Методы машинного обучения

Лекция 12

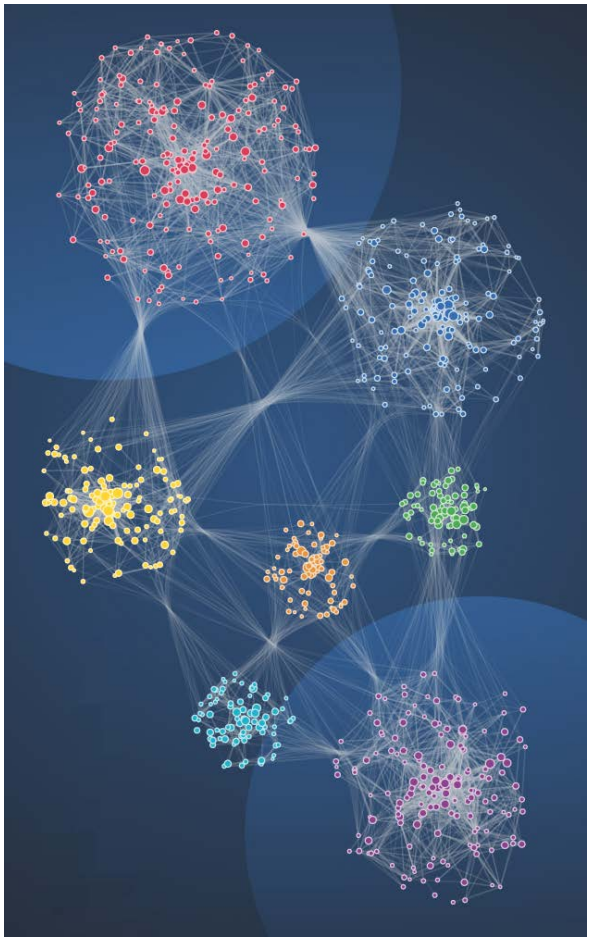
Кластеризация

Кластеризация

Кластеризация (clustering/*cluster analysis*) — задача группировки множества объектов на подмножества (**кластеры**) таким образом, чтобы объекты из одного кластера были более похожи друг на друга, чем на объекты из других кластеров по какому-либо критерию. Это может быть информация о попарном сходстве объектов. Функционалы качества могут определяться по-разному, например, как отношение средних межкластерных и внутрикластерных расстояний.

Исторически возникла из задачи группировки схожих объектов в единую структуру (кластер) с последующим выявлением общих черт.
Примеры областей применения:

- Экономическая география: по физико-географическим и экономическим показателям разбить страны мира на группы схожих по экономическому положению государств;
- Финансовая сфера: по сводкам банковских операций выявить группы «подозрительных», нетипичных банков, сгруппировать остальные по степени близости проводимой стратегии;
- Маркетинг: по результатам маркетинговых исследований среди множества потребителей выделить характерные группы по степени интереса к продвигаемому продукту;
- Социология: по результатам социологических опросов выявить группы общественных проблем, вызывающих схожую реакцию у общества, а также характерные фокус-группы населения.



Формальная постановка задачи

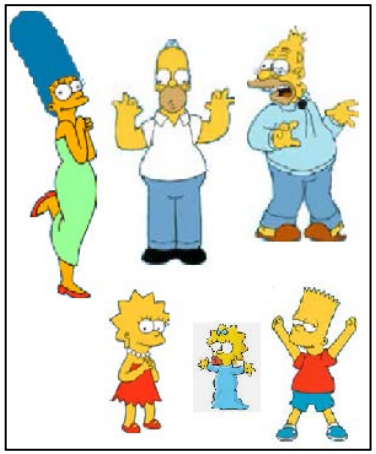
Пусть X — множество объектов, Y — множество идентификаторов (номеров, имён, меток) кластеров. Задана функция расстояний между объектами $\rho(x, x')$. Имеется конечная обучающая выборка объектов $X^m = \{x_1, \dots, x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ, а объекты разных кластеров существенно отличались. При этом каждому объекту $x_i \in X^m$ приписывается номер кластера y_i.

Алгоритм кластеризации — это функция $\alpha: X \to Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$. Множество Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

Принципиальная неоднозначность

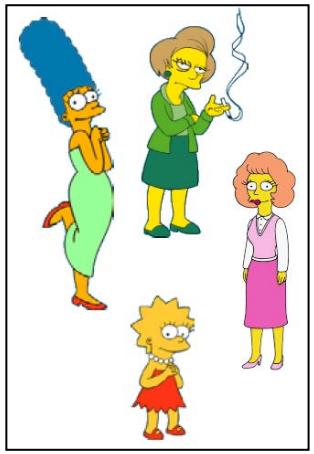
Решение задачи кластеризации принципиально неоднозначно:

- Не существует однозначно наилучшего критерия качества кластеризации.
- Число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным критерием.
- Результат кластеризации существенно зависит от метрики, выбор которой, как правило, также субъективен и определяется экспертом.



Семья

Сотрудники



Женщины

Мужчины

Цели кластеризации

- **Улучшение понимания данных** за счет выявления структурных групп;
- **Классификация объектов.** Попытка понять зависимости между объектами путем выявления их кластерной структуры. Разбиение выборки на группы схожих объектов упрощает дальнейшую обработку данных, позволяет применить к каждому кластеру свой метод анализа (стратегия «разделяй и властвуй»). В данном случае стремятся уменьшить число кластеров для выявления наиболее общих закономерностей;
- **Сжатие данных.** Можно сократить размер исходной выборки, взяв один или несколько наиболее типичных представителей каждого кластера. Здесь важно наиболее точно очертить границы каждого кластера, их количество не является важным критерием;
- **Обнаружение новизны (обнаружение шума).** Выделение объектов, которые не подходят по критериям ни в один кластер. Обнаруженные объекты в дальнейшем обрабатывают отдельно.

Основные этапы

- Отбор выборки объектов для кластерного анализа.
- Определение множества переменных, по которым будут оцениваться объекты в выборке, то есть признакового пространства.
- Выбор и вычисление значений меры сходства (или различия) между объектами.
- Применение метода кластерного анализа для создания групп сходных объектов.
- Проверка достоверности и представление результатов анализа.

Типы входных данных

- **Признаковое описание объектов.** Каждый объект описывается набором своих характеристик/признаков (*features*). Признаки могут быть как числовыми, так и нечисловыми (категориальными);
- **Матрица расстояний между объектами.** Каждый объект описывается расстоянием до всех объектов из обучающей выборки.

Фундаментальные требования, предъявляемые к данным:

- **Однородность (uniformity) -** требует, чтобы все кластеризуемые сущности были одной природы, описывались сходным набором характеристик, т.е. значения всех атрибутов должны быть сравнимыми для всех данных.
- **Полнота (comprehensiveness)** — набор данных должен содержать достаточное количество параметров или признаков, чтобы не осталось неохваченных пограничных случаев.

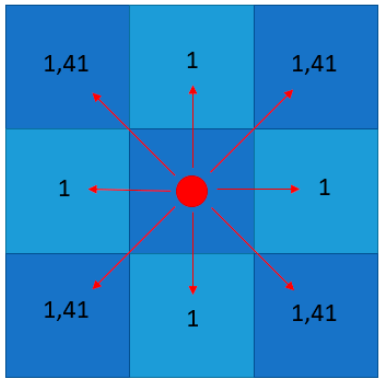
Меры расстояний

- Евклидово расстояние: $\rho(x,x') = \sqrt{\sum_i^n (x_i - x_i')^2}$
- Квадрат евклидова расстояния: $\rho(x,x') = \sum_i^n (x_i - x_i')^2$
- Расстояние городских кварталов L1 (манхэттенское расстояние): $\rho(x,x') = \sum_i^n |x_i - x_i'|$
- Расстояние Чебышева: $\rho(x,x') = \max(|x_i - x_i'|)$
- Степенное расстояние: $\rho(x,x') = \sqrt[r]{\sum_i^n (x_i - x_i')^p}$

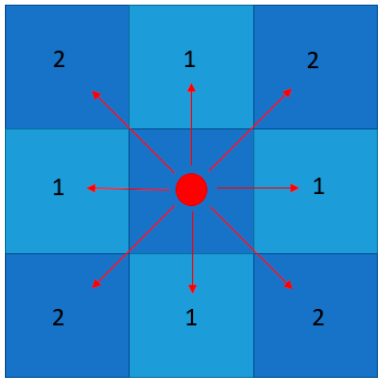
где r и p – параметры,

определяемые пользователем.

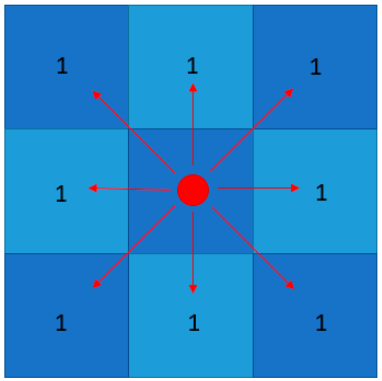
Евклидово расстояние



Расстояние L1



Расстояние Чебышёва



Кластерный центроид

Центром кластера C_k (**центроидом**) называется геометрический центр точек *k*-ого класса в евклидовом пространстве:

$$\mathrm{c}_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

где $|C_k|$ – число точек в *k*-ом кластере, $k = \overline{1, K}$, *K* – число кластеров.

Дисперсия кластера C_k – мера рассеяния точек в пространстве относительно центра кластера:

$$D_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} \rho^2(x_i, c_k)$$

Радиус кластера C_k – мера рассеяния точек относительно центра кластера – максимальное расстояние до центра кластера:

$$R_k = \max_{x_i \in C_k} \rho(x_i, c_k)$$

Меры качества кластеризации

Задачу кластеризации можно ставить как задачу дискретной оптимизации, т.е. необходимо так присвоить номера кластеров y_i объектам x_i, чтобы значение выбранного функционала качества приняло наилучшее значение.

Функционал качества:

- Среднее внутрикластерное расстояние должно быть как можно меньше:

$$F_0 = \frac{\sum_{i<j}[y_i = y_j]\rho(x_i, x_j)}{\sum_{i<j}[y_i = y_j]} \to \min$$

- Среднее межкластерное расстояние должно быть как можно больше:

$$F_1 = \frac{\sum_{i<j}[y_i \neq y_j]\rho(x_i, x_j)}{\sum_{i<j}[y_i \neq y_j]} \to \max$$

Если алгоритм кластеризации вычисляет центры кластеров μ_y, $y \in Y$, то можно определить функционалы:

- Сумма средних внутрикластерных расстояний должна быть как можно меньше:

$$\Phi_0 = \sum_{y \in Y} \frac{1}{|K_y|} \sum_{i:\ y_i = y} \rho^2(x_i, \mu_y) \to \min$$

где $K_y = \{x_i \in X^l | y_i = y\}$ кластер с номером y, состоящий из $|K_y|$ точек.

- Сумма межкластерных расстояний должна быть как можно больше:

$$\Phi_1 = \sum_{y \in Y} \rho^2(\mu_y, \mu) \to \max$$

где μ - центр масс всей выборки.

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

$$\frac{F_0}{F_1} \to min, \text{ или } \frac{\Phi_0}{\Phi_1} \to min$$

Методы оценки качества кластеризации

Принято выделять две группы методов оценки качества кластеризации:

- **Внешние** (англ. *External*) меры основаны на сравнении результата кластеризации с априори известным разделением на классы. Данные меры используют дополнительные знания о кластеризуемом множестве: распределение по кластерам, количество кластеров и т.д.
- **Внутренние** (англ. *Internal*) меры отображают качество кластеризации только по информации в данных и не используя внешней информации.

Внешние меры оценки качества – Таблица сопряженности

Дано множество S из n элементов, разделение на классы $X = \{X_1, X_2, \ldots, X_r\}$, и полученное разделение на кластеры $Y = \{Y_1, Y_2, \ldots, Y_s\}$, совпадения между X и Y могут быть отражены в таблице сопряженности $[n_{ij}]$, где каждое n_{ij} обозначает число объектов, входящих как в X_i, так и в Y_j : $n_{ij} = |X_i \cap Y_j|$.

$X \setminus Y$	Y_1	Y_2	$\ldots$	Y_s	Sums
X_1	n_{11}	n_{12}	$\ldots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\ldots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\ldots$	n_{rs}	a_r
Sums	b_1	b_2	$\ldots$	b_s	n

Пусть $p_{ij} = \frac{n_{ij}}{n}, p_i = \frac{a_i}{n}, p_j = \frac{b_j}{n}$.

Также рассмотрим пары (x_i, x_j) из элементов кластеризуемого множества *X*. и подсчитаем количество пар, в которых:

- Элементы принадлежат одному кластеру и одному классу — *TP*
- Элементы принадлежат одному кластеру, но разным классам — *FP*
- Элементы принадлежат разным кластерам, но одному классу — *FN*
- Элементы принадлежат разным кластерам и разным классам — *TN*

Внешние меры оценки качества - Индексы

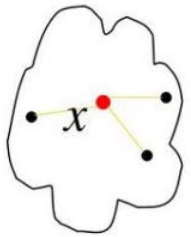
- **Индекс Rand** $$Rand = \frac{TP + TN}{TP + TN + FP + FN}$$
- **Индекс Жаккара (Jaccard Index)** $$Jaccard = \frac{TP}{TP + FN + FP}$$
- **Индекс Фоулкса – Мэллова (Fowlkes-Mallows Index)** $$FM = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}}$$
- **Индекс Phi** $$\Phi = \frac{TP \times TN - FN \times FP}{(TP + FN)(TP + FP)(FN + TN)(FP + TN)}$$
- **Entropy (Энтропия)** $$E = -\sum_i p_i (\sum_j \frac{p_{ij}}{p_i} log(\frac{p_{ij}}{p_i}))$$
- **Purity (чистота)** $$P = \sum_i \max_j p_{ij}$$
- **F-мера** $$F = \sum_j p_j \max_i \left[2 \frac{p_{ij}}{p_i} \frac{p_{ij}}{p_j} / (\frac{p_{ij}}{p_i} + \frac{p_{ij}}{p_j})\right]$$
- **Variation of Information (изменение информации)**

$$VI = -\sum_i p_i \log p_i - \sum_i p_j log p_j - 2 \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{p_i p_j}$$

Внутренние меры оценки качества

Данные меры оценивают качество структуры кластеров опираясь только непосредственно на нее, не используя внешней информации.

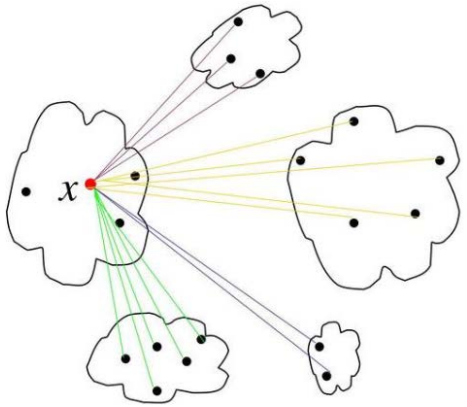
- **Компактность кластеров (Cluster Cohesion)**



*Within cluster Sum of Squares (**WSS**)* → $\boldsymbol{min}$

$$WSS = \sum_{j=1}^{M} \sum_{i=1}^{|C_j|} (x_{ij} - \overline{x_j})^2$$, где M — количество кластеров.

- **Отделимость кластеров (Cluster Separation)**



*Between cluster Sum of Squares (**BSS**)* → $\boldsymbol{max}$

$$\boldsymbol{BSS} = \sum_i |C_i|(m - m_i)^2$$

где C_i – *i*-ый кластер, m_i-центроид C_i , m – общий центр.

- **Силуэт (Silhouette)**

$$Sil(C) = \frac{1}{N} \sum_{c_k \in C} \sum_{x_i \in c_k} \frac{b(x_i, c_k) - a(x_i, c_k)}{max\{a(x_i, c_k), b(x_i, c_k)\}},$$

$$a(x_i, c_k) = \frac{1}{|c_k|} \sum_{x_j \in c_k} \|x_i - x_j\|$$ - компактность.

$$b(x_i, c_k) = min_{c_l \in C \setminus c_k} \{\frac{1}{|c_l|} \sum_{x_j \in c_l} \|x_i - x_j\|\}$$ - отделимость.

Значение: $-1 \leq Sil(C) \leq 1$. Чем ближе к 1, тем лучше.

Внутренние меры оценки качества

- **Индекс Дэвиса-Болдуина (Davies–Bouldin Index)**

$$DB(C) = \frac{1}{K} \sum_{c_k \in C} \max_{c_l \in C \setminus c_k} \left\{ \frac{S(c_k) + S(c_l)}{\|\overline{c_k} - \overline{c_l}\|} \right\}, \quad S(c_k) = \frac{1}{|c_k|} \sum_{x_i \in c_k} \|x_i - \overline{c_k}\|$$

- **Индекс Calinski–Harabasz**

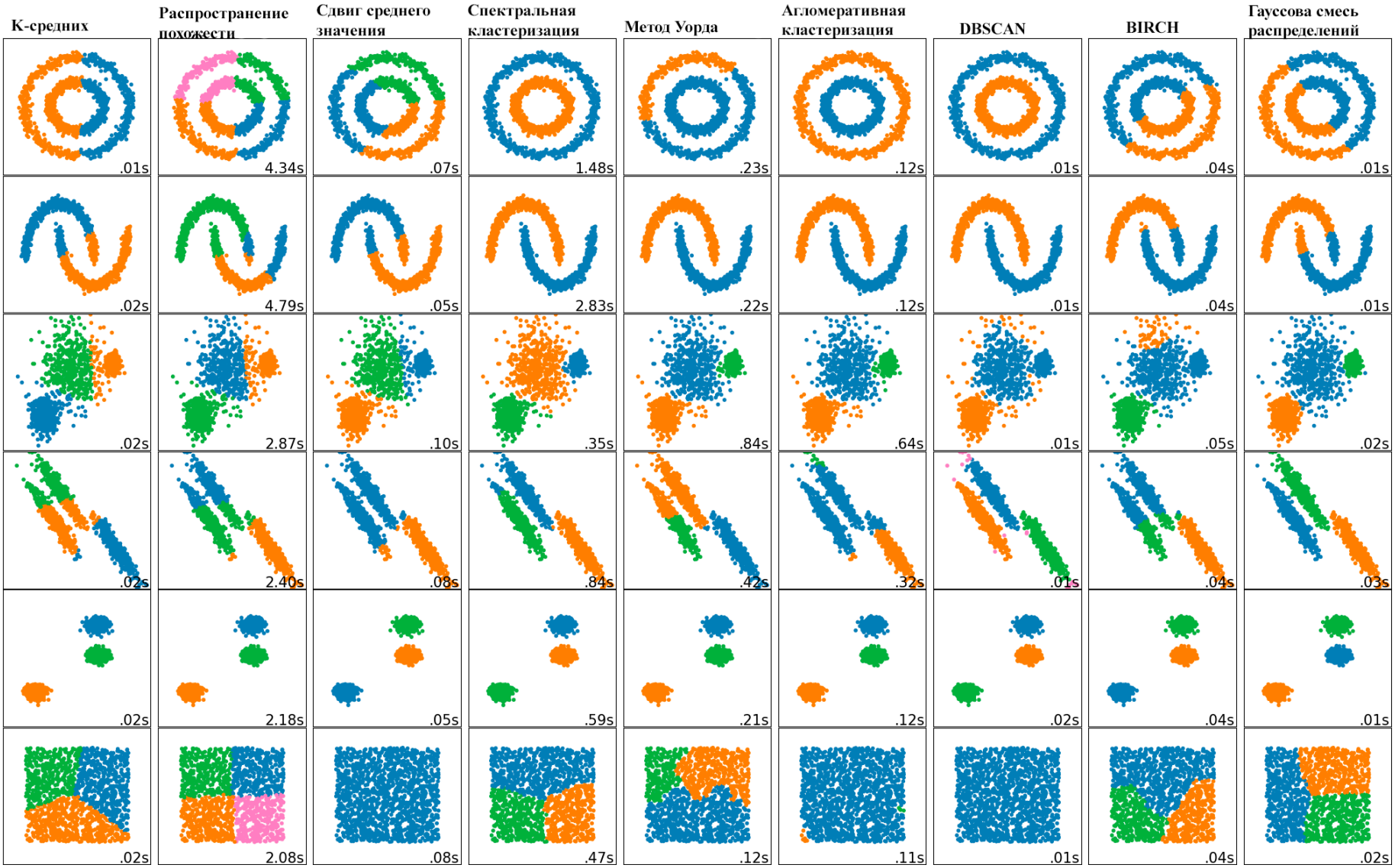
$$CH(C) = \frac{N - K}{K - 1} \cdot \frac{\sum_{c_k \in C} |c_k| \cdot \|\overline{c_k} - \overline{X}\|}{\sum_{c_k \in C} \sum_{x_i \in c_k} \|x_i - \overline{c_k}\|}$$

- **Score function**

$$SF(C) = 1 - \frac{1}{e^{e^{bcd(C) - wcd(C)}}},$$

$$bcd(C) = \frac{\sum_{c_k \in C} |c_k| \cdot \|\overline{c_k} - \overline{X}\|}{N \times K}, \quad wcd(C) = \sum_{c_k \in C} \frac{1}{|c_k|} \sum_{x_i \in c_k} \|x_i - \overline{c_k}\|$$

Алгоритмы кластеризации



Типы алгоритмов кластеризации

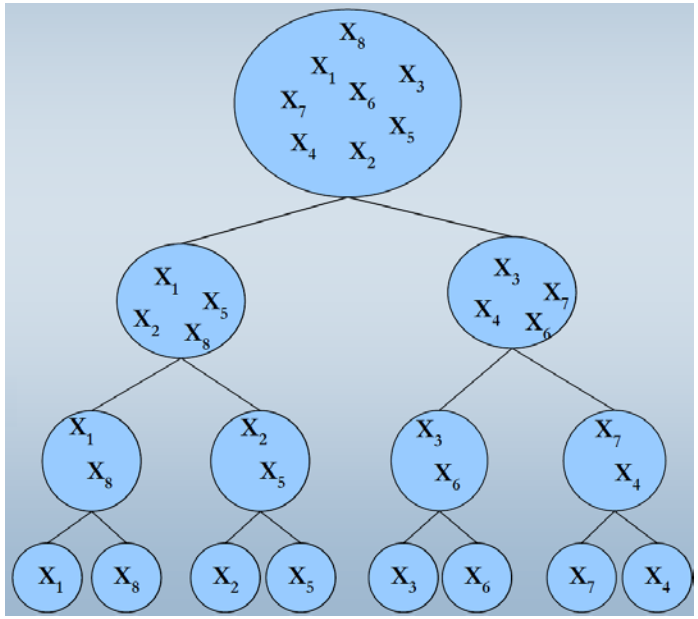
Алгоритмы кластеризации можно разделить на 4 типа:

1. **Кластерный центроид.** Он объединяет данные в кластеры, основываясь на заранее заданных условиях и характеристиках. k-средних — наиболее популярный алгоритм из этой категории.
2. **Кластеризация на основе плотности.** В этом типе используется алгоритм, который соединяет области с высокой плотностью в кластеры, создавая распределения произвольной формы.
3. **Кластеризация на основе распределений.** Алгоритм этого типа предполагает гауссовские распределения данных, которые далее объединяются в различные варианты того же распределения.
4. **Иерархические алгоритмы.** В этом алгоритме строится иерархическая древо кластеров. Количество кластеров можно менять, объединяя их на определённом уровне древа.

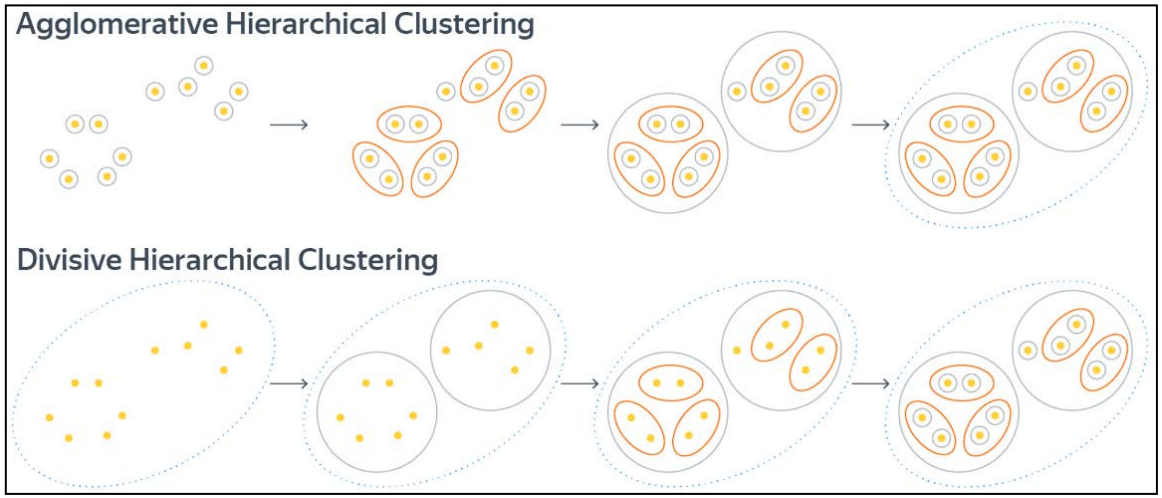
Иерархическая кластеризация

Среди алгоритмов иерархической кластеризации выделяются два основных типа:

1. Нисходящая кластеризация (divisive): Работает по принципу «сверху-вниз»: в начале, все объекты принадлежат одному кластеру. В ходе итеративного процесса крупные кластеры разделяются на более мелкие. Такие задачи называются задачами таксономии. При этом, получается дерево кластеров (дендрограмма).



2. Восходящая кластеризация (agglomerative): Изначально каждый элемент множества является отдельным кластером. Процесс образования новых кластеров заключается в объединение некоторых кластеров в один на основе заданного расстояния. В итоге итеративного объединения получаем дерево, которое сходится к одному кластеру.



Алгоритмы иерархической кластеризации - расстояние между кластерами

- Одиночная связь (расстояния ближайшего соседа) - расстояние между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами (ближайшими соседями) в различных кластерах.
- Полная связь (расстояние наиболее удаленных соседей) - расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах (т.е. наиболее удаленными соседями).
- Невзвешенное попарное среднее - расстояние между двумя различными кластерами вычисляется как среднее расстояние между всеми парами объектов в них.
- Взвешенное попарное среднее - метод идентичен методу невзвешенного попарного среднего, за исключением того, что при вычислениях размер соответствующих кластеров (т.е. число объектов, содержащихся в них) используется в качестве весового коэффициента.
- Невзвешенный центроидный метод - расстояние между двумя кластерами определяется как расстояние между их центрами тяжести.
- Взвешенный центроидный метод (медиана) - идентичен предыдущему, за исключением того, что при вычислениях используются веса для учета разницы между размерами кластеров.

Метод k-средних (k-means)

Алгоритм относится к классу эвристических EM-алгоритмов (Expectation-maximization), используемых в математической статистике для нахождения оценок максимального правдоподобия параметров вероятностных моделей.

Основная идея метода — итеративное повторение двух шагов:

- распределение объектов выборки по кластерам;
- пересчёт центров кластеров.

Алгоритм заключается в том, что он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров:

$$V = \sum_{k=1}^{K} \sum_{x_i \in C_k} \rho^2(x_i, c_k) \to \min_C$$

где K — число кластеров, C_k — полученные кластеры, c_k — центр кластера C_k, x_i вектор из кластера C_k.

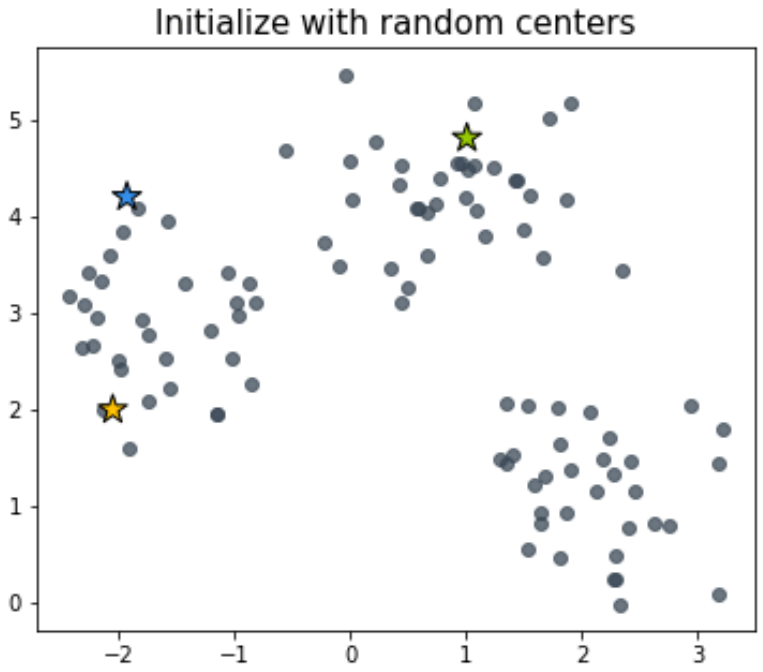
В начале работы алгоритма выбираются K случайных центров в пространстве признаков. Каждый объект выборки относят к тому кластеру, к центру которого объект оказался ближе по выбранной метрике.

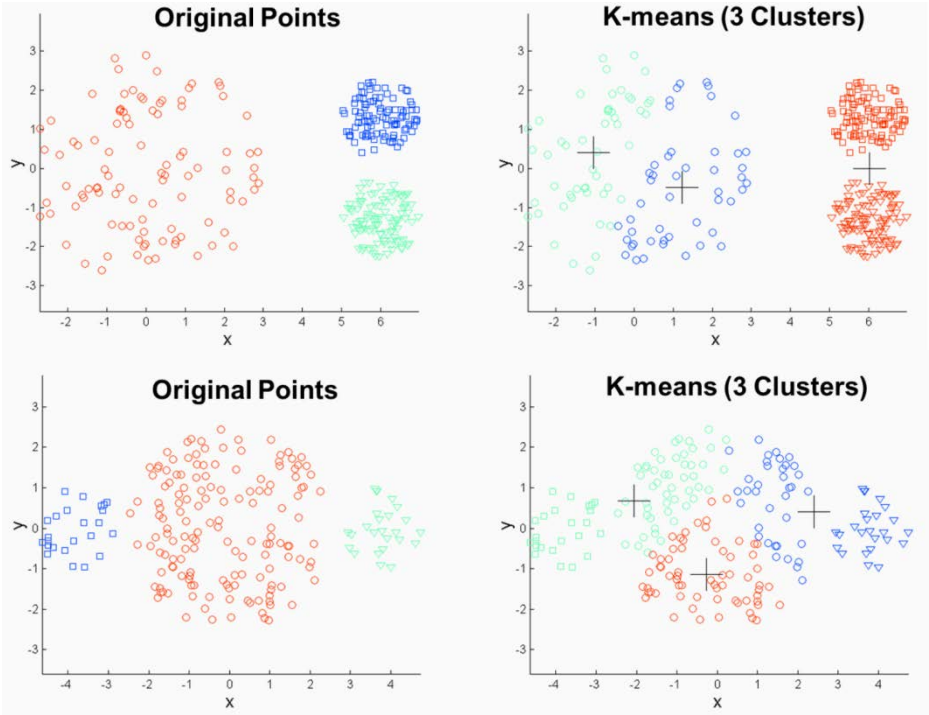
$$x_i \in C_k, \text{ если } \rho(x_i, c_k) = \min_{k=1,K}$$

Далее центры кластеров пересчитывают как среднее арифметическое векторов признаков всех вошедших в этот кластер объектов (то есть центр масс кластера). Как только мы обновили центры кластеров, объекты заново перераспределяются по ним, а затем можно снова уточнить положение центров. Процесс продолжается до тех пор, пока центры кластеров не перестанут меняться.

Недостатки метода k-средних

- Не гарантирует достижения глобального минимума суммарного квадратичного отклонения V, а только одного из локальных минимумов.
- Результат зависит от начального выбора центров кластеров $\{c_k^{(0)}\}$, а их оптимальный выбор неизвестен.
- Число кластеров K надо знать заранее.





Метод k-средних
Инициализация центров кластеров

- Метод Forgy. Из имеющегося набора данных случайным образом выбираются K наблюдений.
- Случайное разбиение. Каждому наблюдению на начальном этапе случайным образом присваивается номер кластера.
- Алгоритм k-means++. Из имеющегося набора данных случайным образом выбирается одна точка (первый центроид). Затем следующая точка выбирается из оставшихся с вероятностью, пропорционально зависящей от квадрата расстояния от точки до ближайшего центроида. Итерации повторяются до тех пор, пока не будут выбраны K центроидов.

Алгоритмы семейства FOREL

FOREL (Формальный Элемент) — алгоритм кластеризации, основанный на идее объединения в один кластер объектов в областях их наибольшего сгущения.

Цель: Разбить выборку на такое (заранее неизвестное) число таксонов (групп/кластеров), чтобы сумма расстояний от объектов кластеров до центров кластеров была минимальной по всем кластерам. То есть наша задача — выделить группы максимально близких друг к другу объектов, которые в силу гипотезы схожести и будут образовывать наши кластеры.

Входные данные

- Кластеризуемая выборка - может быть задана признаковыми описаниями объектов либо матрицей попарных расстояний между объектами.
- Параметр R — радиус поиска локальных сгущений (можно задавать как из априорных соображений, так и настраивать скользящим контролем).
- В модификациях возможно введение параметра k — количества кластеров.

Выходные данные

- Кластеризация на заранее неизвестное число таксонов.

Минимизируемый алгоритмом функционал качества

$$F = \sum_{j=1}^{k} \sum_{x \in K_j} \rho(x, W_j),,$$

где первое суммирование ведется по всем кластерам выборки, второе суммирование — по всем объектам x, принадлежащим текущему кластеру K_j, а W_j — центр текущего кластера, $\rho(x, y)$ — расстояние между объектами.

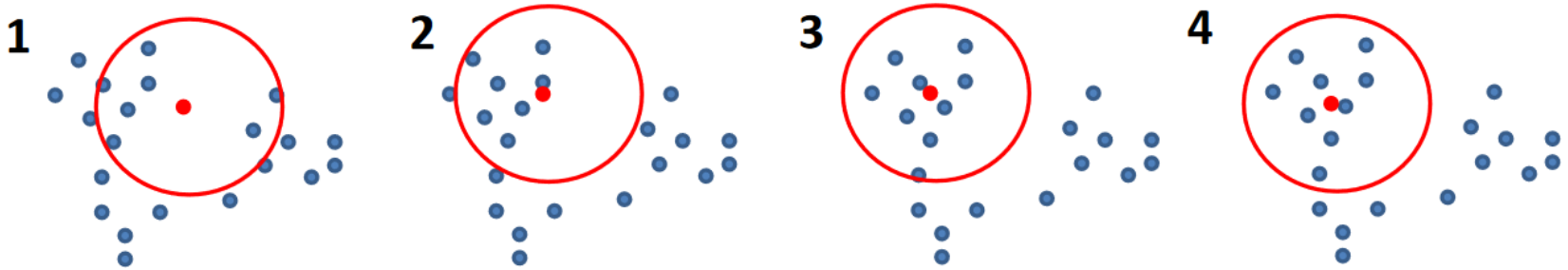
FOREL- Алгоритм

Алгоритм

1. Случайно выбираем текущий объект из выборки;
2. Помечаем объекты выборки, находящиеся на расстоянии менее, чем R от текущего;
3. Вычисляем их центр тяжести, помечаем этот центр как новый текущий объект;
4. Повторяем шаги 2-3, пока новый текущий объект не совпадет с прежним;
5. Помечаем объекты внутри сферы радиуса R вокруг текущего объекта как кластеризованные, выкидываем их из выборки;
6. Повторяем шаги 1-5, пока не будет кластеризована вся выборка.

Выбор центра тяжести:

- В линейном пространстве — центр масс;
- В метрическом пространстве — объект, сумма расстояний до которого минимальна, среди всех внутри сферы;
- Объект, который внутри сферы радиуса R содержит максимальное количество других объектов из всей выборки (медленно);
- Объект, который внутри сферы маленького радиуса содержит максимальное количество объектов (из сферы радиуса R).



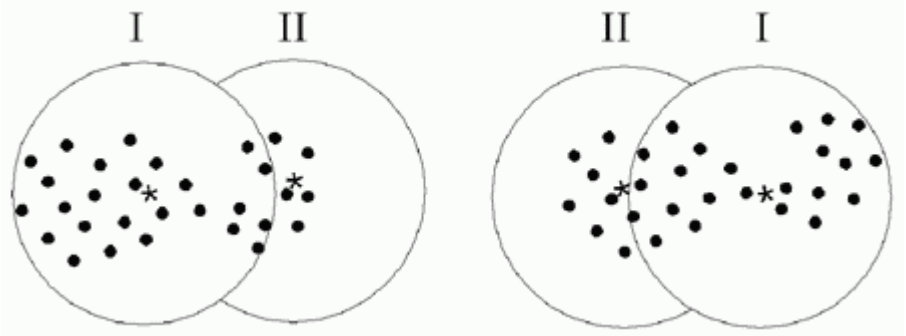
FOREL- Преимущества и Недостатки

Преимущества

- Точность минимизации функционала качества (при удачном подборе параметра R);
- Наглядность визуализации кластеризации;
- Сходимость алгоритма;
- Возможность операций над центрами кластеров — они известны в процессе работы алгоритма;
- Возможность подсчета промежуточных функционалов качества, например, длины цепочки локальных сгущений;
- Возможность проверки гипотез схожести и компактности в процессе работы алгоритма.

Недостатки

- Относительно низкая производительность (решается введением функции пересчета поиска центра при добавлении 1 объекта внутрь сферы);
- Плохая применимость алгоритма при плохой разделимости выборки на кластеры;
- Неустойчивость алгоритма (зависимость от выбора начального объекта);
- Произвольное по количеству разбиение на кластеры;
- Необходимость априорных знаний о ширине (диаметре) кластеров.



DBSCAN

*Density-based spatial clustering of applications with noise (***DBSCAN) - Основанная на плотности пространственная кластеризация для приложений с шумами.**

Если дан набор точек в некотором пространстве, алгоритм группирует вместе точки, которые тесно расположены (точки со многими близкими соседями), помечая как выбросы точки, которые находятся одиноко в областях с малой плотностью (ближайшие соседи которых лежат далеко). **DBSCAN** развивает идею кластеризации с помощью выделения связных компонент.

Плотность в DBSCAN определяется в окрестности каждого объекта выборки x_i как количество других точек выборки в шаре $B(\varepsilon, x_i)$. Кроме радиуса ε окрестности в качестве гиперпараметра алгоритма задается порог $\boldsymbol{MinPts}$ по количеству точек в окрестности.

Все объекты выборки делятся на три типа:

- **внутренние / основные точки** (**core points**),
- **граничные/** *достижимые по плотности* (**border points**)
- **шумовые/** *выпадающие* **точки** (**noise points**).

К основным относятся точки, в окрестности которых больше $MinPts$ объектов выборки. К граничным — точки, в окрестности которых есть основные, но общее количество точек в окрестности меньше $MinPts$. Шумовыми называют точки, в окрестности которых нет основных точек и в целом содержится менее $MinPts$ объектов выборки.

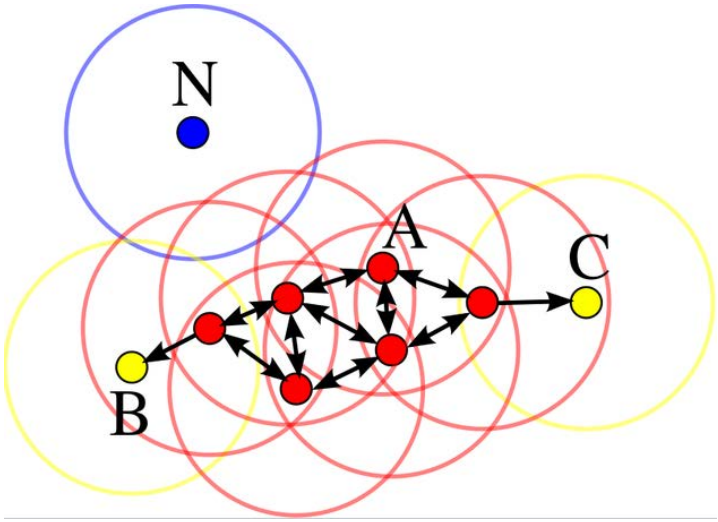
DBSCAN - Алгоритм

1. Выделяются основные точки.
2. Основные точки, у которых есть общая окрестность, соединяются ребром. Говорят, что эти точки *прямо достижимы.*
3. В полученном графе выделяются компоненты связности.
4. Каждая граничная точка (*достижимая* точка*)* относится к тому кластеру, в который попала ближайшая к ней основная точка.
5. Шумовые точки (не достижимые из основных точек) убираются из рассмотрения и не приписываются ни к какому кластеру.

Если *A* является основной точкой, то она формирует *кластер* вместе со всеми точками (основными или неосновными), достижимые из этой точки. Каждый кластер содержит по меньшей мере одну основную точку. Неосновные точки могут быть частью кластера, но они формируют его «край», поскольку не могут быть использованы для достижения других точек.

Две точки *P* и *Q* связаны по плотности, если имеется точка *O*, такая что и *P*, и *Q* достижимы из *O*. Кластер удовлетворяет двум свойствам:

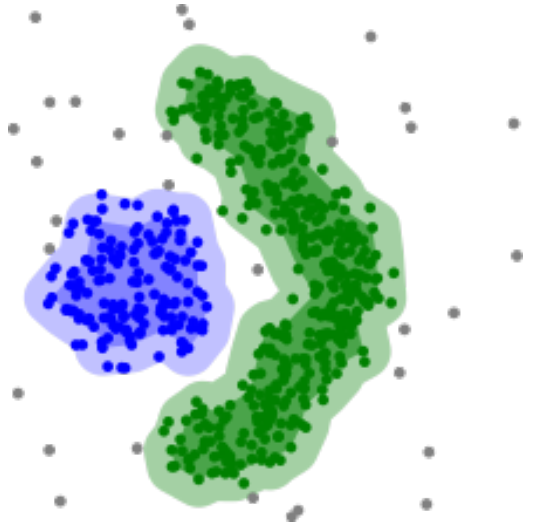
- Все точки в кластере попарно связны по плотности.
- Если точка достижима по плотности из какой-то точки кластера, она также принадлежит кластеру.



DBSCAN - Преимущества и Недостатки

Преимущества

- Не требует спецификации числа кластеров в данных.
- Может найти кластеры произвольной формы.
- Имеет понятие шума и устойчив к выбросам.
- Требует лишь двух параметров и большей частью нечувствителен к порядку выбора точек.
- Параметры minPts и ε могут быть установлены экспертами в рассматриваемой области, если данные хорошо понимаются.



Недостатки

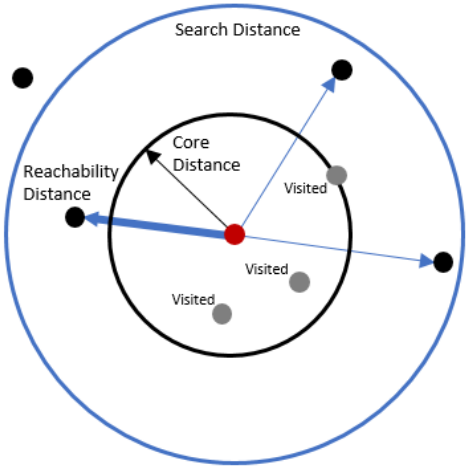
- Не полностью однозначен — краевые точки, которые могут быть достигнуты из более чем одного кластера, могут принадлежать любому из этих кластеров, что зависит от порядка просмотра точек.
- Качество DBSCAN зависит от измерения расстояния, используемого для подсчета количества других точек выборки в шаре $B(\varepsilon, x_i)$.
- Не может хорошо кластеризовать наборы данных с большой разницей в плотности, поскольку не удается выбрать приемлемую для всех кластеров комбинацию minPts и ε.
- Если есть сложности с пониманием особенностей данных и масштаба, выбор порога расстояния ε может оказаться проблематичным.

OPTICS

*Ordering points to identify the clustering structure (***OPTICS) - Упорядочение точек для обнаружения кластерной структуры** — это алгоритм нахождения кластеров в пространственных данных на основе плотности. Основная идея алгоритма похожа на DBSCAN, но алгоритм предназначен для избавления от одной из главных слабостей алгоритма DBSCAN— проблемы обнаружения содержательных кластеров в данных, имеющих различные плотности. Чтобы это сделать, точки имеющихся данных (линейно) упорядочиваются так, что пространственно близкие точки становятся соседними в упорядочении. Кроме того, для каждой точки запоминается специальное расстояние, представляющее плотность, которую следует принять для кластера, чтобы точки принадлежали одному кластеру.

Основное расстояние, которое описывает расстояние до *MinPts*-ой ближайшей точки:

$$\text{core-dist}_{\varepsilon,MinPts} = \begin{cases} \text{UNDEFINED} & |N_\varepsilon(p)| < MinPts \\ MinPts\text{-th}N_\varepsilon(p) & |N_\varepsilon(p)| \geqslant MinPts \end{cases}$$



Достижимое расстояние точки *o* от точки *p:*

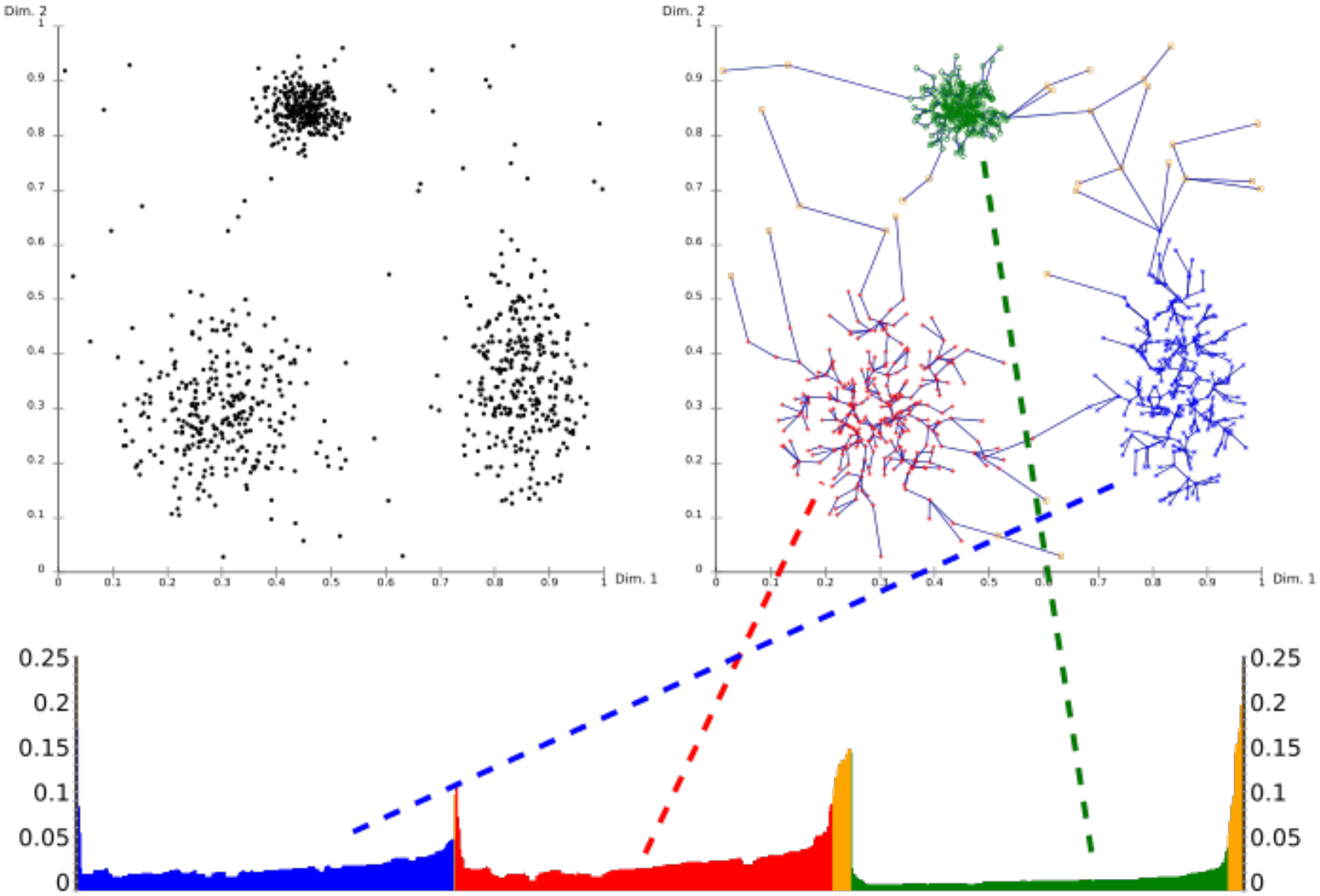
$$\text{reachability-dist}_{\varepsilon,MinPts}(o,p) = \begin{cases} \text{UNDEFINED} & |N_\varepsilon(p)| < MinPts \\ \max(\text{core-dist}_{\varepsilon,MinPts}(p), \text{dist}(p,o)) & |N_\varepsilon(p)| \geqslant MinPts \end{cases}$$

OPTICS - Алгоритм

```
OPTICS(DB, eps, MinPts)
   для каждой точки p из DB
      p.достижимое_расстояние=не_определено
   для каждой необработанной точки p из DB
      N=получитьСоседей (p, eps)
      пометить p как обработанную
      поместить p в упорядоченный список
      если (основное_расстояние (p, eps, Minpts) != не_определено)
         Seeds=пустая приоритетная очередь
         обновить(N, p, Seeds, eps, Minpts)
         для каждой следующей q из Seeds
            N'=получитьСоседей(q, eps)
            пометить q как обработанную
            поместить q в упорядоченный список
            если (основное_расстояние(q, eps, Minpts) != не_определено)
               обновить(N', q, Seeds, eps, Minpts)
```

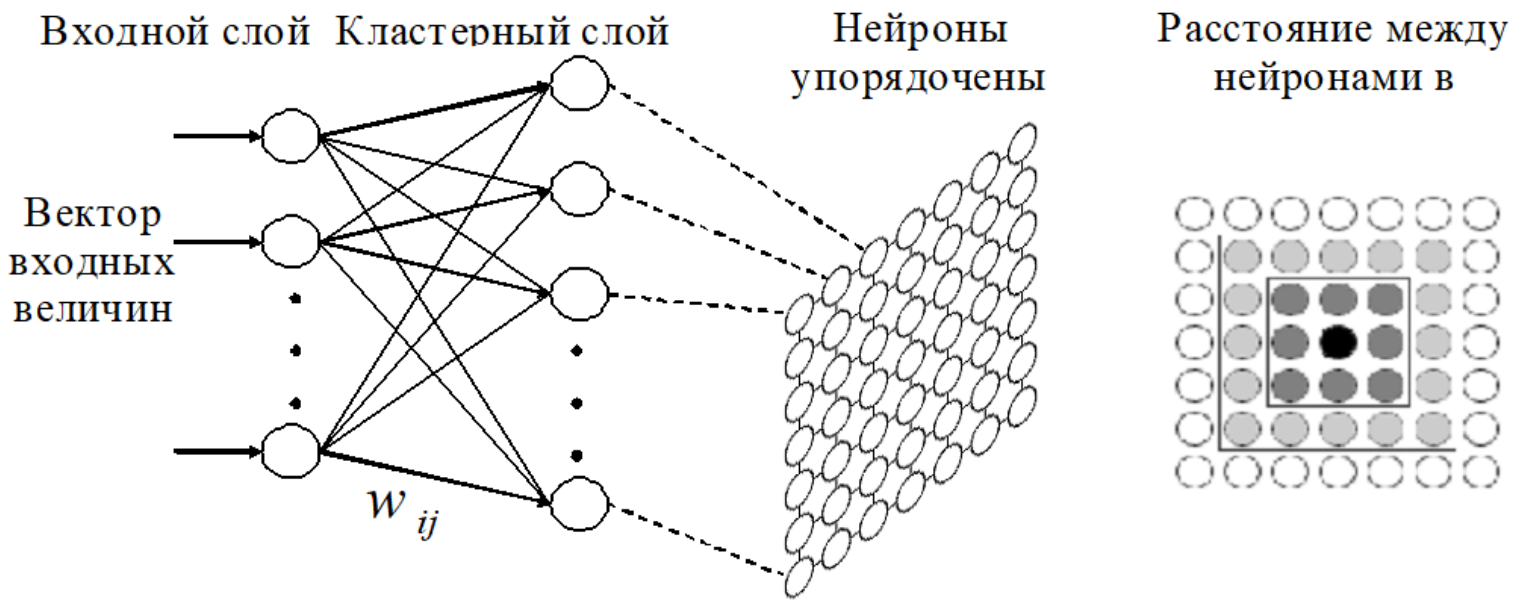
```
обновить (N, p, Seeds, eps, Minpts)
   coredist=основное_расстояние(p, eps, MinPts)
   для каждого o в N
      если (o не обработана)
         новое_дост_расст=max(coredist, dist(p,o))
         если (o.достижимое_расстояние == не_определено) // точка o не в Seeds
             o.достижимое_расстояние=новое_дост_расст
             Seeds.вставить(o, новое_дост_расст)
         иначе                  // точка o в Seeds, проверить на улучшение
             если (новое_дост_раст < o.достижимое_расстояние)
                o.достижимое_расстояние=новое_дост_раст
                Seeds.передвинуть_вверх(o, новое_дост_раст)
```

OPTICS – результат разбиения



Самоорганизующиеся карты Кохонена (SOM)

Состоит из входного слоя и одного конкурирующего кластерного слоя, нейроны которого вычисляют расстояние между входным вектором и вектором весовых коэффициентов. Победителем является тот нейрон, расстояние до которого минимально. Важная особенность алгоритма SOM (Self Organizing Maps) заключается в том, что все нейроны кластерного слоя (ядра классов) упорядочены в некоторую структуру, что позволяет ввести меру взаимодействия между нейронами кластерного слоя не в пространстве входных признаков, а на используемой карте. Величина этого взаимодействия определяется расстоянием между нейронами на карте.



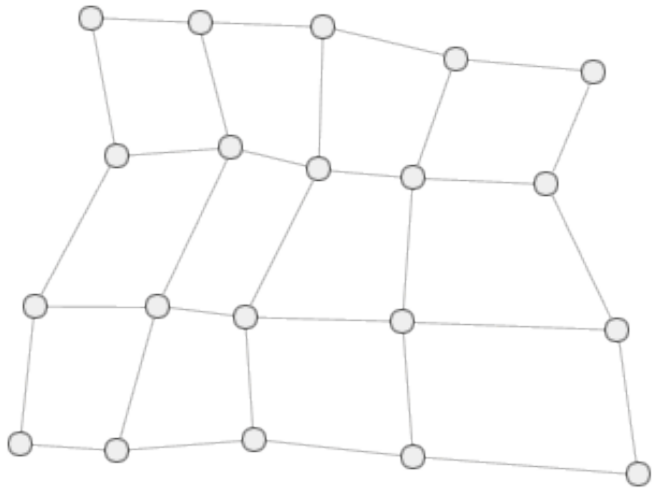
SOM - Обучение сети

- инициализацию сети (количество нейронов в сети и условие взаимодействия между нейронами кластерного слоя)
- инициализация весовых коэффициентов
  - Инициализация случайными значениями
  - Инициализация примерами
  - Линейная инициализация.
- Обучение (корректировка весов). Подстройке подлежат веса, как нейрона-победителя, так и его соседей по сетке (одно или двумерной), но в меньшей степени.

$$w_i(t+1) = w_i(t) + h_{ci}(t) * [x(t) - w(t)]$$

$$h(t) = h(\| a_c - a_i \|, t) * f(t)$$

$$h(d,t) = e^{-\left(\frac{d}{b(t)}\right)^2}$$



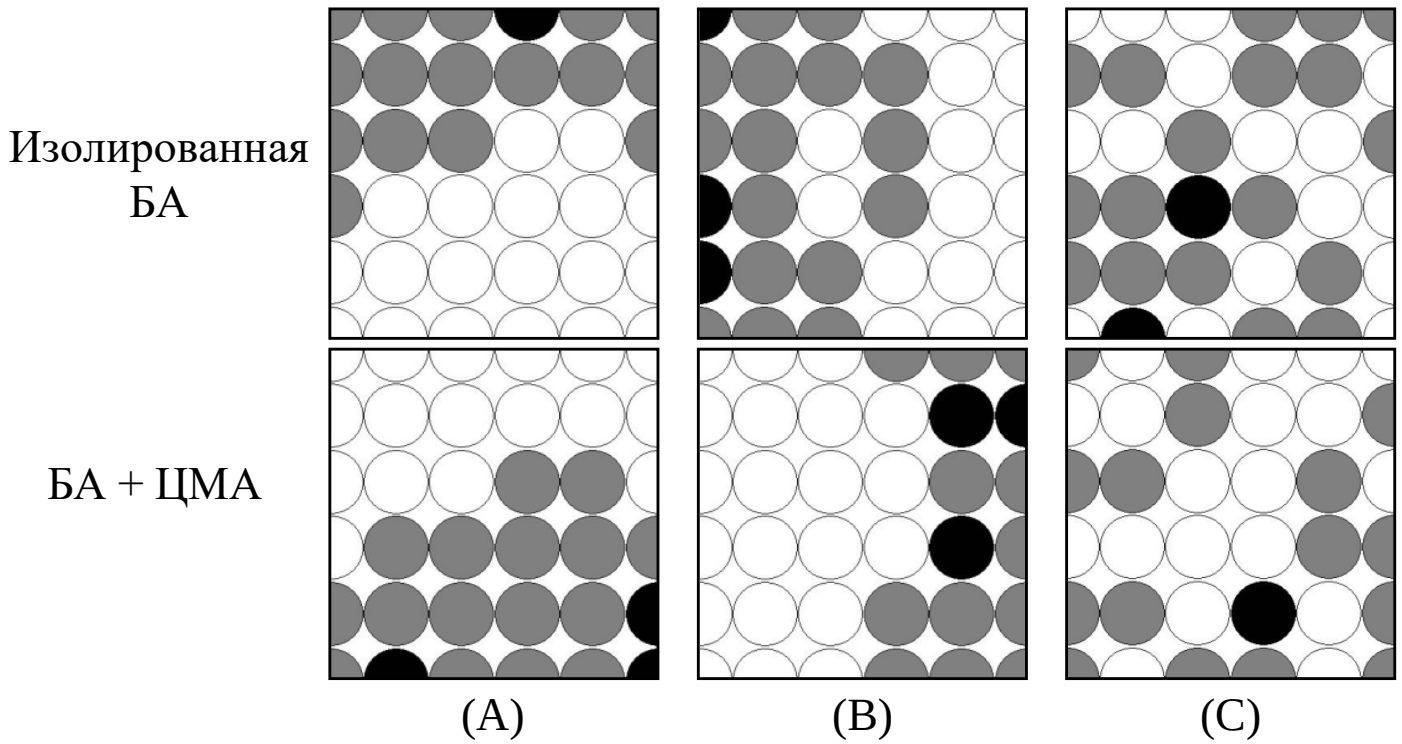
SOM - Раскраска обученной карты

Добавление дополнительных связей в пространстве отображения результата позволяют использовать сеть Кохонена, как для решения задач распознавания и определения близости кластеров в данных, так и для визуализации многомерного пространства

- в соответствии с расстояниями между векторами весовых коэффициентов нейронов кластерного слоя;
- в зависимости от значений некоторой компоненты вектора обучающей выборки;
- в зависимости от числа примеров отнесенных к соответствующему нейрону карты.

В совокупности полученные раскраски образуют атлас, отображающий присутствующие в данных кластеры близких объектов, зависимости от значений отдельных или наборов компонент, относительное расположение их различных значений, что в значительной степени может помочь пользователю разобраться в структуре обрабатываемых данных.

Пример раскраски обученной SOM



На рисунке представлены результата обучения и раскраски SOM-сетей, как для всей совокупности измеренных ДТ-МРТ показателей всех областей мозга (A), так и для отдельных анатомических структур на примере наиболее характерных случаев, когда классы разделимы (B) и не разделимы (C). Белый цвет нейрона говорит о том, что он не был активирован ни одним из примеров выборки указанного класса. Градации серого характеризуют число примеров выборки, принадлежащих соответствующему нейрону.

Спасибо за внимание