очень разреженными. Кластеры часто существуют только в пределах небольшого подмножества признаков, а в других измерениях точки могут быть распределены случайным образом. CLIQUE создан для решения именно этой проблемы.

CLIQUE. Определения

У алгоритма два входных параметра:

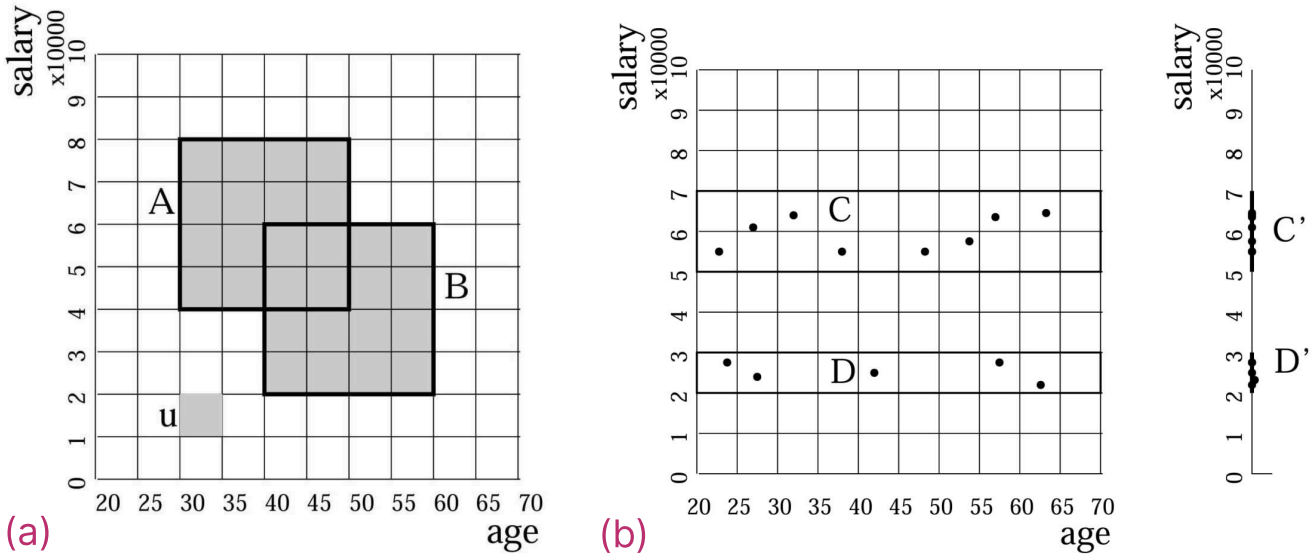
- ξ — параметр сетки;
- τ — пороговое значение плотности.

- Пространство признаков разбивается на ξ равных частей по каждому признаку. Пересечение одного интервала из каждого измерения называется **единицей** (unit).
- **Селективность** (selectivity) единицы определяется как доля общего количества точек данных, содержащихся в этой единице.
- **Плотная единица** (dense unit) — единица считается плотной, если ее селективность превышает пороговое значение плотности τ.
- **Кластер** — максимальное множество связанных (т.е. имеющих общую грань) плотных единиц в одном и том же подпространстве.

CLIQUE. Описание кластеров

- CLIQUE генерирует описания кластеров в виде выражений в **дизъюнктивной нормальной форме** (ДНФ), покрывая кластер минимальным количеством максимальных, возможно перекрывающихся, прямоугольников и описывая кластер как объединение этих прямоугольников.
- Кроме того, последний шаг CLIQUE принимает в качестве входных данных покрытие для каждого кластера и находит минимальное покрытие, определяемое с точки зрения количества максимальных областей (прямоугольников), необходимых для покрытия кластера.

CLIQUE. Иллюстрация определений



- График (a): сетка 10×10, единица — u, кластер $A \cup B$, его мин. описание в виде ДНФ

$$((30 \leq \text{age} < 50) \wedge (4 \leq \text{salary} < 8)) \vee ((40 \leq \text{age} < 60) \wedge (2 \leq \text{salary} < 6)).$$

- График (b): $\tau = 20\%$, ни одна двумерная единица не является плотной, и в исходном пространстве данных нет кластеров. Однако если спроецировать точки на измерение зарплаты, то получим три одномерные плотные единицы. Две из них соединены, поэтому в одномерном подпространстве зарплаты есть два кластера.

CLIQUE. 3 основных этапа

1. **Поиск плотных подпространств**
   - Представляем данные как ξ-мерную сетку;
   - Находим **плотные ячейки** (где точек $>$ порога τ);
   - Поиск плотных единиц в подпр-ах высокой размерности ведется снизу вверх, начиная с 1-мерных подпр-в. Используем принцип **монотонности**: «Если k-мерная ячейка плотная, то все её (k-1)-мерные проекции тоже плотные».
2. **Формирование кластеров**
   - После того как найдены все плотные k-мерные единицы в определенном подпространстве, алгоритм объединяет их в кластеры;
   - Строится граф , где вершины — это плотные единицы, а ребра соединяют соседние единицы;
   - Все связанные плотные единицы объединяются в один кластер. Для этого используется алгоритм поиска в глубину (DFS).
3. **Генерация описаний**
   - Покрываем кластер **минимальным количеством максимальных прямоугольников**;
   - Строим **минимальное покрытие**: удаляются те прямоугольники, которые полностью покрываются другими, более крупными прямоугольниками из покрытия;
   - Записываем результат в виде **ДНФ-формул**.

CLIQUE. Плюсы и минусы

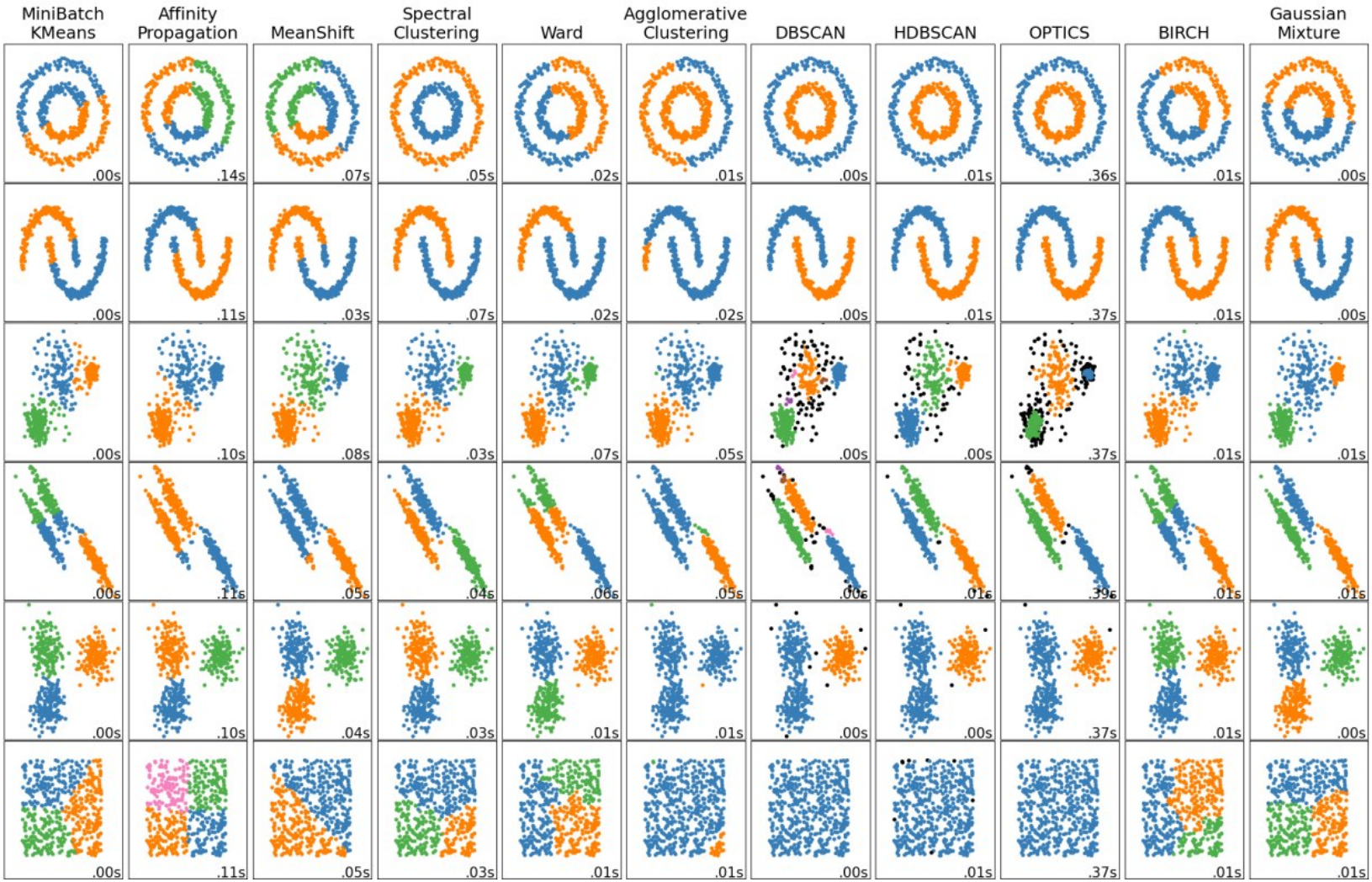
Преимущества:

- Хорошо работает для данных с большой размерностью;
- Не требует знания числа кластеров заранее, они формируются автоматически из плотных ячеек;
- Каждый кластер можно описать в логической форме;
- Алгоритм устойчив к пропущенным значениям во входных данных.

Недостатки:

- Результат сильно зависит от входных параметров: размера сетки и пороговой плотности;
- Размер сетки растёт экспоненциально с размерностью.

Сравнение методов



Функционалы качества кластеризации

Задачу кластеризации можно ставить как задачу *дискретной оптимизации*: необходимо так приписать номера кластеров y_i объектам $\boldsymbol{x}_i$, чтобы значение выбранного функционала качества приняло наилучшее значение.

- **Среднее внутрикластерное расстояние**:

$$F_0 = \frac{\sum_{i<j} \mathbf{I}_{\{y_i=y_j\}} \rho(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sum_{i<j} \mathbf{I}_{\{y_i=y_j\}}} \to \min .$$

- **Среднее межкластерное расстояние**:

$$F_1 = \frac{\sum_{i<j} \mathbf{I}_{\{y_i\neq y_j\}} \rho(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sum_{i<j} \mathbf{I}_{\{y_i\neq y_j\}}} \to \max .$$

На практике вычисляют отношение пары функционалов, чтобы учесть как межкластерные, так и внутрикластерные расстояния:

$$F_0/F_1 \to \min$$

.

Функционалы качества кластеризации. Silhouette

- Пусть a и b есть среднее расстояние между наблюдением и всеми другими точками в том же кластере/в следующем ближайшем кластере
- Коэффициент силуэта для наблюдения есть

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

- Для выборки коэффициент силуэта задается средним значением коэффициентов каждого наблюдения
- Значения в интервале [-1; 1]. Чем больше, тем лучше. Если коэффициент близок к 0, то это свидетельство в сторону того, что кластеры "накладываются"друг на друга

Функционалы качества кластеризации. Calinski-Harabasz Index (Variance Ratio Criterion)

- Индекс Калинского–Харабэсза определяется как отношение между средней межкластерной дисперсией и средней дисперсией внутри кластеров

$$s = \frac{\operatorname{tr}(B)}{k-1} \Big/ \frac{\operatorname{tr}(W)}{n-k}$$

- Тут k – число кластеров, а матрицы B и W имеют вид

$$W = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T, \quad B = \sum_{q=1}^{k} n_q (c_q - c_x)(c_q - c_E)^T$$

где C_q – наблюдения из кластера q, c_q – центр кластера q, c_x – центр всего набора данных X, а n_q – число наблюдений в кластере q.

- Значение индекса тем больше, чем разделеннее кластеры и чем более сгруппированы в кластерах наблюдения. Ну и применять для случая сферических/эллиптических кластеров

Функционалы качества кластеризации. Davies-Bouldin Index

- Индекс Дэвиса–Болдина оценивает среднее "сходство"между кластерами, где сходство есть мера, которая сравнивает расстояние между кластерами с размером самих кластеров
- Пусть s_i есть среднее расстояние между каждой точкой кластера i и центроидом этого кластера, а d_{ij} есть расстояние между центроидами кластеров. Определим схожесть между кластерами следующим образом

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

- Тогда индекс имеет вид

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij}$$

- Чем индекс меньше, тем лучше. Близок к 0 – отличное разделение. Недостаток тот же, что и у индекса Калинского–Харабэсза