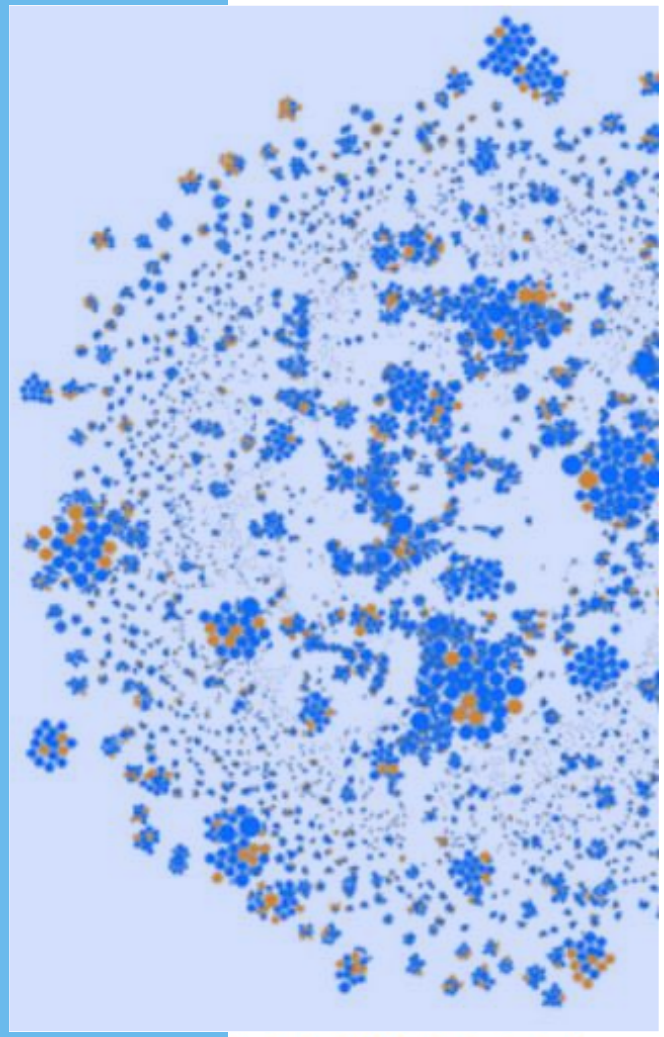
СНИЖЕНИЕ РАЗМЕРНОСТИ ДАННЫХ И КЛАСТЕРИЗАЦИЯ

Снижение размерности данных

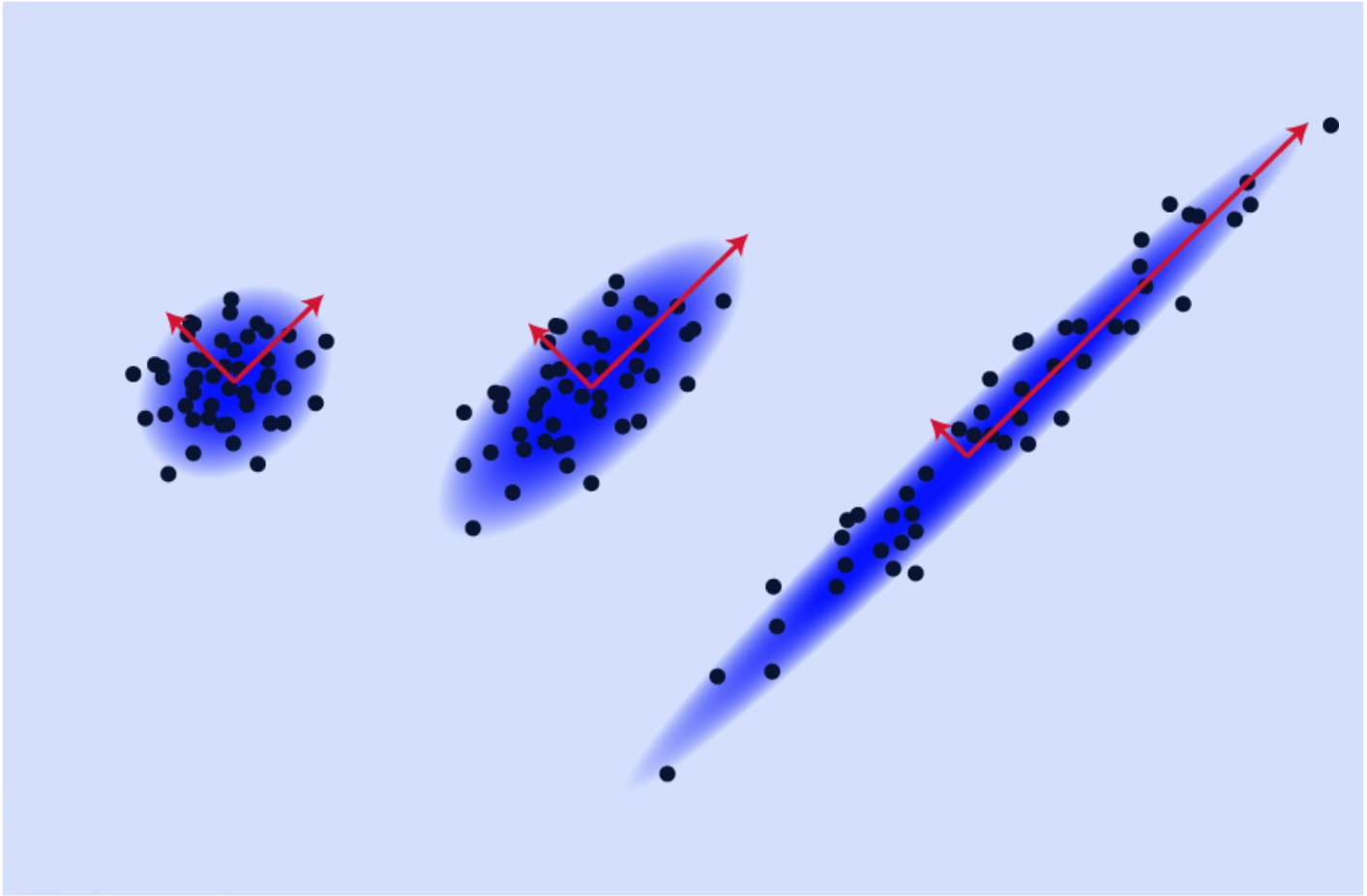


Снижение размерности используется для:

- Сокращение вычислительных затрат при обработке данных;
- Борьба с переобучением. Чем меньше количество признаков, тем меньше требуется объектов для уверенного восстановления скрытых зависимостей в данных и тем больше качество восстановления подобных зависимостей;
- Сжатие данных для более эффективного хранения информации. В этом случае помимо преобразования $X \rightarrow T$ требуется иметь возможность осуществлять также обратное преобразование $T \rightarrow X$;
- Визуализация данных. Проектирование выборки на двух-/трехмерное пространство позволяет графически представить выборку;
- Извлечение новых признаков. Новые признаки, полученные в результате преобразования $X \rightarrow T$, могут оказывать значимый вклад при последующем решении задачи.

Метод главных компонент (PCA)

Идея метода заключается в поиске в исходном пространстве гиперплоскости заданной размерности с последующим проектированием выборки на данную гиперплоскость. При этом выбирается та гиперплоскость, ошибка проектирования данных на которую является минимальной в смысле суммы квадратов отклонений.

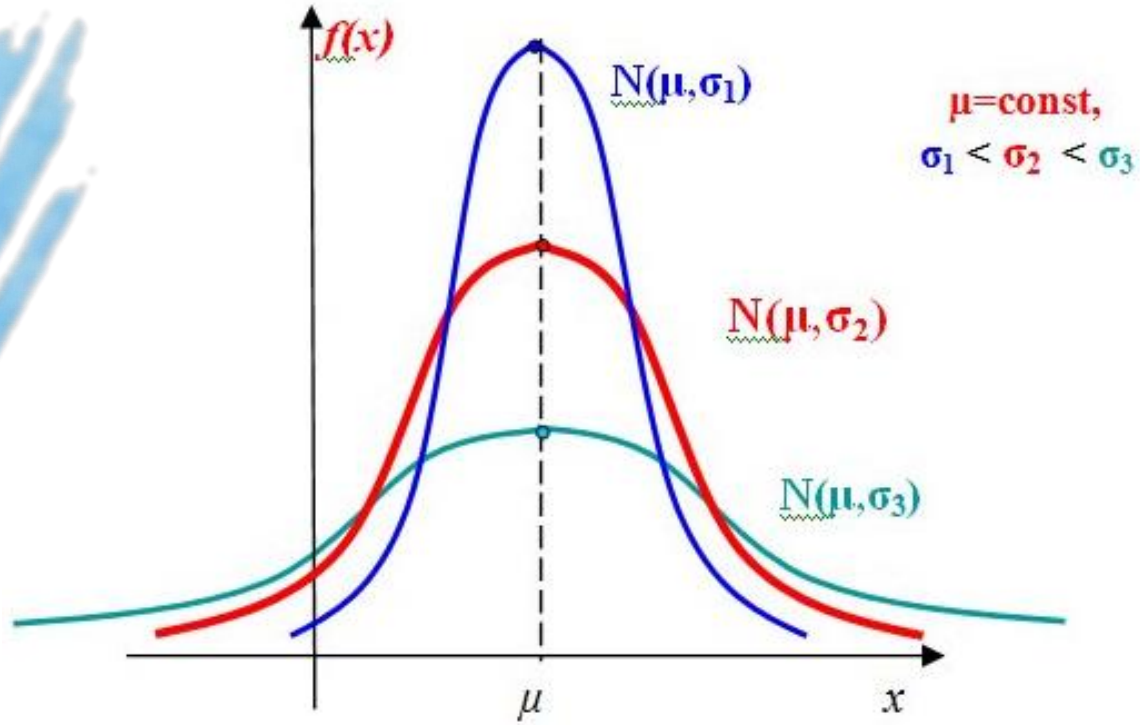


МЕТОД ГЛАВНЫХ КОМПОНЕНТ

Дисперсия

Дисперсия - это мера разброса значений случайной величины относительно ее математического ожидания. Дисперсия показывает, насколько в среднем значения сосредоточены, сгруппированы около математического ожидания.

$$D(X) = E[(X - E(X))^2]$$



Если дисперсия маленькая - значения сравнительно близки друг к другу, если большая - далеки друг от друга.

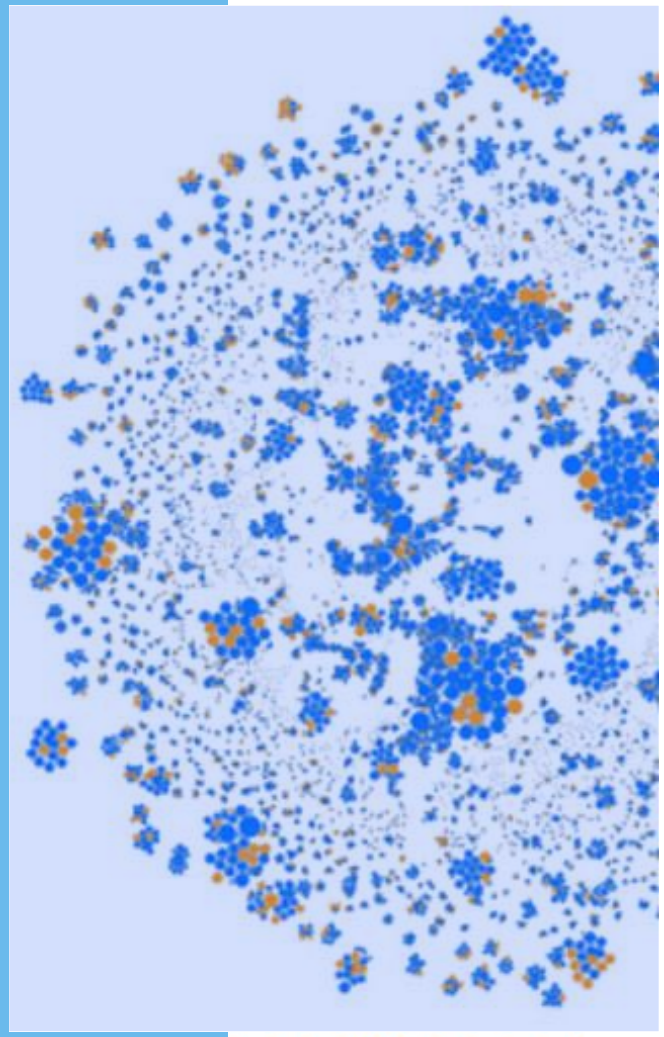
МЕТОД ГЛАВНЫХ КОМПОНЕНТ

Дисперсия

Дисперсия - это мера разброса значений случайной величины относительно ее математического ожидания. Дисперсия показывает, насколько в среднем значения сосредоточены, сгруппированы около математического ожидания.

$$D(X) = E[(X - E(X))^2]$$

Метод главных компонент (PCA)



Ковариация и ковариационная матрица

- Пусть X и Y - две случайные величины, определенные на одном вероятностном пространстве. Тогда их ковариация определяется следующим образом:

  $cov(X,Y\ )=E[(X-E(X))(Y-E(Y))]$

- Пусть $X\_1, X\_2, \ldots, X\_n$ и $Y\_1, Y\_2, \ldots, Y\_n$ - выборки случайных величин X и Y. Тогда их ковариация вычисляется следующим образом:

  $$cov(X,Y\ ) = \frac{1}{n}\sum_{i=1}^{n} X_i Y_i\ - (\frac{1}{n}\sum_{i=1}^{n} X_i)(\frac{1}{n}\sum_{i=1}^{n} Y_i)$$

- Ковариационная матрица случайного вектора $(\xi_1, \xi_2, \ldots, \xi_N)$:

$$cov\ (\xi_1, \xi_2, \ldots, \xi_N) = \begin{pmatrix} cov(\xi_1, \xi_1) & \cdots & cov(\xi_1, \xi_N) \\ \vdots & \ddots & \vdots \\ cov(\xi_N, \xi_1) & \cdots & cov(\xi_N, \xi_N) \end{pmatrix}$$

Метод главных компонент (PCA)

СХЕМА АЛГОРИТМА

Есть датасет:

X_1	X_2	…	X_N
…	…	…	…

Задача: получить датасет меньшей размерности с минимальной потерей информации:

X'_1	X'_2	…	X'_M
…	…	…	…

$$M < N$$

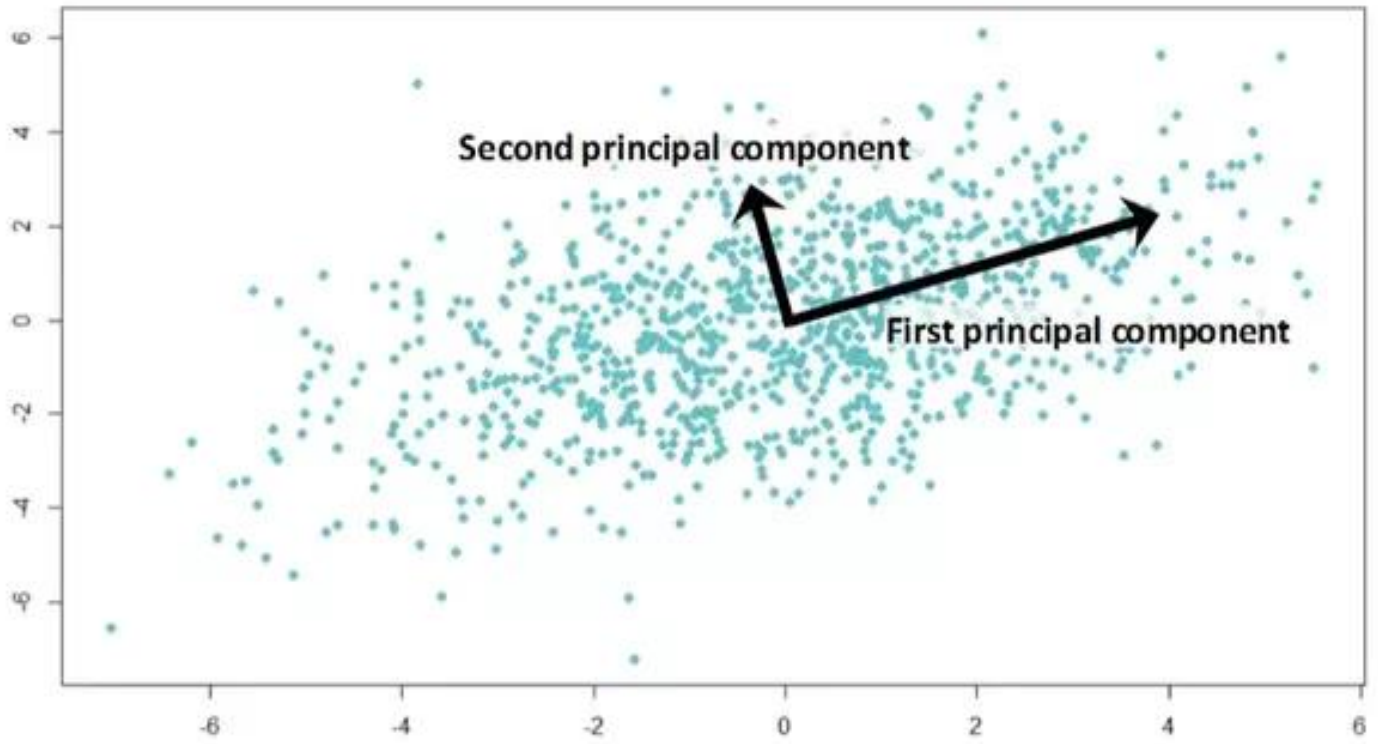
Алгоритм:

- находим ковариационную матрицу $cov\,(X_1, X_2, \dots, X_N)$ признаков как случайных величин;
- находим собственные значения полученной матрицы;
- находим собственный вектор соответствующий максимальному собственному значению.

Направление максимальной дисперсии у проекции всегда совпадает с собственным вектором, имеющим максимальное собственное значение, равное величине этой дисперсии.

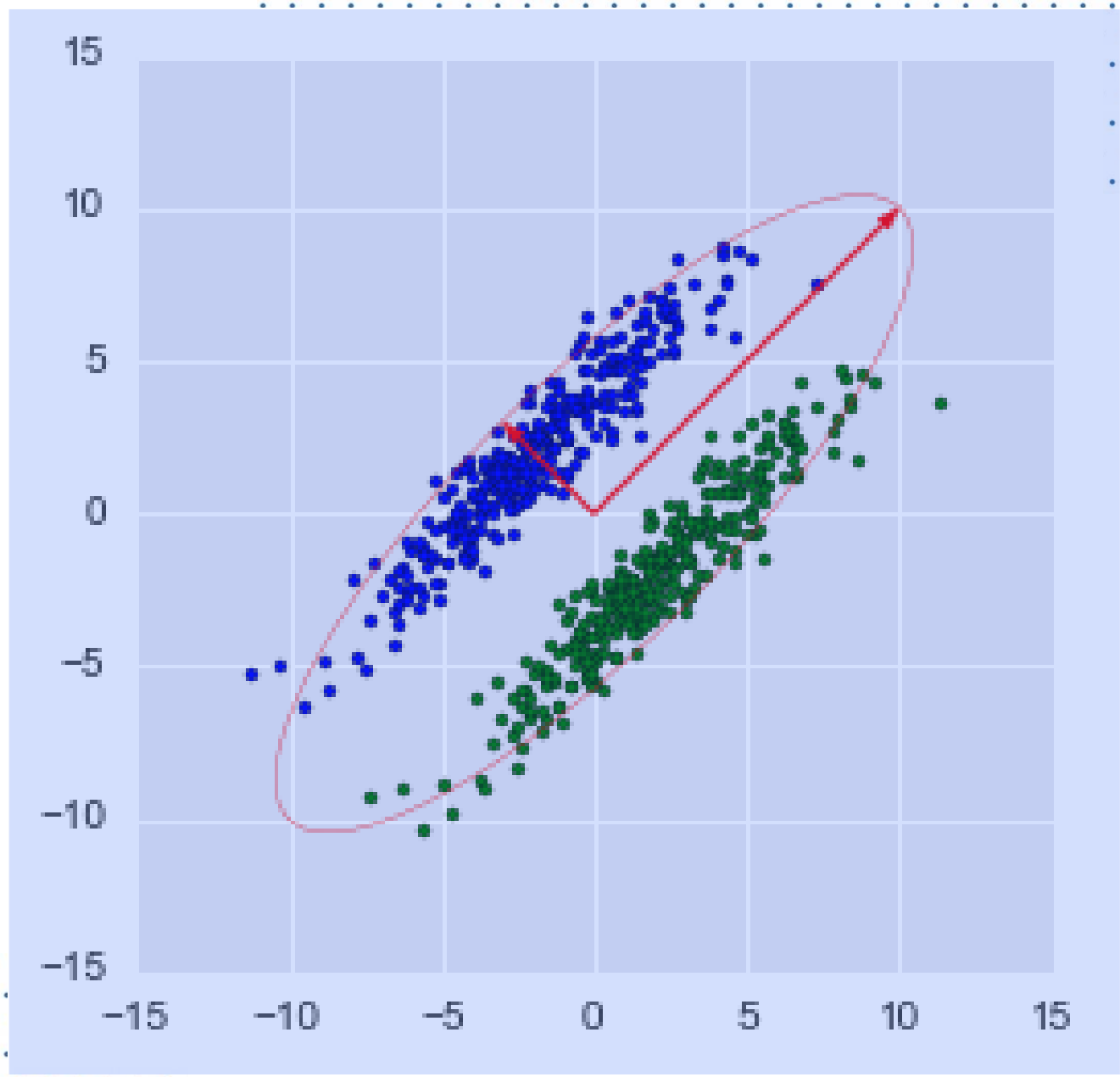
Метод главных Компонент (PCA)

Работает!



Метод главных Компонент (PCA)

Не работает(



t-SNE

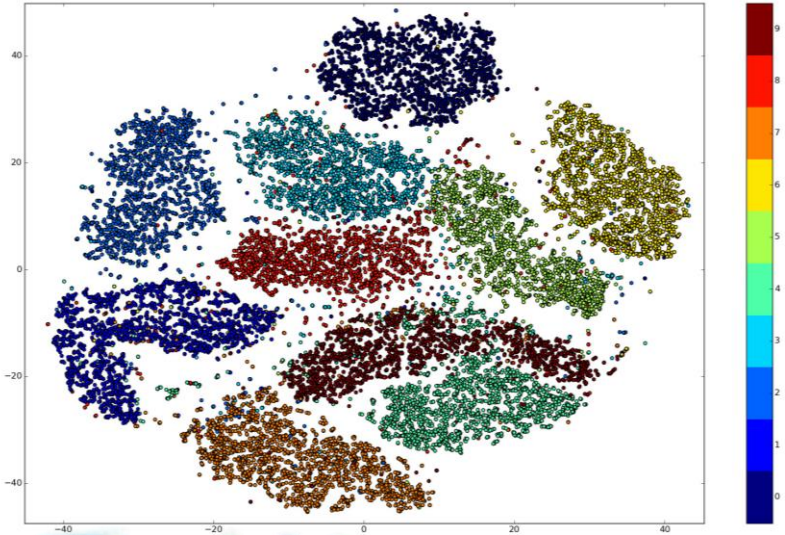
- Дано $x_1, x_2, \dots, x_n \in \mathbb{R}^N$. Нужно найти $y_1, y_2, \dots, y_n \in \mathbb{R}^M, M < N$, причем если $\|x_i - x_j\| = \alpha\|x_i - x_k\|$, то $\|y_i - y_j\| = \alpha\|y_i - y_k\|$.

- Найдем условные вероятности $p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$ и $q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}$

- Находить вектора $y_1, y_2, \dots, y_n$ будем методом градиентного спуска минимизируя следующую функцию потерь:

$$Loss = \sum_{i,j} p_{j|i} \, log(\frac{p_{j|i}}{q_{j|i}}) -$$

расстояние Кульбака-Лейблера между распределениями.

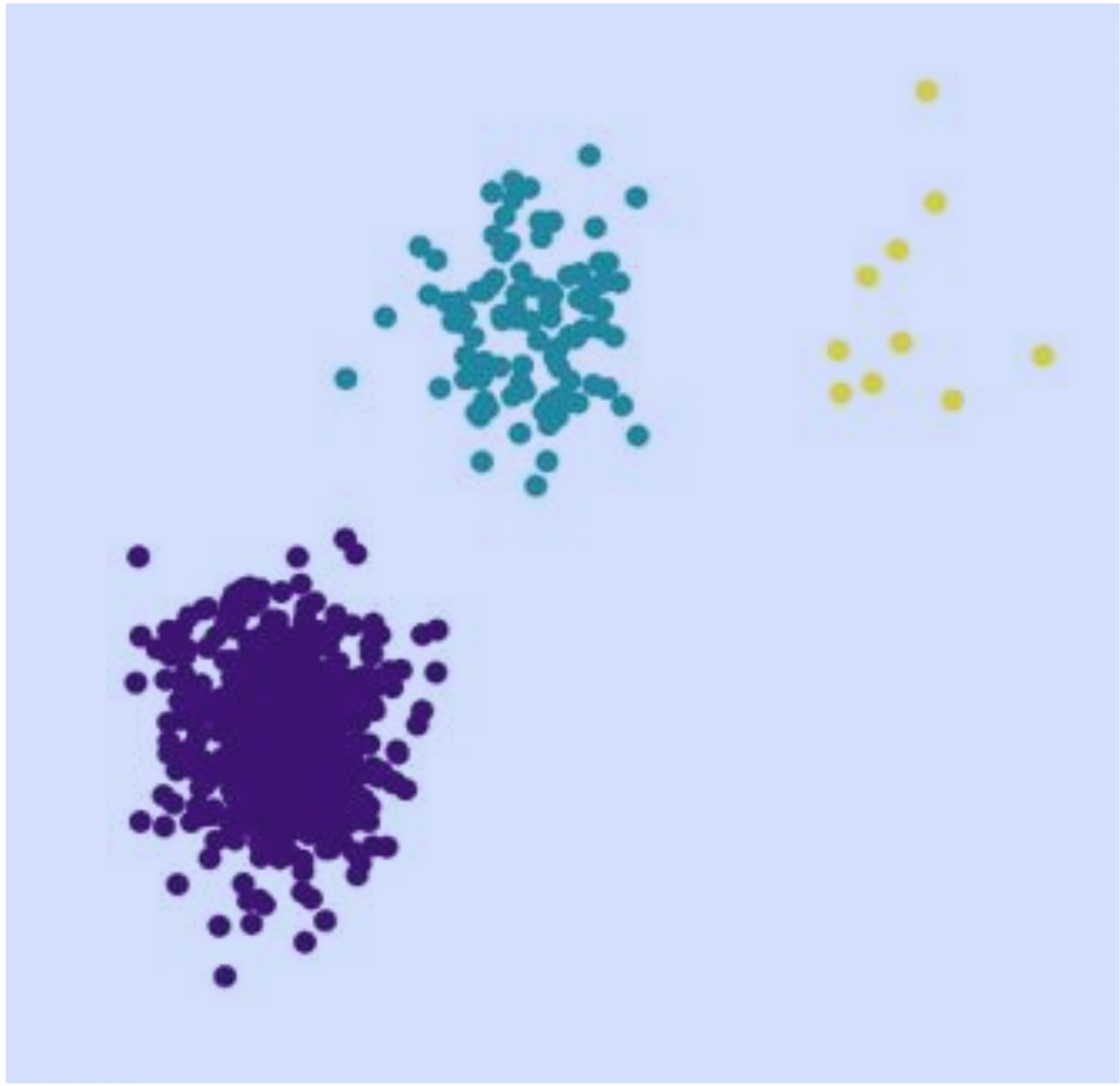


- Формула градиента: $\frac{\delta Loss}{\delta y_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - p_{i|j})(y_i - y_j)$

Кластеризация

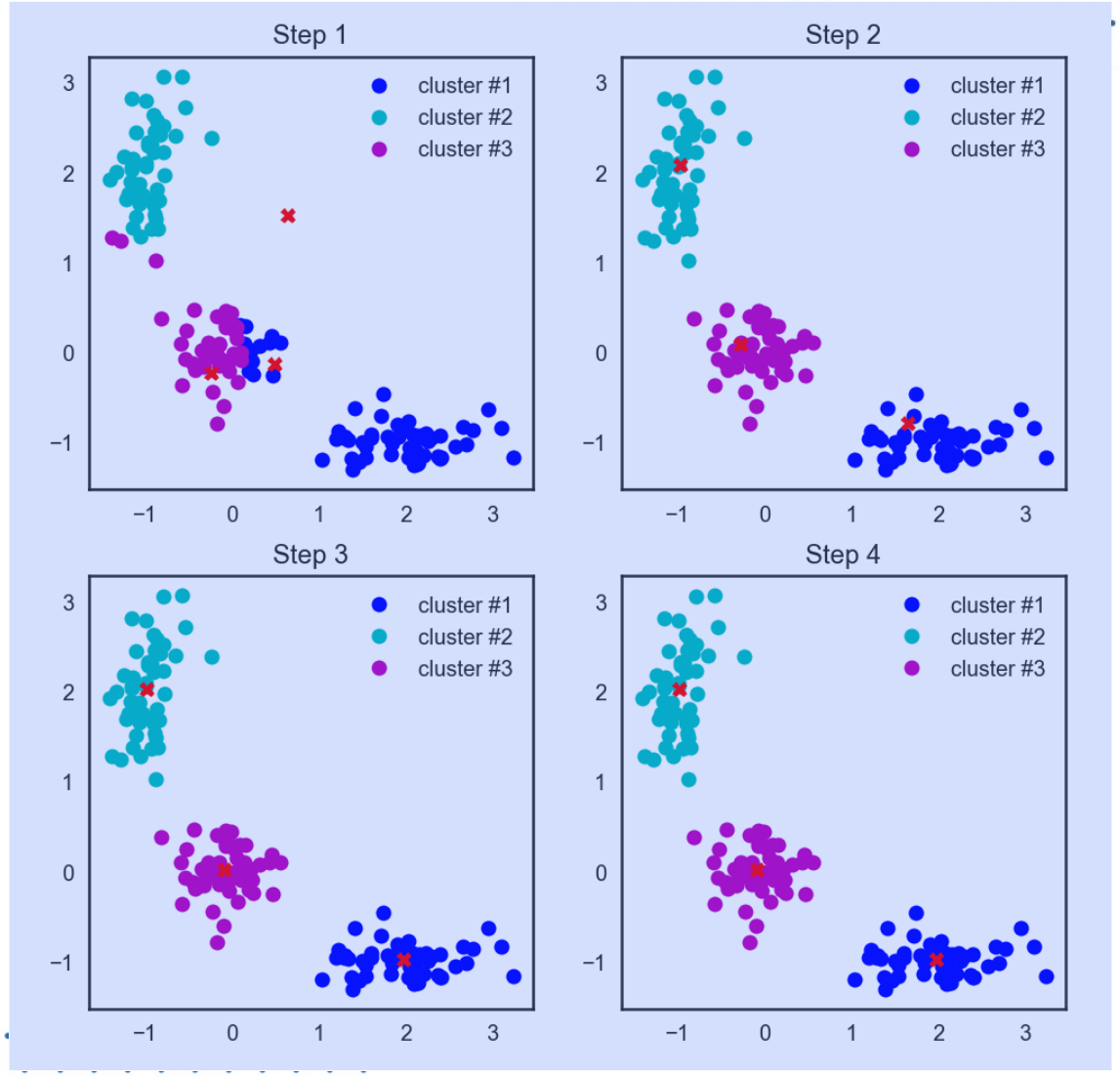
Кластеризация

(англ. 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. Задача кластеризации относится к классу задач обучения без учителя.



Кластеризация. k-Means

- Выбрать количество кластеров k, которое нам кажется оптимальным для наших данных.
- Высыпать случайным образом в пространство наших данных k точек (центроидов).
- Для каждой точки нашего набора данных посчитать, к какому центроиду она ближе.
- Переместить каждый центроид в центр выборки, которую мы отнесли к этому центроиду.
- Повторять последние два шага фиксированное число раз, либо до тех пор пока центроиды не "сойдутся" (обычно это значит, что их смещение относительно предыдущего положения не превышает какого-то заранее заданного небольшого значения).



Кластеризация. k-Means

ПРОБЛЕМЫ АЛГОРИТМА K-MEANS:

- **необходимо заранее знать количество кластеров.** Мной было предложено метод определения количества кластеров, который основывался на нахождении кластеров, распределенных по некоему закону (в моем случае все сводилось к нормальному закону). После этого выполнялся классический алгоритм k-means, который давал более точные результаты.
- **алгоритм очень чувствителен к выбору начальных центров кластеров.** Классический вариант подразумевает случайный выбор кластеров, что очень часто являлось источником погрешности. Как вариант решения, необходимо проводить исследования объекта для более точного определения центров начальных кластеров. В моем случае на начальном этапе предлагается принимать в качестве центов самые отдаленные точки кластеров.
- не справляется с задачей, когда объект принадлежит к разным кластерам в равной степени или не принадлежит ни одному.

Кластеризация

c-Means (Fuzzy Classifier Means, Fuzzy C-Means)

Метод **нечеткой кластеризации c-Means** можно рассматривать как усовершенствованный метод k-means, при котором для каждого элемента из рассматриваемого множества рассчитывается степень его принадлежности каждому из кластеров.

https://code.google.com/archive/p/peach/

Алгоритм c-Means:

- Задать случайным образом k центров кластеров $c_j, j = 1,2, \dots, k$.
- Рассчитать матрицу принадлежности к кластерам

  $\Delta = (\delta_{i,j})$, где $\delta_{i,j} = \frac{N(d(x_i,c_j))}{\sum_j N(d(x_i,c_j))}$.
- Переместить центры кластеров $c_j = \frac{\sum_i \delta_{i,j} x_i}{\sum_i \delta_{i,j}}$.
- Рассчитать функцию потерь:

  loss = $\sum_{j=1}^{k} \sum_{i=1}^{N} d(x_i, c_j)^2 \delta_{i,j}$.
- Если значение функции потерь уменьшается, повторить цикл с пункта 2.

Кластеризация

Hierarchical (agglomerative) clustering

- Присваиваем каждой точке свой кластер
- Сортируем попарные расстояния между центрами кластеров по возрастанию
- Берём пару ближайших кластеров, склеиваем их в один и пересчитываем центр кластера
- Повторяем п. 2 и 3 до тех пор, пока все данные не склеятся в один кластер

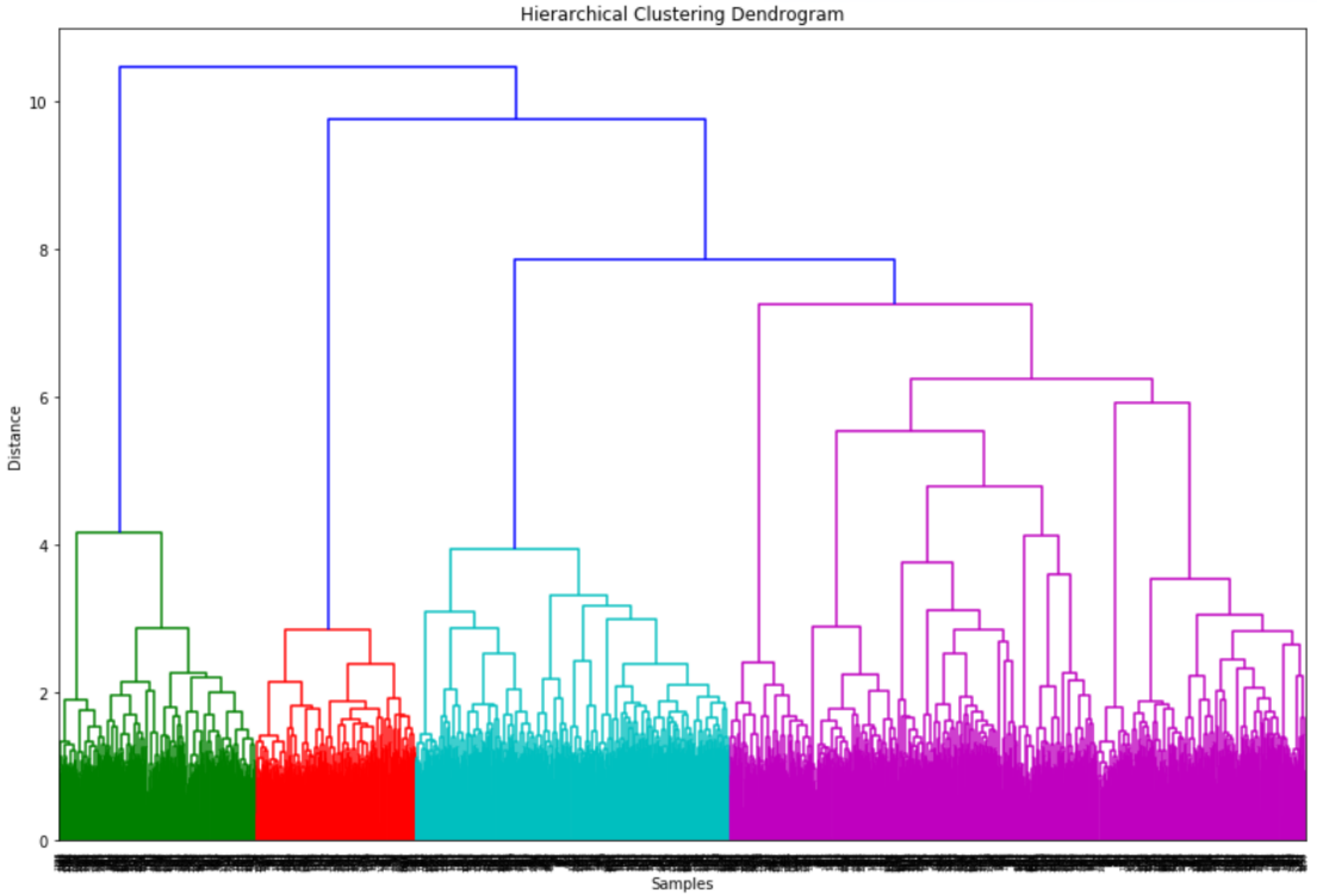
Single linkage – минимум попарных расстояний между точками из двух кластеров: $d(C_i, C_j) = min_{x_i \in C_i, x_j \in C_j} \|x_i - x_j\|$

Complete linkage – максимум попарных расстояний между точками из двух кластеров: $d(C_i, C_j) = max_{x_i \in C_i, x_j \in C_j} \|x_i - x_j\|$

Average linkage – среднее попарных расстояний между точками из двух кластеров: $d(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x_i \in C_i} \sum_{x_j \in C_j} \|x_i - x_j\|$

Кластеризация.

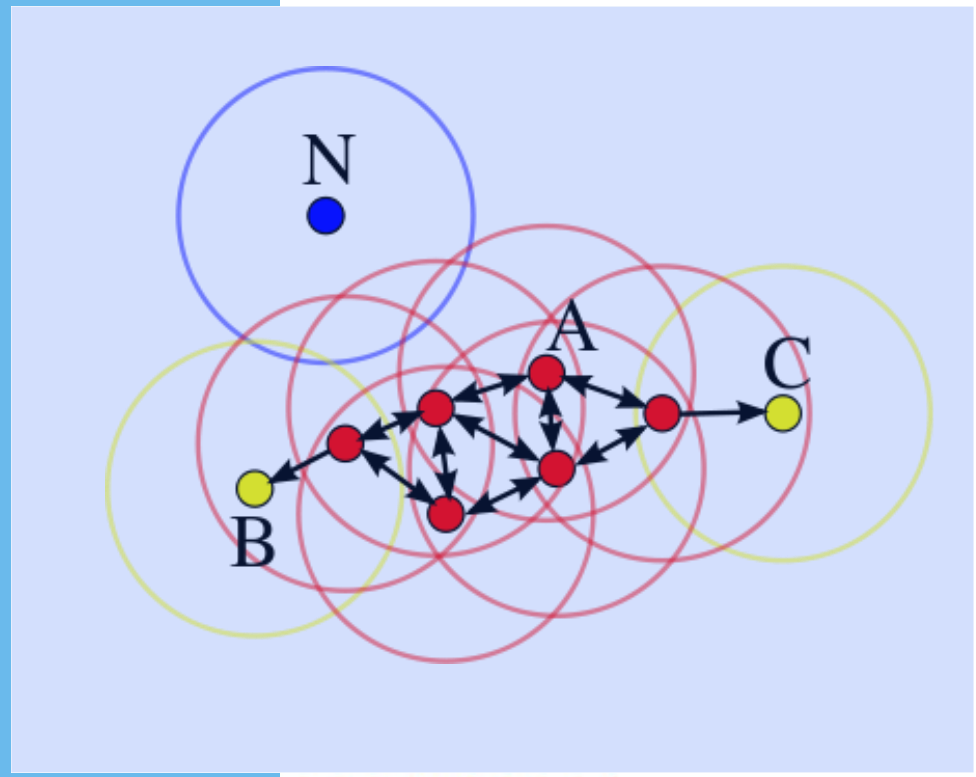
Hierarchical (agglomerative) clustering



Кластеризация

DBSCAN - Density-based spatial clustering of applications with noise

Для выполнения кластеризации DBSCAN точки делятся на основные точки, достижимые по плотности точки и выпадающие следующим образом:



- Точка p является основной точкой, если по крайней мере minPts точек находятся на расстоянии не превосходящем ε. Эти точки достижимы прямо из p.
- Точка q прямо достижима из точки p, если она находится на расстоянии не больше ε от p и p является основной точкой.
- Точка q достижима из p, если имеется путь p_1 p_2…p_n, где ⟦p= p⟧_1, q=p_n и каждая точка p_(i+1) прямо достижима из p_i.
- Все точки, не достижимые из основных точек, считаются выбросами.

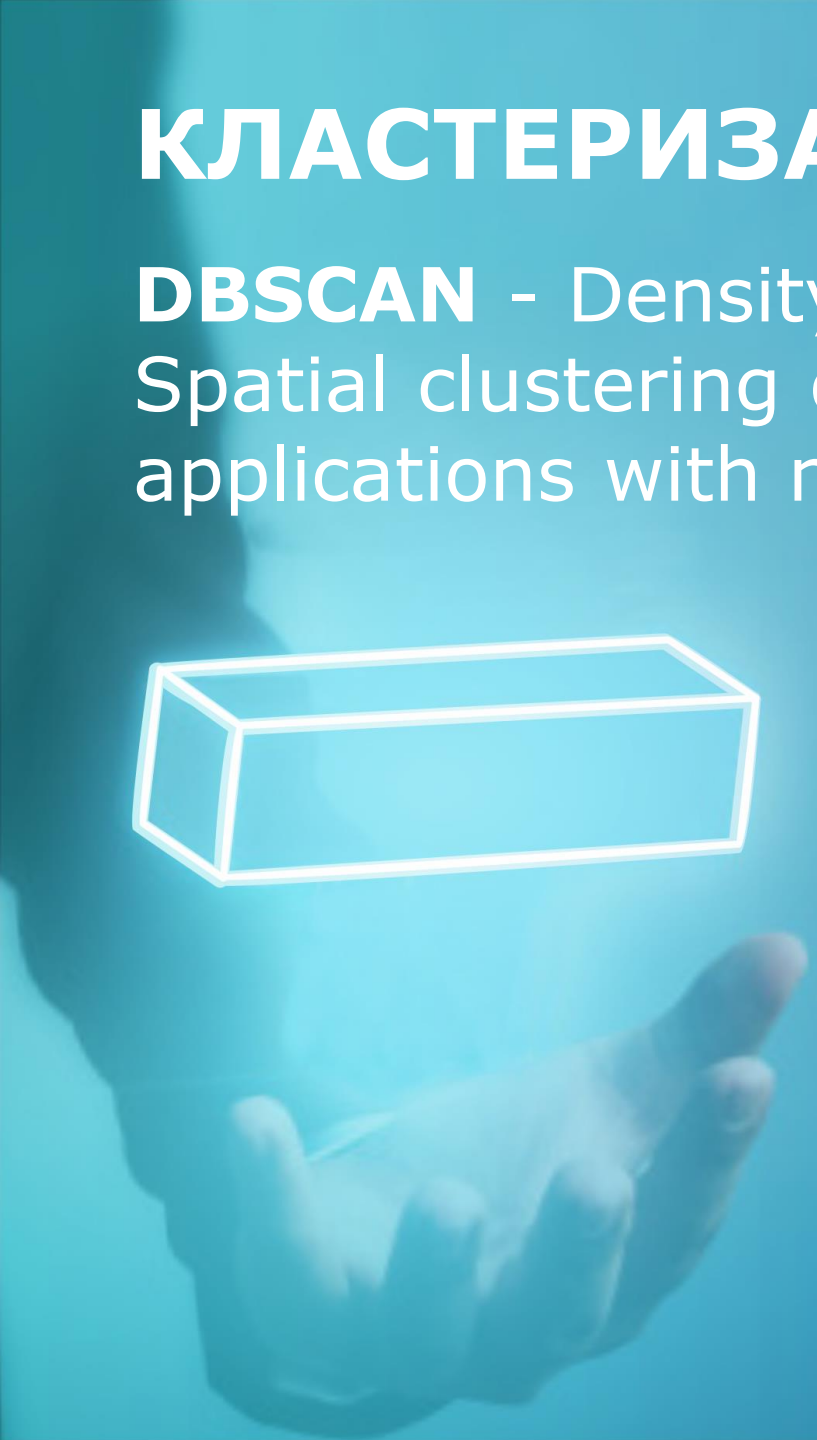
КЛАСТЕРИЗАЦИЯ

DBSCAN - Density-based spatial clustering of applications with noise

- DBSCAN не требует спецификации числа кластеров в данных априори в отличие от метода k-means.
- DBSCAN может найти кластеры произвольной формы. Он может найти даже кластеры полностью окружённые (но не связанные с) другими кластерами. Благодаря параметру MinPts уменьшается так называемый эффект одной связи (связь различных кластеров тонкой линией точек).
- DBSCAN имеет понятие шума и устойчив к выбросам.
- DBSCAN требует лишь двух параметров и большей частью нечувствителен к порядку точек в базе данных.
- DBSCAN разработан для применения с базами данных, которые позволяют ускорить запросы в диапазоне значений, например, с помощью

КЛАСТЕРИЗАЦИЯ

DBSCAN - Density-based Spatial clustering of applications with noise



- DBSCAN 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 выбранной функции расстояний между точкам.
- DBSCAN не может хорошо кластеризовать наборы данных с большой разницей в плотности, поскольку не удается выбрать приемлемую для всех кластеров комбинацию параметров.

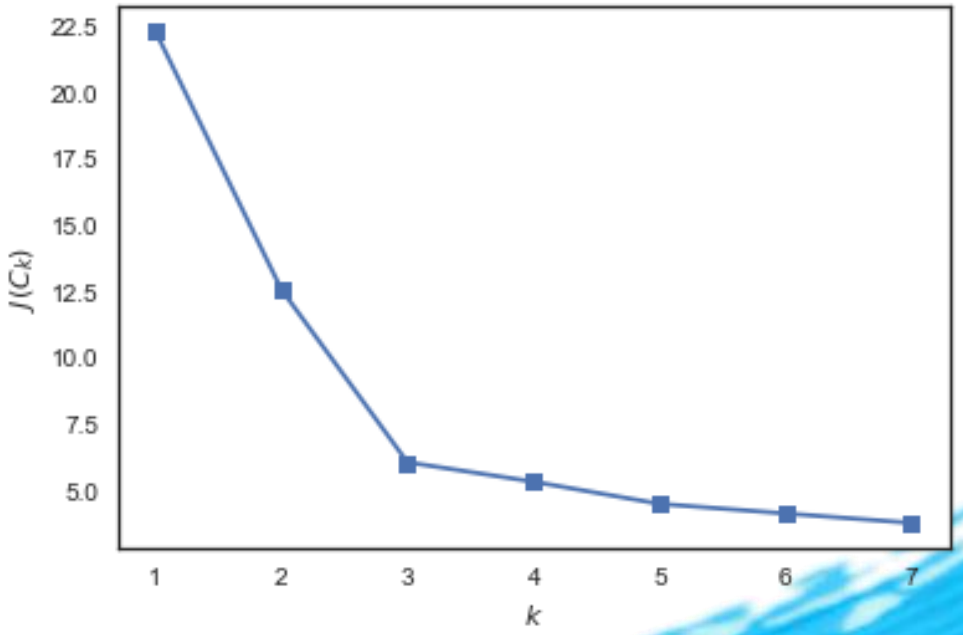
КЛАСТЕРИЗАЦИЯ

Выбор числа кластеров

Задача кластеризации → задача минимизации $J(C) = \sum_{k=1}^{N} \sum_{i \in c_k} \|x_i - c_k\|^2 \rightarrow min$,
где C – множество точек, разбитых на N кластеров $C_1, C_2, \dots, C_N$ с центрами $c_1, c_2, \dots, c_N$.

Для решения этого вопроса (выбора числа кластеров) часто пользуются такой эвристикой: выбирают то число кластеров, начиная с которого описанный функционал $J(C)$ падает "уже не так быстро".

$$D(N) = \frac{J_N(C) - J_{N+1}(C)}{J_{N-1}(C) - J_N(C)} \rightarrow min$$



КЛАСТЕРИЗАЦИЯ

Метрики

Adjusted Rand Index (ARI)

$X \backslash Y$	Y_1	Y_2	...	Y_s	Sums
X_1	n_{11}	n_{12}	...	n_{1s}	a_1
X_2	n_{21}	n_{22}	...	n_{2s}	a_2
⋮	⋮	⋮	⋱	⋮	⋮
X_r	n_{r1}	n_{r2}	...	n_{rs}	a_r
Sums	b_1	b_2	...	b_s	

$$\overbrace{ARI}^{\text{Adjusted Index}} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}$$

Adjusted Mutual Information (AMI)

$$MI(X,Y) = \sum_{i=1}^{r}\sum_{j=1}^{s} P(i,j)log(\frac{P(i,j)}{P(i)P'(j)}),$$

где $P(i) = \frac{|X_i|}{|C|}$, $P'(j) = \frac{|Y_i|}{|C|}$, $P(i,j) = \frac{|X_i \cap Y_j|}{|C|}$.

$$AMI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{\max\{H(U),H(V)\} - E\{MI(U,V)\}}$$

http://neerc.ifmo.ru/wiki/index.php?title=%D0%9E%D1%86%D0%B5%D0%BD%D0%BA%D0%B0_%D0%BA%D0%B0%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%B0_%D0%B2_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8#Adjusted_Rand_Index

Кластеризация

Метрики

Пусть a - среднее расстояние от данного объекта до объектов из того же кластера.

Пусть b - среднее расстояние от данного объекта до объектов хиз ближайшего кластера (отличного от того, в котором лежит сам объект).

Силуэт объекта – это величина $s = \frac{b-a}{\max\{a,b\}}$

Силуэтом выборки называется средняя величина силуэта объектов данной выборки.

Данные метрики оценивают качество структуры кластеров опираясь только непосредственно на нее, не используя внешней информации:

- davies–Bouldin Index
- score function
- gamma Index
- COP Index

Code

from sklearn import metrics