



ПРОГРАММИРОВАНИЕ CUDA  
C/C++, АНАЛИЗ ИЗОБРАЖЕНИЙ  
И DEEP LEARNING

Лекция №5

Спасёнов Алексей

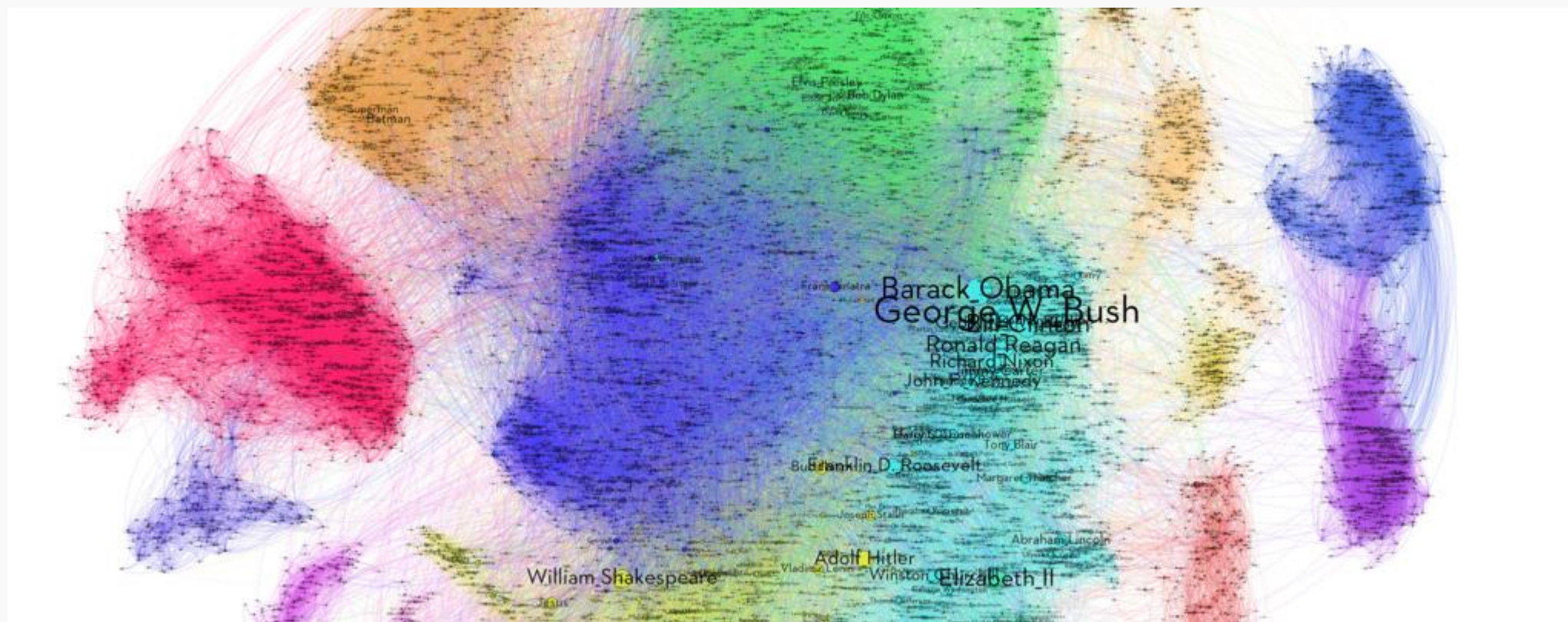
## Кластеризация (обучение без учителя)

1. Задача кластеризации
2. Алгоритм K-Means
3. EM-алгоритм
4. Иерархическая кластеризация
5. Алгоритм DBSCAN
6. Оценка качества

# Примеры задачи кластеризации (1/3)



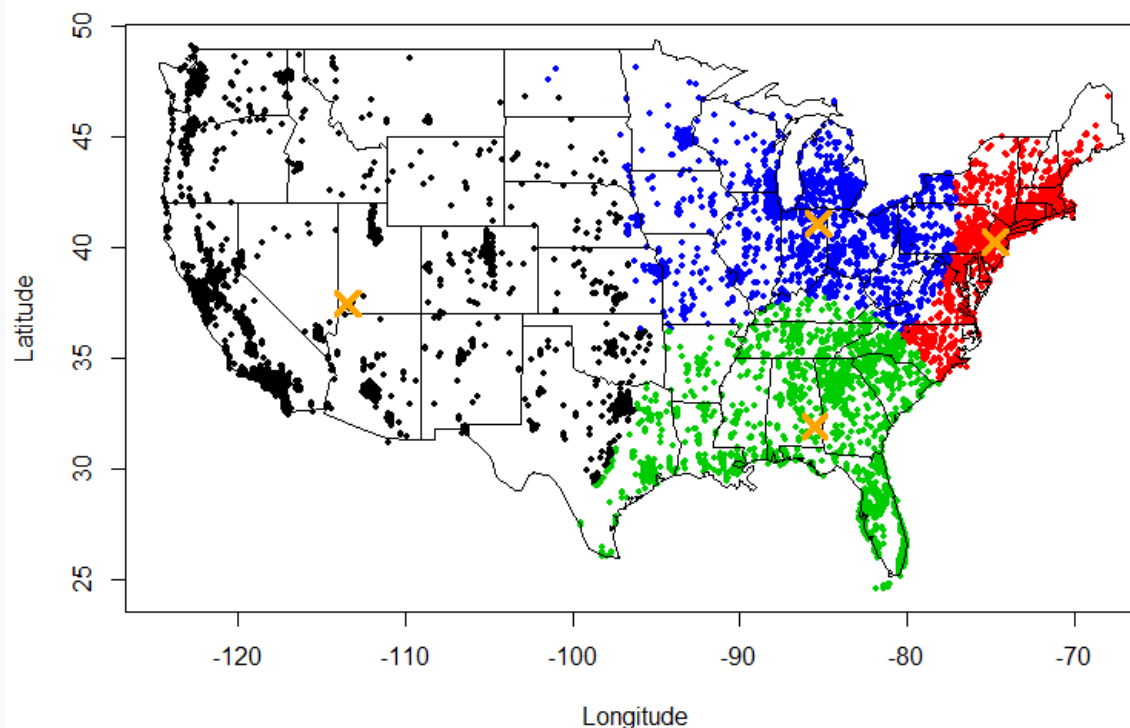
## Анализ текста



# Примеры задачи кластеризации (2/3)



## Анализ геоданных



# Задача кластеризации (1/3)



## Дано:

$x_1, \dots, x_l$  — объекты обучающей выборки  $X$

$\rho$  — функция расстояния между объектами

## Задача:

Поиск меток  $y_1, \dots, y_l$ , таких, чтобы объекты с одинаковыми метками были близки по  $\rho$ , а с разными метками существенно различались



# Задача кластеризации (2/3)



## Цели:

- 1) Упрощение обработки данных за счёт разбиения исходного набора данных на схожие подгруппы
- 2) Уменьшение объёма хранимых данных
- 3) Поиск объектов, не относящихся ни к одному из исследуемых классов
- 4) Построение иерархии множества объектов

# Задача кластеризации (3/3)

---



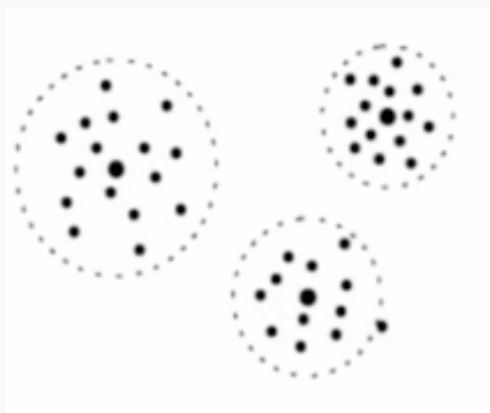
## Проблемы:

- 1) Выбор критерия качества кластеризации
- 2) Выбор метода кластеризации
- 3) Выбор числа кластеров, на которые требуется разбить исходное множество объектов
- 4) Выбор функции расстояния  $\rho$  (метрики)

# Конфигурации кластеров (1/4)



## Кластеры с центром



Расстояние между объектами внутри кластера меньше межкластерного



# Конфигурации кластеров (2/4)



**Ленточные кластеры**



**Кластеры с перемычками**



# Конфигурации кластеров (3/4)



**Присутствие фона**



**Кластеры с перекрытием**



# Конфигурации кластеров (4/4)



**Внутренние особенности**



**Отсутствие кластеров**

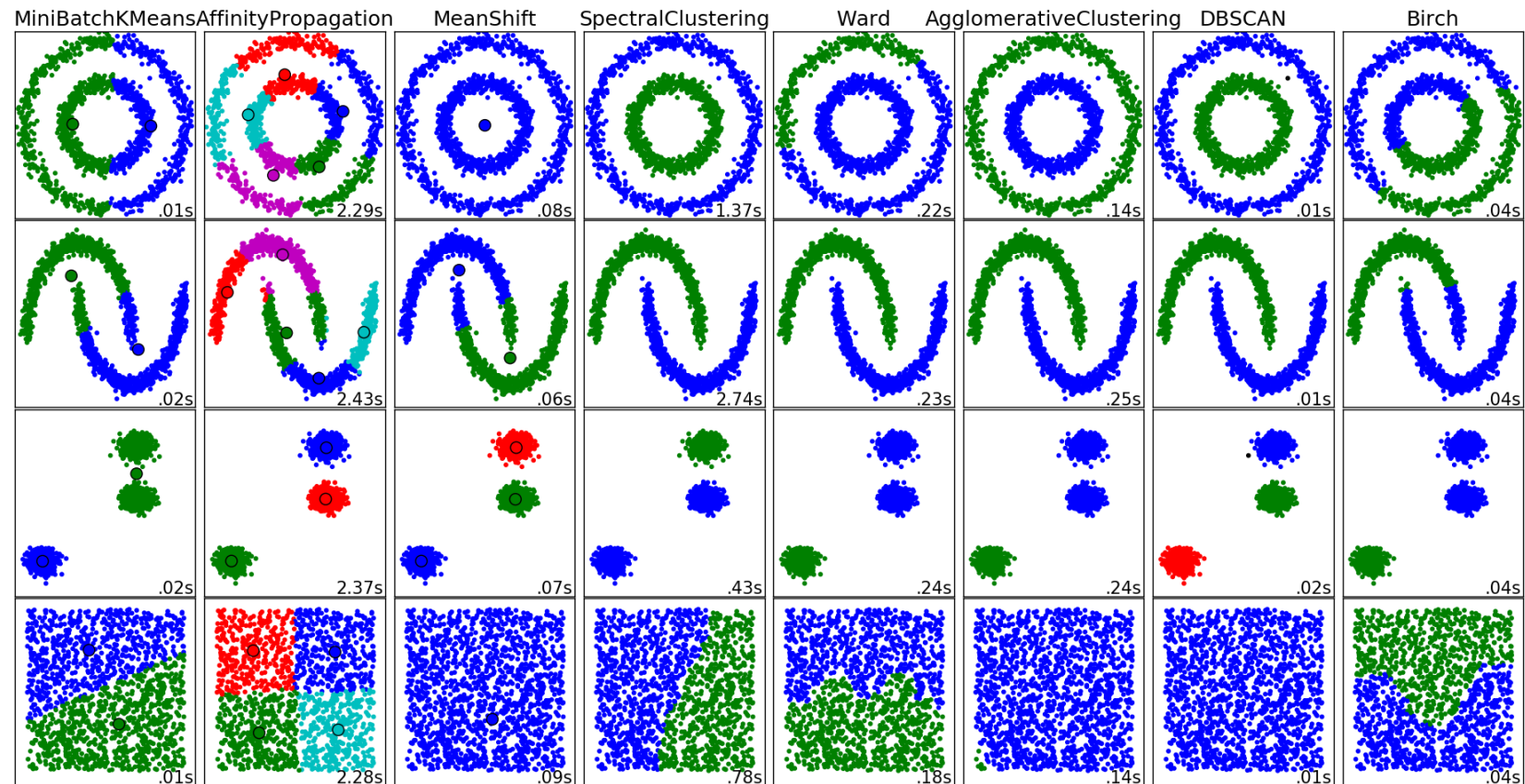




# Типы кластеризации



## Алгоритмы кластеризации



## ЕМ-алгоритм

Имеется выборка  $X^l$ , состоящая из смеси распределений

$$p(x) = \sum_{y=1}^M w_y p_y(x), \sum_{y=1}^M w_y = 1$$

$p_y(x)$  – плотность

$w_y$  – априорная вероятность кластера  $y$



## ЕМ-алгоритм

$$X = R^n,$$

Кластеры  $n$ -мерные гауссовские:

$$p_y = (2\pi)^{-\frac{n}{2}} (\sigma_{y1} \dots \sigma_{yn})^{-1} \exp\left(-\frac{1}{2} \rho_y^2(x, \mu_y)\right)$$

$\mu_y = (\mu_{y1} \dots \mu_{yn})$  - центр кластера  $y$ ,

$\Sigma_y = \text{diag}(\sigma_{y1}^2, \dots, \sigma_{yn}^2)$  - диагональная матрица ковариаций,

$$\rho_y^2(x, x') = \sum_{j=1}^n \sigma_{yj}^{-1} |f_j(x) - f_j(x')|^2$$

## ЕМ-алгоритм

Шаг 1: Выбираем начальные приближения для  $w_y, \mu_y, \Sigma_y$

Шаг 2: do

Шаг 2.1: E-шаг (expectation)

$$g_{iy} = P(y|x_i) = \frac{w_y p_y(x_i)}{\sum_{j=1}^M w_j p_j(x_i)}, y \in Y, i = 1, \dots, l$$

Шаг 2.2: M-шаг (maximization)

$$w_y = \frac{1}{l} \sum_{i=1}^l g_{iy}, y \in Y$$

$$\mu_{yj} = \frac{1}{l w_y} \sum_{i=1}^l g_{iy} f_j(x_i), y \in Y, j = 1, \dots, n$$

$$\sigma_{yj}^2 = \frac{1}{l w_y} \sum_{i=1}^l g_{iy} (f_j(x_i) - \mu_{yj})^2, y \in Y, j = 1, \dots, n$$

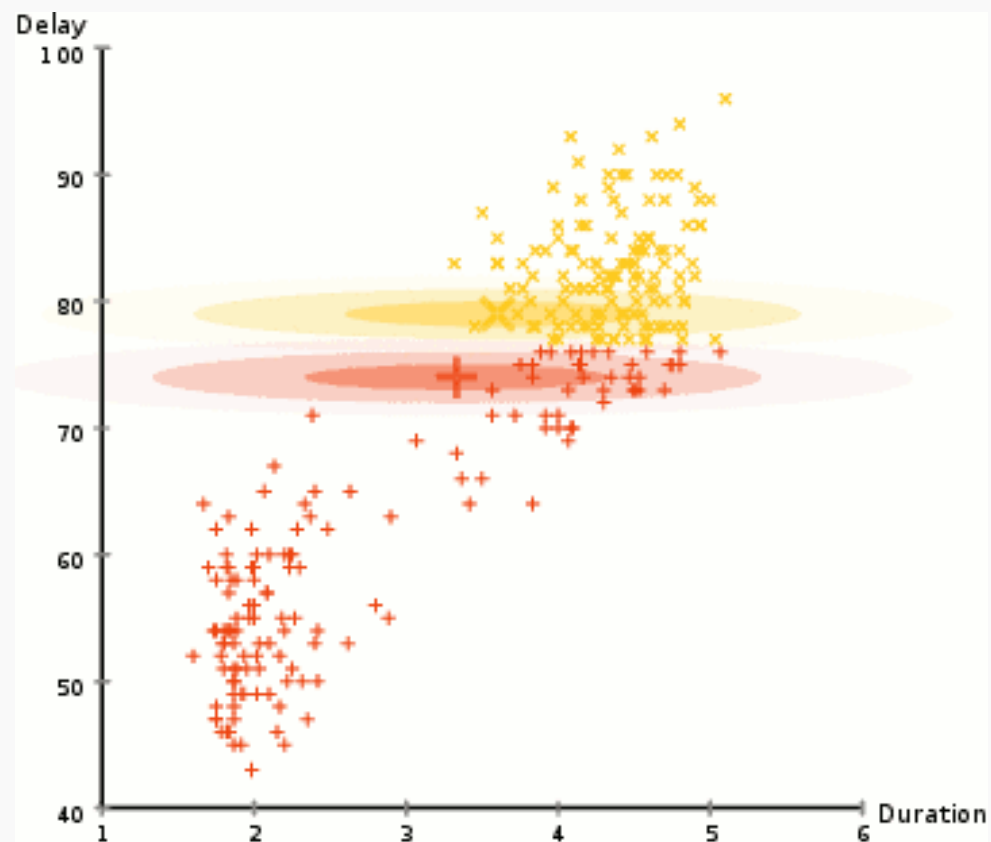
$$y_i = \arg \max_{y \in Y} g_{iy}, i = 1, \dots, l$$

while Не будут изменяться  $y_i$

# Алгоритмы кластеризации



## ЕМ-алгоритм



## Алгоритм K-Means

Шаг 1: Выбираем начальные приближения для  $\mu_y$  (положения центров)

Шаг 2: do

Шаг 2.1: Аналог E-шага

Относим каждый объект  $x_i$  к ближайшему центру:

$$y_i = \arg \min \rho(x_i, \mu_y), y \in Y, i = 1, \dots, l$$

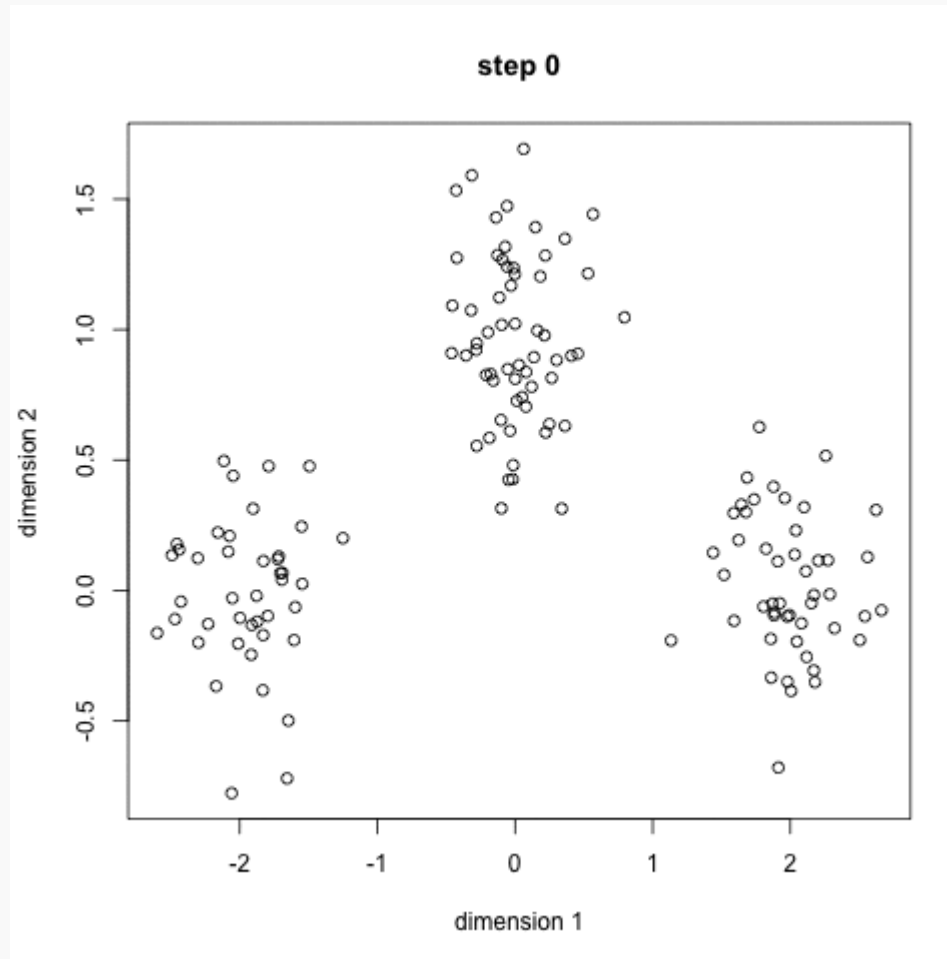
Шаг 2.2: Аналог M-шага

Вычисляем новые положения центров

$$\mu_y = \frac{\sum_{i=1}^l [y_i=y] f_j(x_i)}{\sum_{i=1}^l [y_i=y]}, y \in Y, j = 1, \dots, n$$

while Не будут изменяться  $y_i$

## Алгоритм K-Means



## Алгоритм K-Means

Оптимизируем среднее внутриклассовое расстояние:

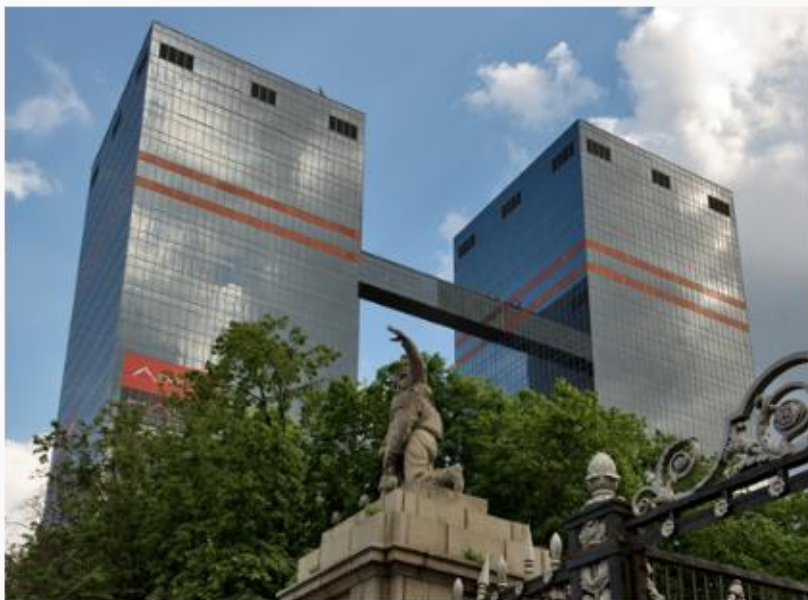
$$F = \frac{\sum_{i < j} [y_i = y_j] \rho(x_i, x_j)}{\sum_{i < j} [y_i = y_j]} \rightarrow \min$$



# Алгоритмы кластеризации



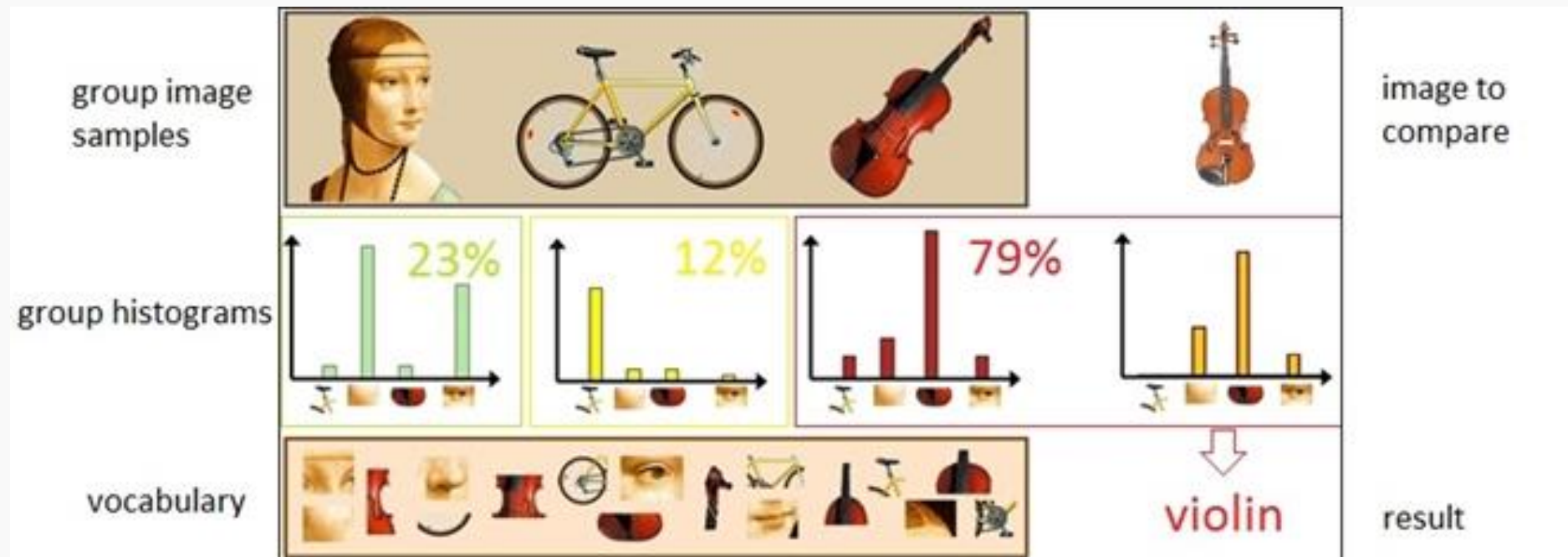
## Алгоритм K-Means



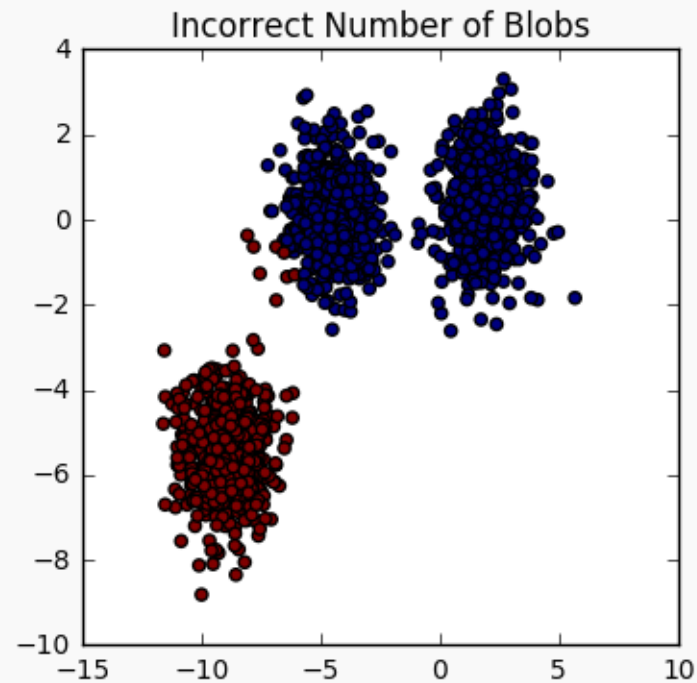
64 цвета  
(кластера)



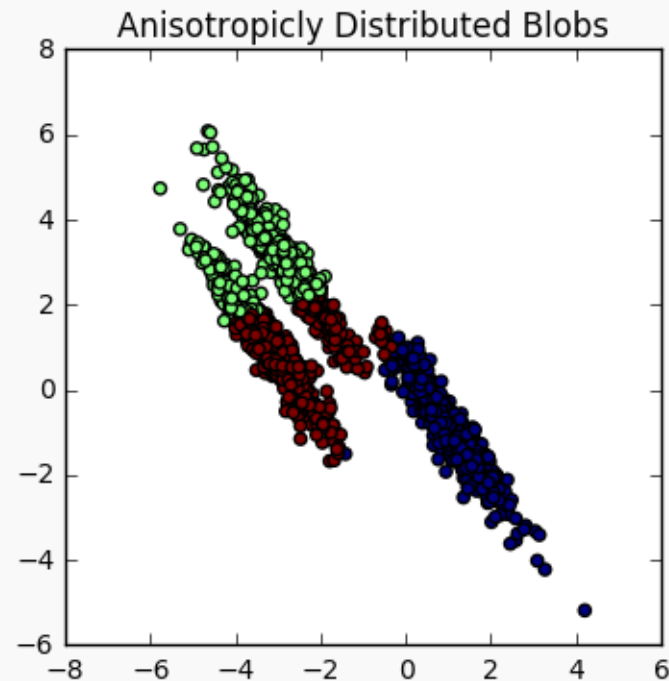
## Bag of visual words



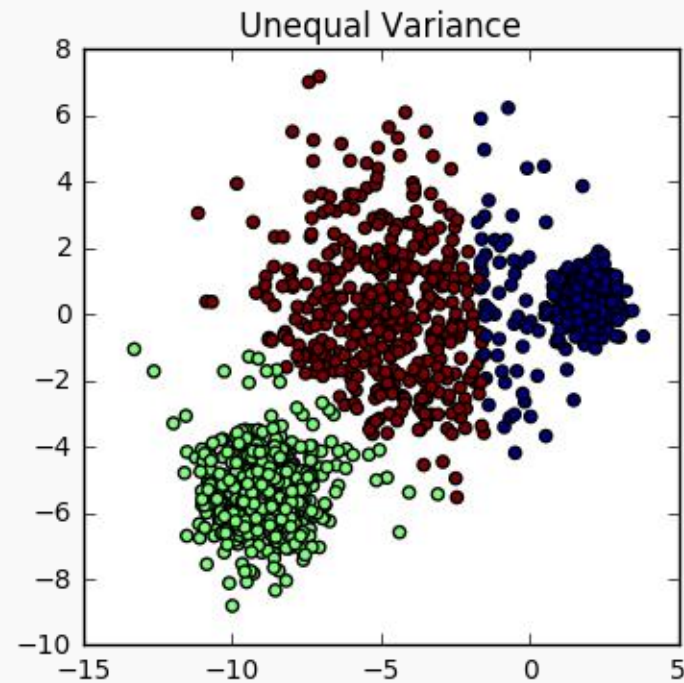
## Алгоритм K-Means



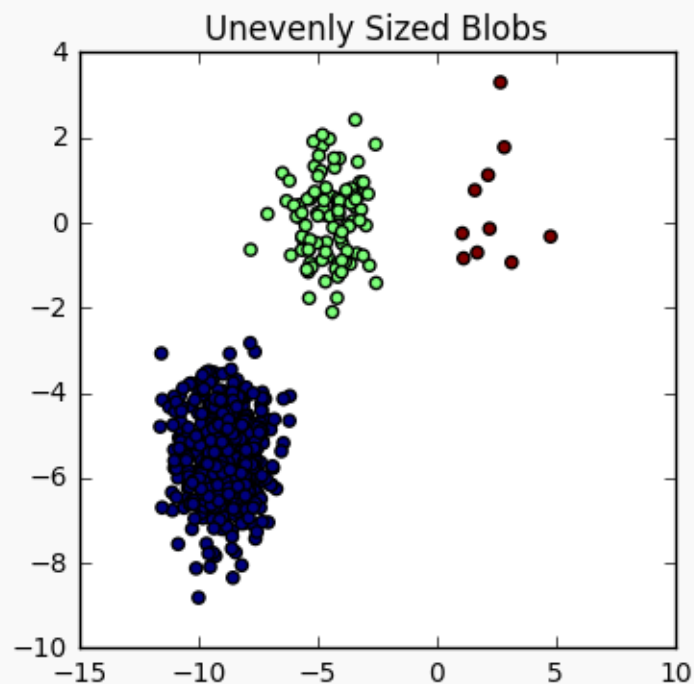
## Алгоритм K-Means



## Алгоритм K-Means



## Алгоритм K-Means





## Модификации K-Means

### Mini-Batch K-Means

- 1) Если данных достаточно много, то вычисление расстояний от всех объектов до центров кластеров может занять достаточно много времени
- 2) Решение: На каждом шаге выбирать из набора данных случайную подвыборку

## Модификации K-Means

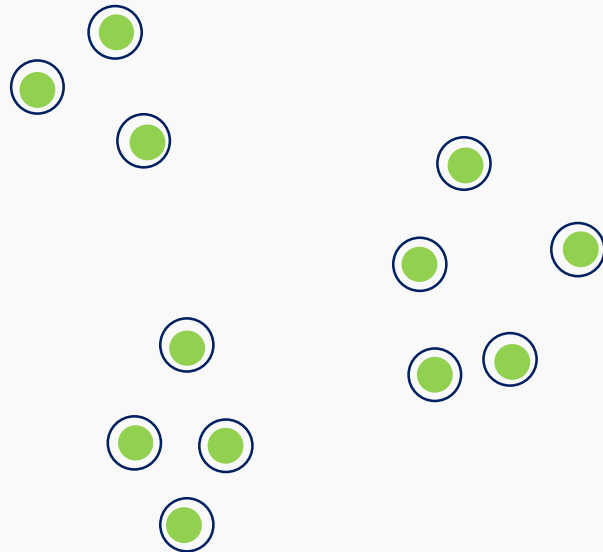
### K-Means++

- 1) Выбор начального приближения центров кластеров значительно влияет на скорость сходимости алгоритма
- 2) Выбираем начальные положения центров на максимальном расстоянии друг от друга
- 3) Решение:
  - 1) Выбираем начальные центры из равномерного распределения на выборке
  - 2) Каждый следующий центр выбираем случайно из оставшихся точек так, чтобы вероятность выбора точки была пропорциональна квадрату расстояний от неё до ближайшего центра

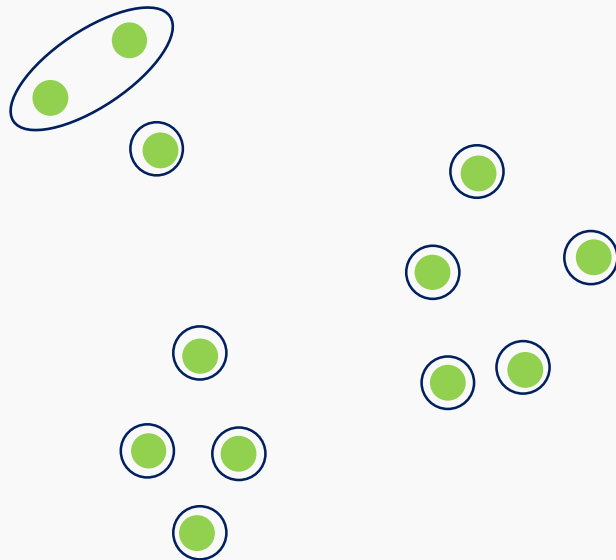
## Иерархическая кластеризация

- 1) Агломеративная
- 2) Дивизионная

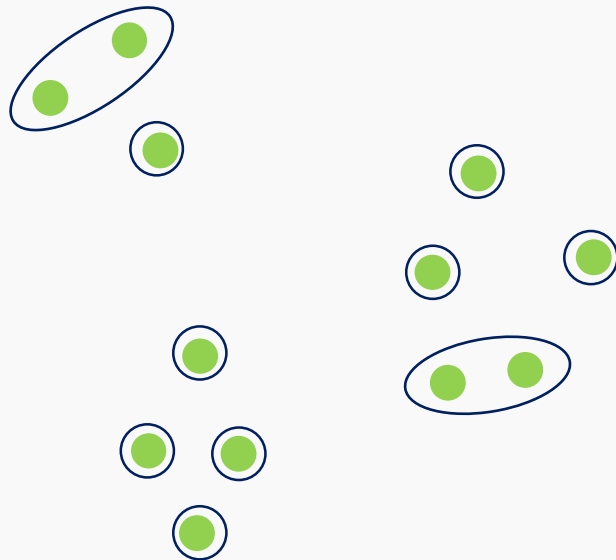
## Агломеративная кластеризация



## Агломеративная кластеризация

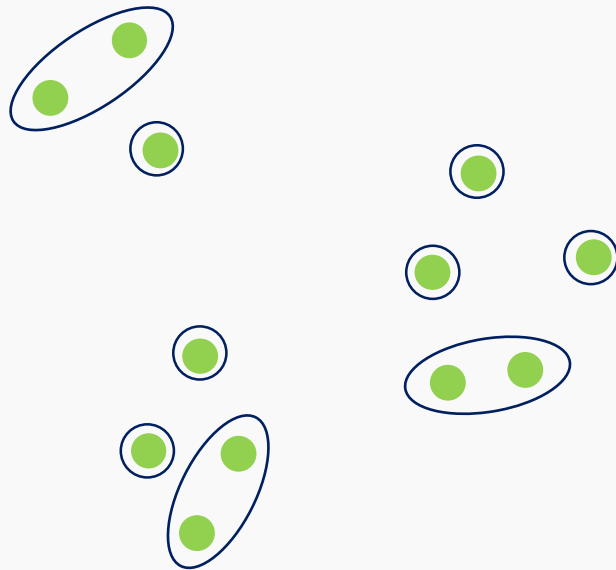


## Агломеративная кластеризация

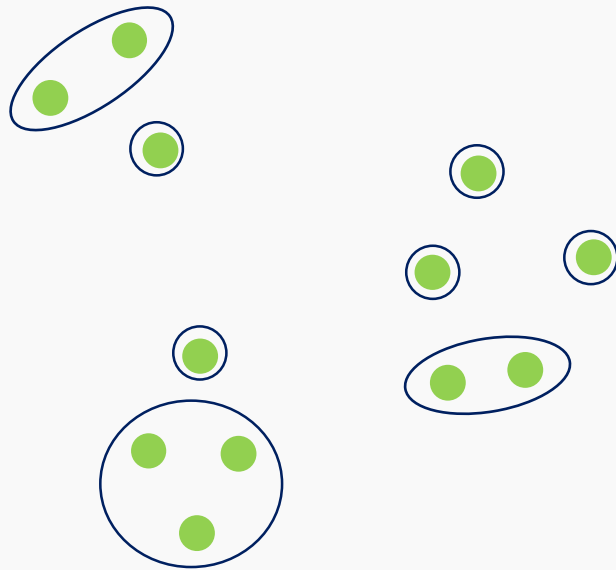




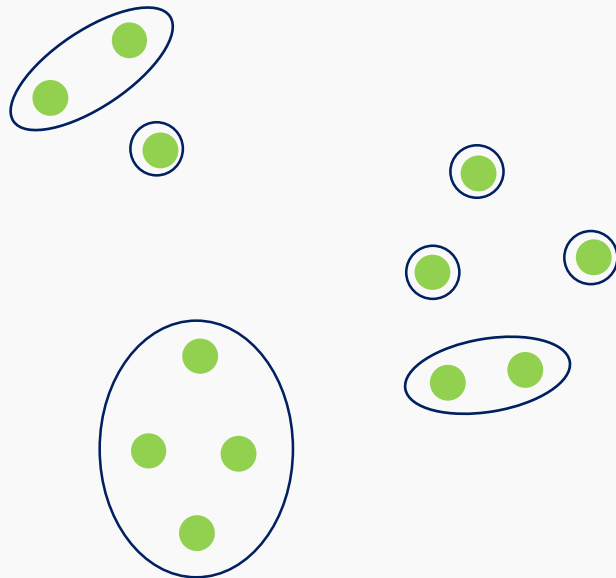
## Агломеративная кластеризация



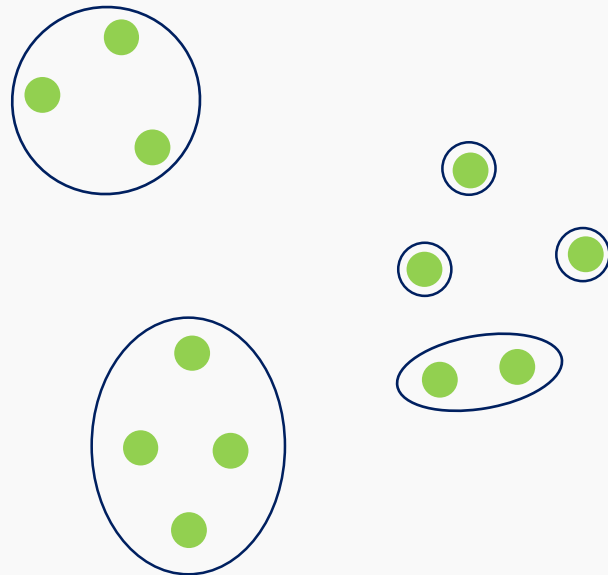
## Агломеративная кластеризация



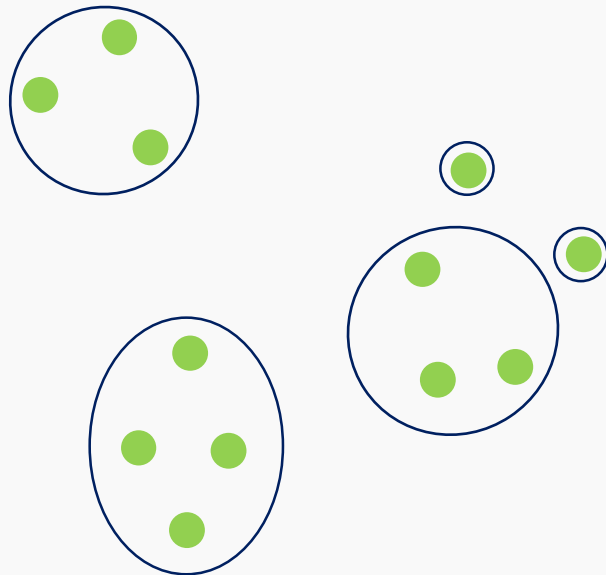
## Агломеративная кластеризация



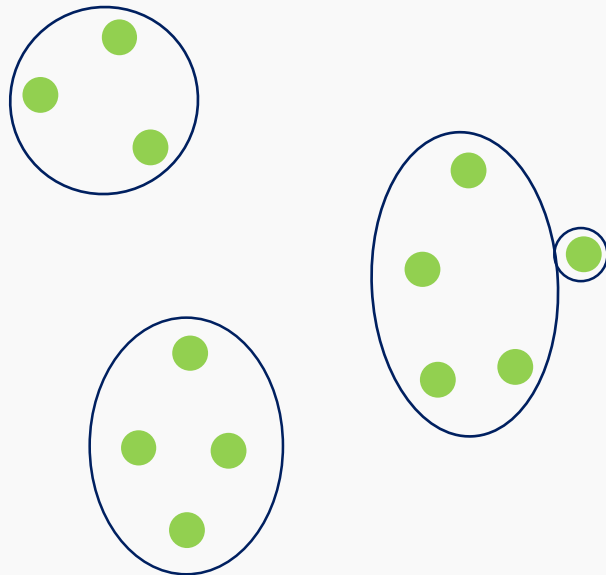
## Агломеративная кластеризация



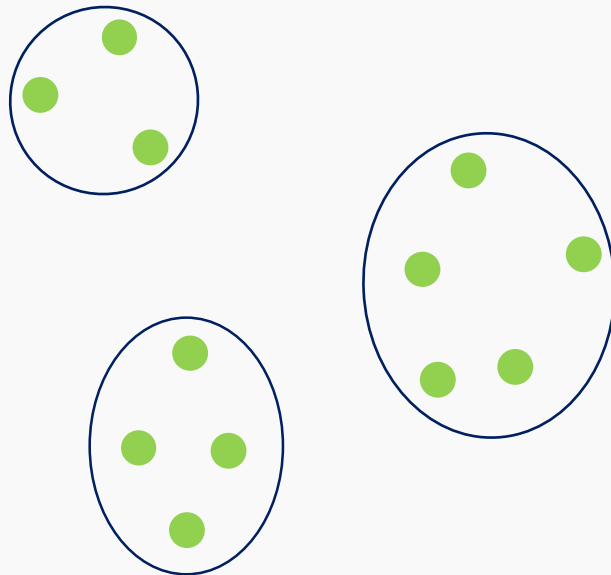
## Агломеративная кластеризация



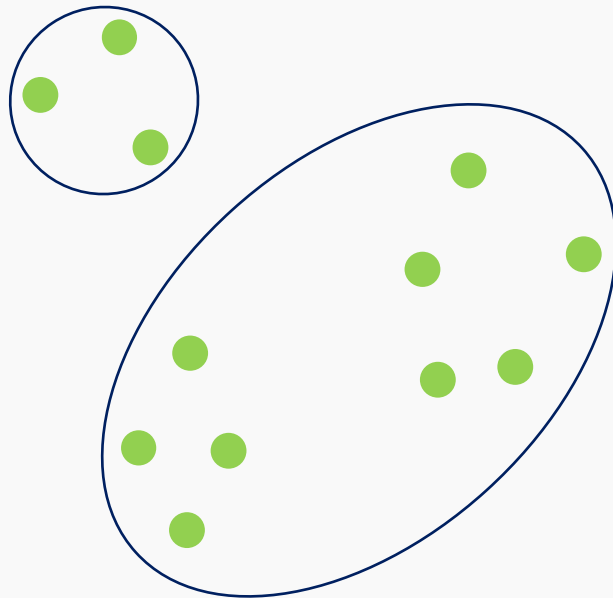
## Агломеративная кластеризация



## Агломеративная кластеризация

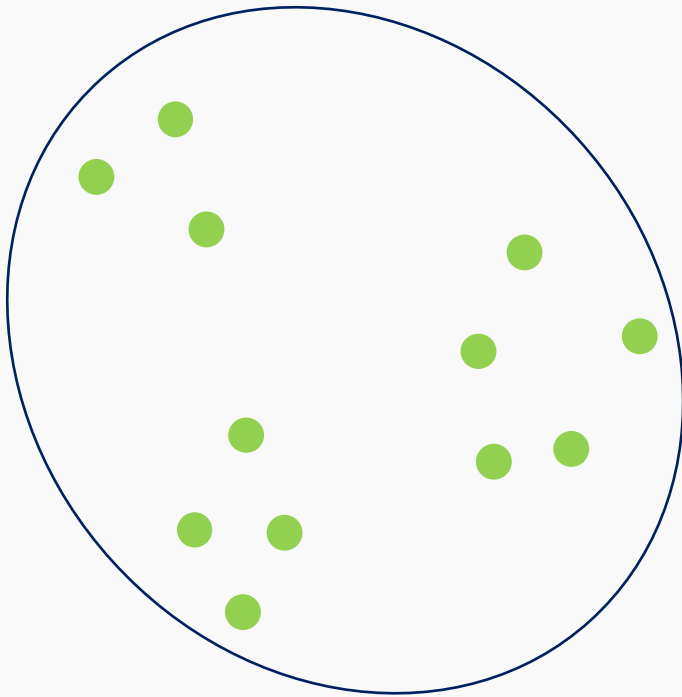


## Агломеративная кластеризация





## Агломеративная кластеризация



## Формула Ланса-Уильямса

$$\begin{aligned} R(U \cup V, S) = & \alpha_U \cdot R(U, S) + \\ & + \alpha_V \cdot R(V, S) + \\ & + \beta \cdot R(U, V) + \\ & + \gamma \cdot |R(U, S) - R(V, S)|, \end{aligned}$$

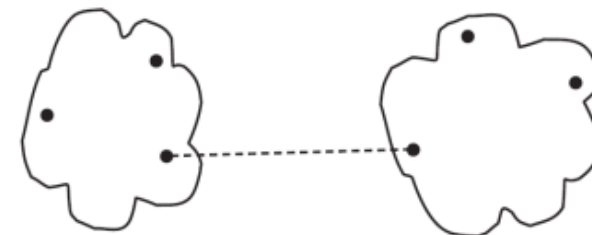
где  $\alpha_U, \alpha_V, \beta, \gamma$  — числовые параметры.

# Алгоритмы кластеризации



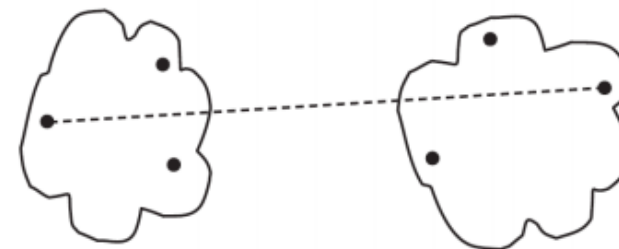
Расстояние ближайшего соседа

$$R^b(W, S) = \min_{w \in W, s \in S} \rho(w, s);$$
$$\alpha_U = \alpha_V = \frac{1}{2}, \quad \beta = 0, \quad \gamma = -\frac{1}{2}.$$



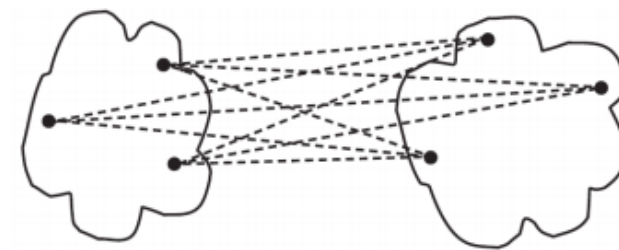
Расстояние дальнего соседа

$$R^d(W, S) = \max_{w \in W, s \in S} \rho(w, s);$$
$$\alpha_U = \alpha_V = \frac{1}{2}, \quad \beta = 0, \quad \gamma = \frac{1}{2}.$$



Групповое среднее расстояние

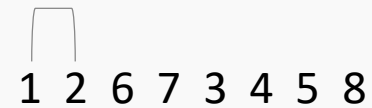
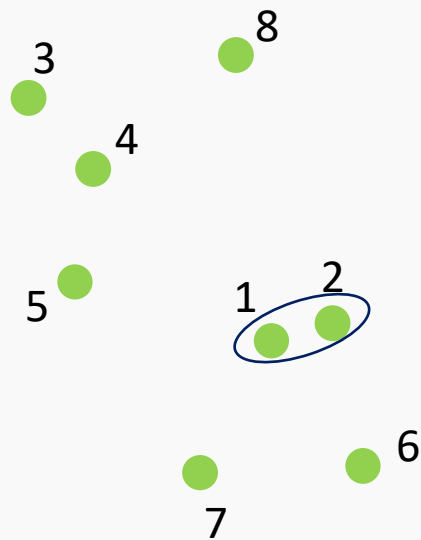
$$R^r(W, S) = \frac{1}{|W||S|} \sum_{w \in W} \sum_{s \in S} \rho(w, s);$$
$$\alpha_U = \frac{|U|}{|W|}, \quad \alpha_V = \frac{|V|}{|W|}, \quad \beta = \gamma = 0.$$



# Алгоритмы кластеризации



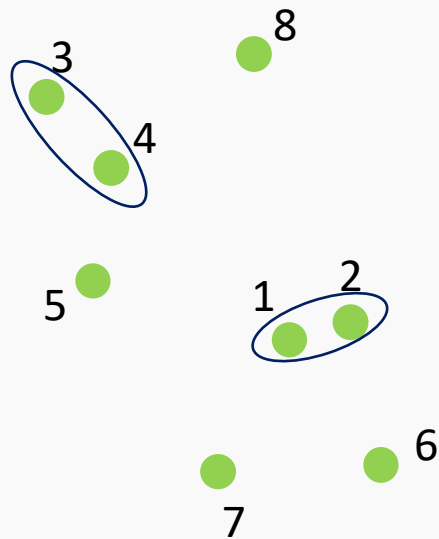
## Дендограмма



# Алгоритмы кластеризации



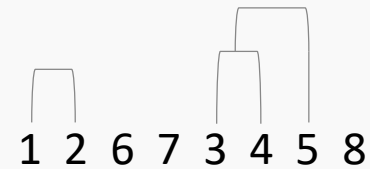
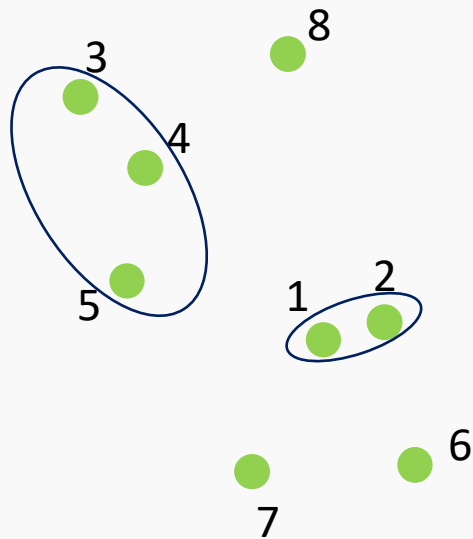
## Дендограмма



# Алгоритмы кластеризации



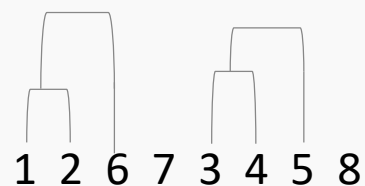
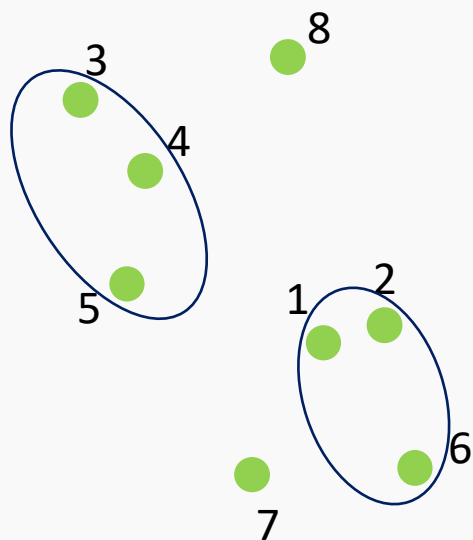
## Дендограмма



# Алгоритмы кластеризации



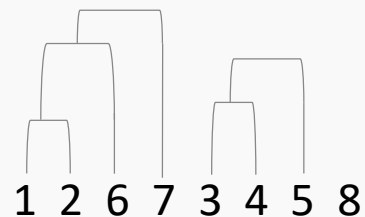
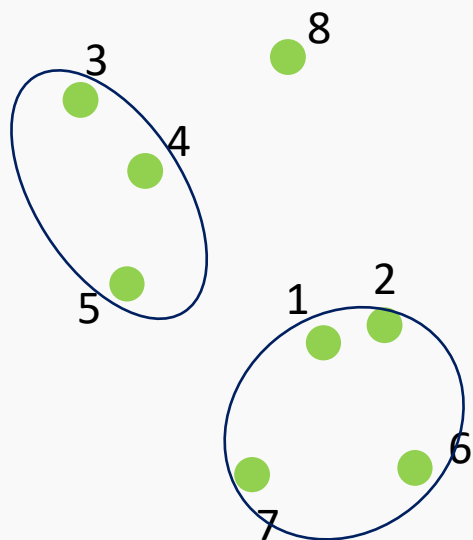
## Дендограмма



# Алгоритмы кластеризации



## Дендограмма

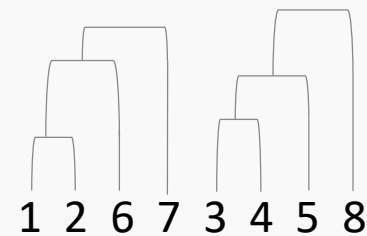
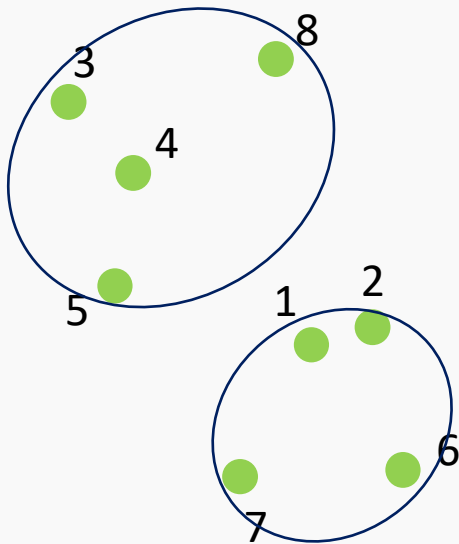




# Алгоритмы кластеризации



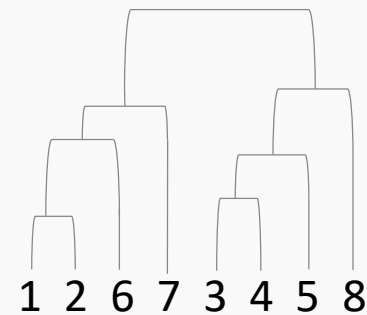
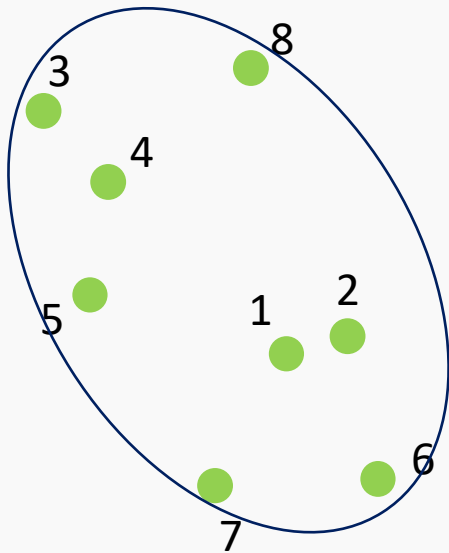
## Дендограмма



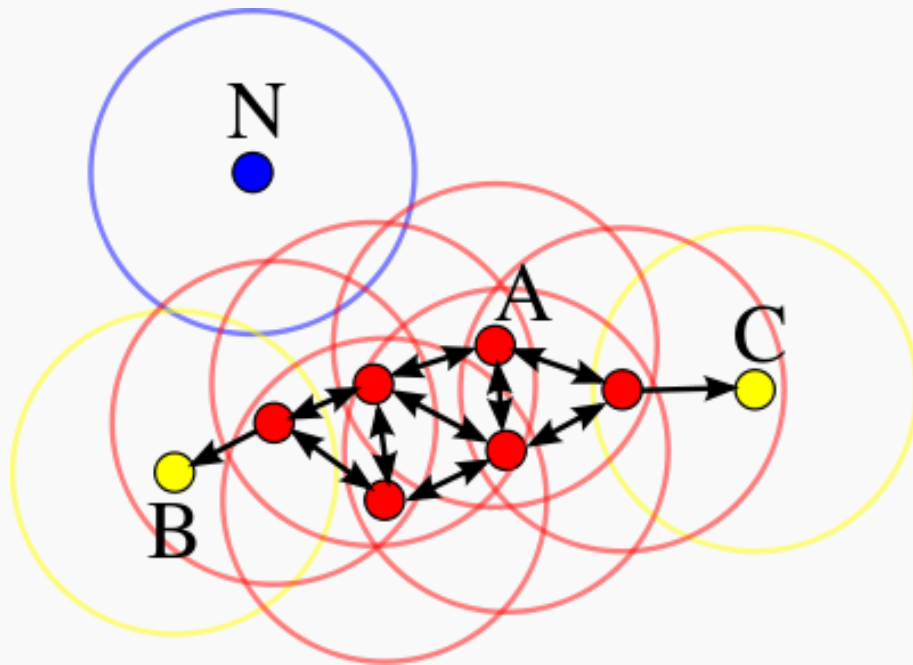
# Алгоритмы кластеризации



## Дендограмма



## Алгоритм DBSCAN



Шаг 1: Помечаем все точки как основные, пограничные или шумовые

Шаг 2: Отбрасываем точки шума

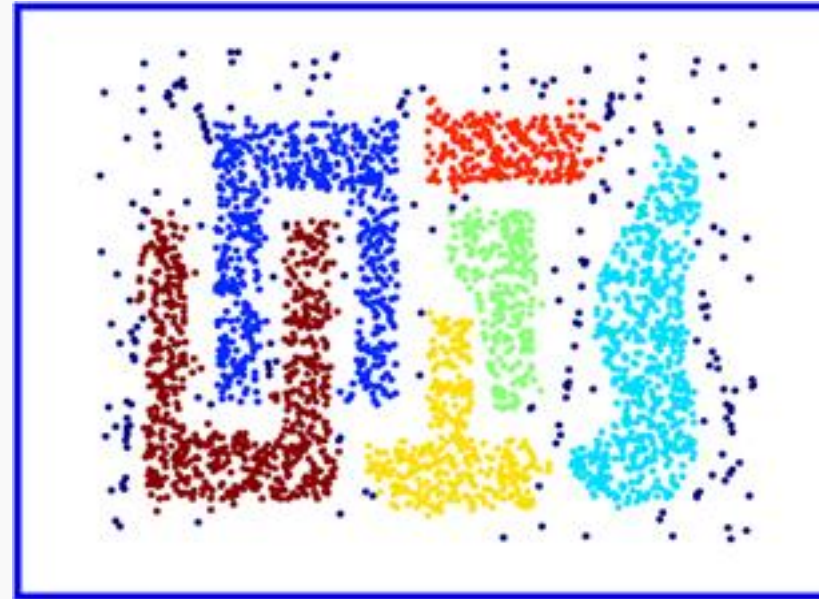
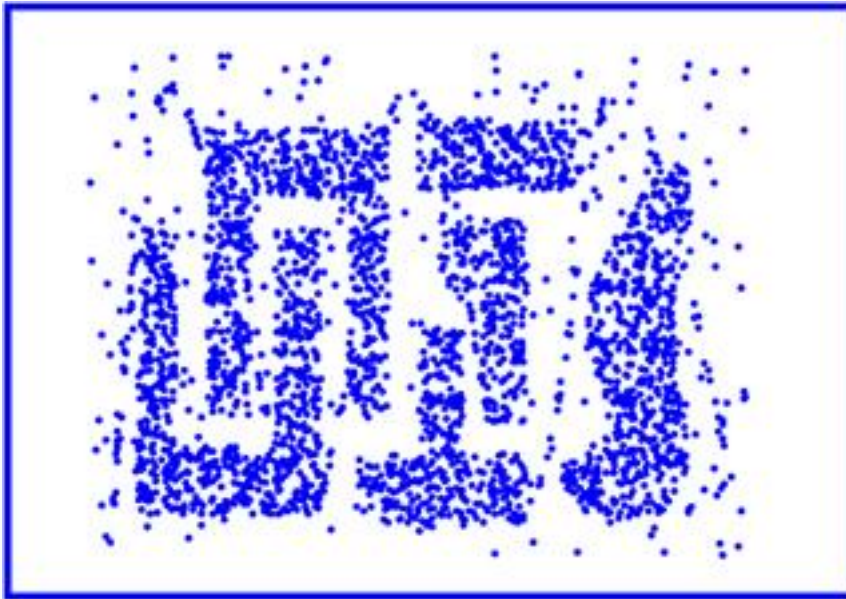
Шаг 3: Соединяем все основные точки, находящиеся на расстоянии  $Eps$  друг от друга

Шаг 4: Соединяем каждую группу объединённых точек в отдельный кластер

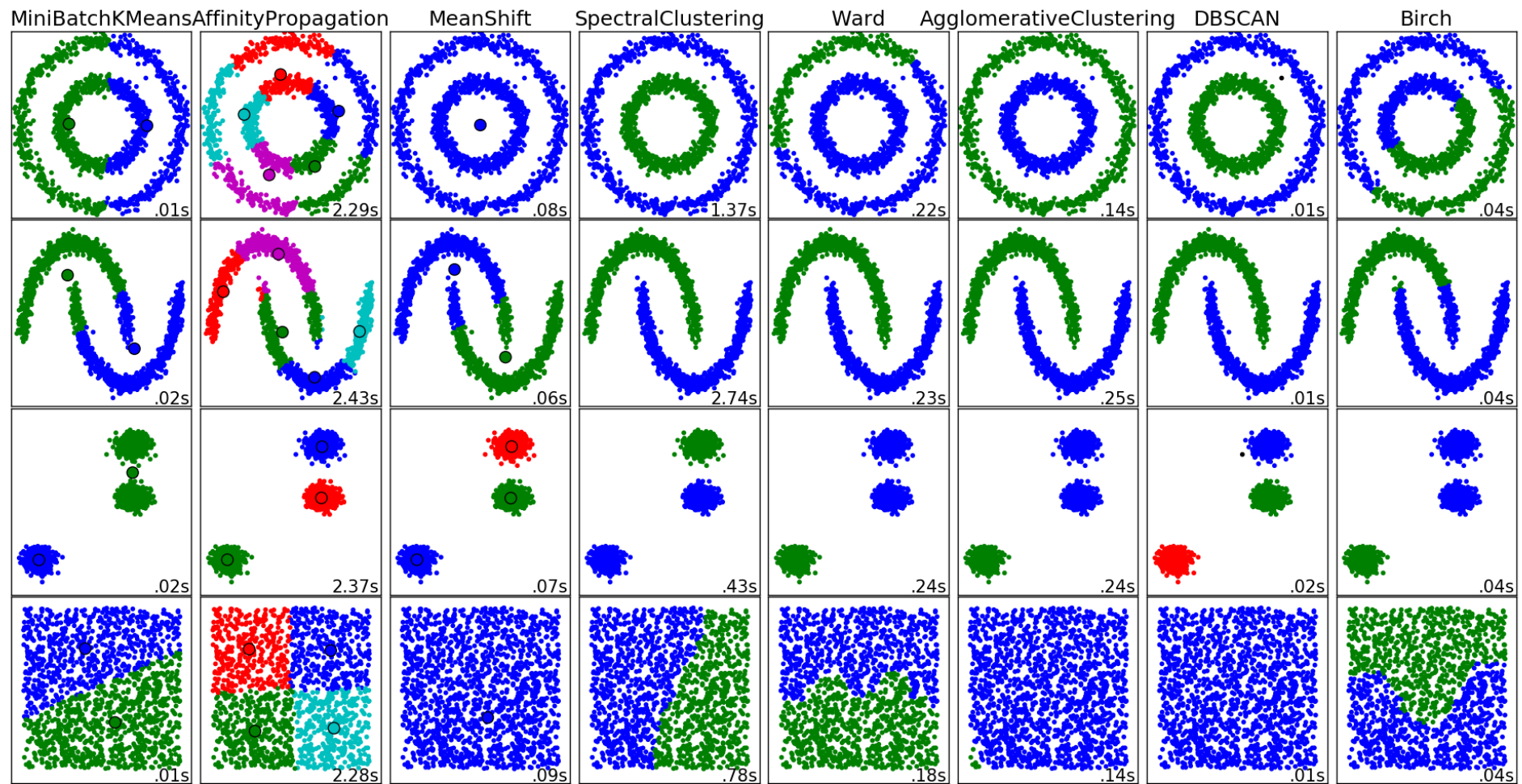
# Алгоритмы кластеризации



## Алгоритм DBSCAN



# Алгоритмы кластеризации



## Оценка качества кластеризации

Внутриклассовое расстояние:

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

Межкластерное расстояние:

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

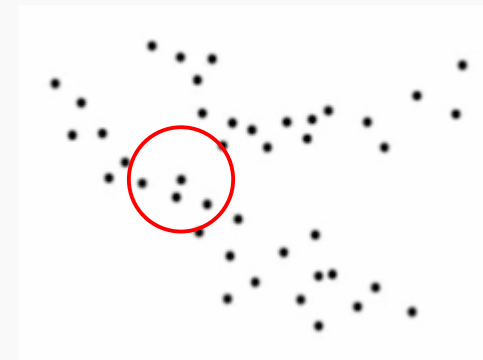
## Оценка качества кластеризации

Коэффициент силуэта (Silhouette Coefficient)

$D_{1i}$  - среднее расстояние от объекта  $i$  до всех остальных объектов внутри кластера, в котором находится объект  $i$

$D_{2i}$  - среднее расстояние от объекта  $i$  до всех остальных объектов внутри ближайшего кластера, в котором не находится объект  $i$

$$S = \frac{D_{2i} - D_{1i}}{\max(D_{1i}, D_{2i})}$$







## Оценка качества кластеризации

### Однородность (Homogeneity)

Кластеры состоят из объектов одного класса

labels\_true, labels\_pred

homogeneity_score([0, 0, 1, 1], [0, 0, 1, 2])	1.0
homogeneity_score([0, 0, 1, 1], [0, 1, 2, 3])	1.0
homogeneity_score([0, 0, 1, 1], [0, 1, 0, 1])	0.0
homogeneity_score([0, 0, 1, 1], [0, 0, 0, 0])	0.0

### Полнота (Completeness)

Объекты из одного класса принадлежат одному кластеру

completeness_score([0, 0, 1, 1], [0, 0, 0, 0])	1.0
completeness_score([0, 1, 2, 3], [0, 0, 1, 1])	1.0
completeness_score([0, 0, 1, 1], [0, 1, 0, 1])	0.0
completeness_score([0, 0, 0, 0], [0, 1, 2, 3])	0.0

V-мера: 
$$V = 2 * \frac{H * C}{H + C}$$

Разметка используется для проверки результата кластеризации





**Контакты:**  
**[a.spasenov@corp.mail.ru](mailto:a.spasenov@corp.mail.ru)**  
**[alex\\_spasenov](#) (Skype)**

**Спасибо за внимание!**