

# Лекция 9

## Кластеризация и визуализация данных. Поиск аномалий.

Кантонистова Е.О.

ВШЭ, 2019

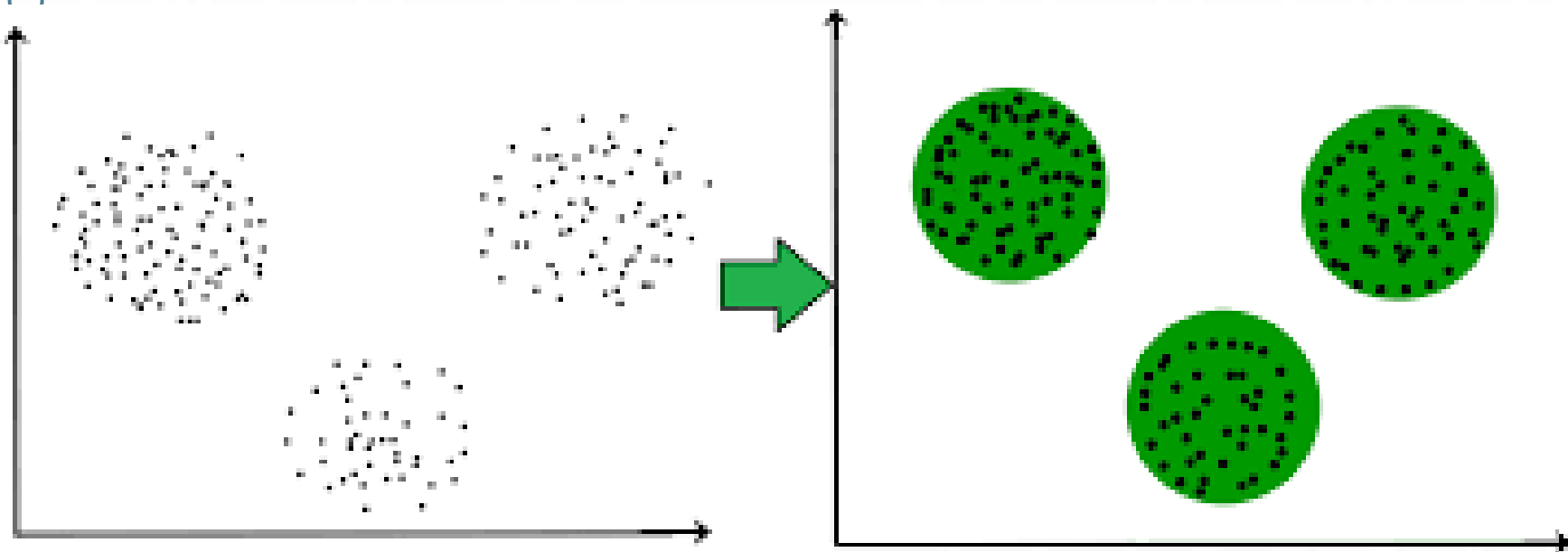
# КЛАСТЕРИЗАЦИЯ

# КЛАСТЕРИЗАЦИЯ

Даны объекты  $x_1, \dots, x_l, x_i \in X$ .

- Требуется выявить в данных  $K$  кластеров – таких областей, что объекты внутри одного кластера похожи друг на друга, а объекты из разных кластеров друг на друга не похожи.
- Формализация задачи: необходимо построить алгоритм  $a: X \rightarrow \{1, \dots, K\}$ , сопоставляющий каждому объекту  $x$  номер кластера.

# КЛАСТЕРИЗАЦИЯ



# МЕТРИКИ КАЧЕСТВА КЛАСТЕРИЗАЦИИ

- ***Внешние метрики*** – используют информацию об истинных метках объектов
- ***Внутренние метрики*** – оценивают качество кластеризации, основываясь только на наборе данных.

The background features a light gray circular pattern. The corners are decorated with stylized circuit lines: dark blue in the top-left and top-right, and light blue in the bottom-left and bottom-right. These lines include small circles at various points, resembling nodes or components of a circuit.

# ВНУТРЕННИЕ МЕТРИКИ КАЧЕСТВА КЛАСТЕРИЗАЦИИ

# ВНУТРИКЛАСТЕРНОЕ РАССТОЯНИЕ

Пусть  $c_k$  - центр  $k$ -го кластера

Внутри кластера все объекты максимально похожи, поэтому наша цель – минимизировать внутрикластерное расстояние:

$$\sum_{k=1}^K \sum_{i=1}^l [a(x_i) = k] \rho(x_i, c_k) \rightarrow \min_a$$

# МЕЖКЛАСТЕРНОЕ РАССТОЯНИЕ

Объекты из разных кластеров должны быть как можно менее похожи друг на друга, поэтому мы максимизируем межкластерное расстояние:

$$\sum_{i,j=1}^l [a(x_i) \neq a(x_j)] \rho(x_i, x_j) \rightarrow \max_a$$



# ИНДЕКС ДАННА (DUNN INDEX)

Хотим минимизировать внутрикластерное расстояние и одновременно максимизировать межкластерное расстояние:

$$\frac{\min_{1 \leq k < k' \leq K} d(k, k')}{\max_{1 \leq k \leq K} d(k)} \rightarrow \max_a$$

Здесь  $d(k, k')$  – расстояние между кластерами  $k$  и  $k'$ ,  
 $d(k)$  – внутрикластерное расстояние для  $k$ -го кластера.

# K-MEANS

Дано: выборка  $x_1, \dots, x_l$

Параметр: число кластеров  $K$

*Идея метода - минимизация внутрикластерного расстояния*

$$\sum_{k=1}^K \sum_{i=1}^l [a(x_i) = k] \rho(x_i, c_k) \rightarrow \min_a$$

с  $\rho(a, b) = (a - b)^2$ , т.е.

$$\sum_{k=1}^K \sum_{i=1}^l [a(x_i) = k] (x_i - c_k)^2 \rightarrow \min_a$$

# K-MEANS

Дано: выборка  $x_1, \dots, x_l$

Параметр: число кластеров  $K$

Начало: случайно выбрать центры кластеров  $c_1, \dots, c_K$

Повторять по очереди до сходимости:

- отнести каждый объект к ближайшему центру

$$y_i = \operatorname{argmin}_{j=1, \dots, K} \rho(x_i, c_j)$$

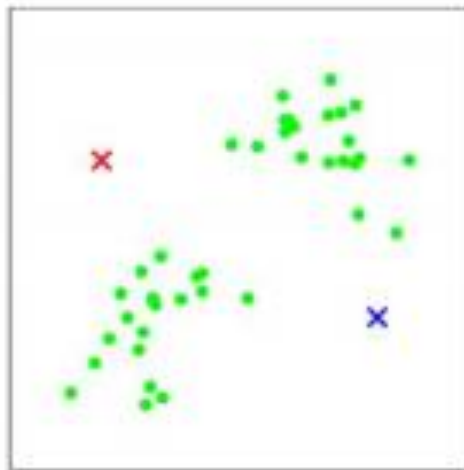
- переместить центр каждого кластера в центр тяжести

$$c_j = \frac{\sum_{i=1}^l x_i [y_i = j]}{\sum_{i=1}^l [y_i = j]}$$

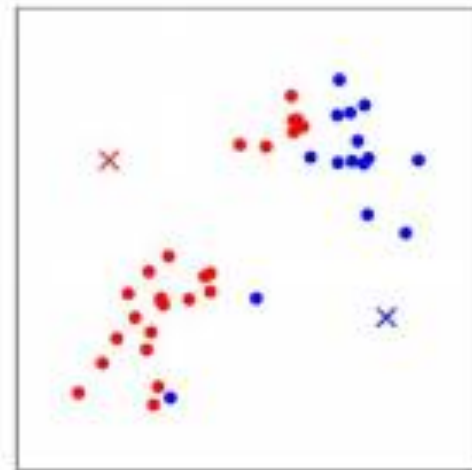
# K-MEANS (ДВА КЛАСТЕРА)



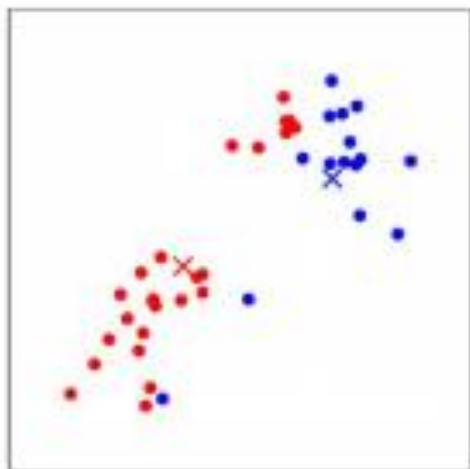
(a)



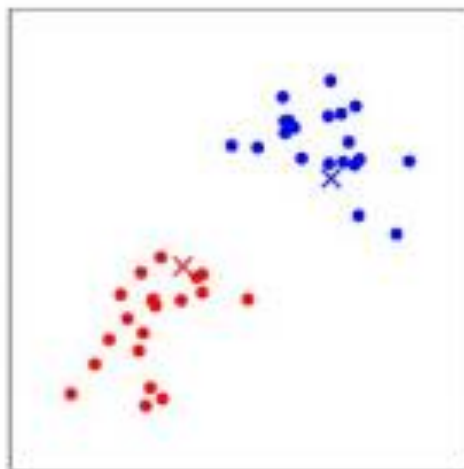
(b)



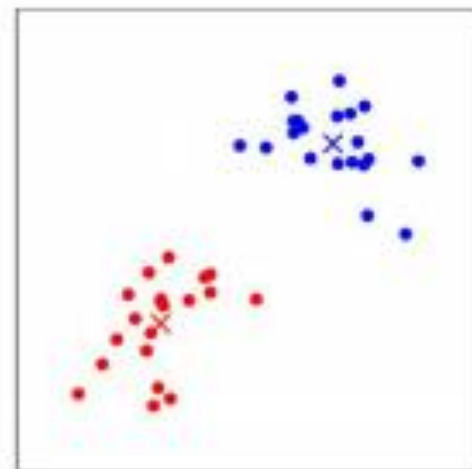
(c)



(d)



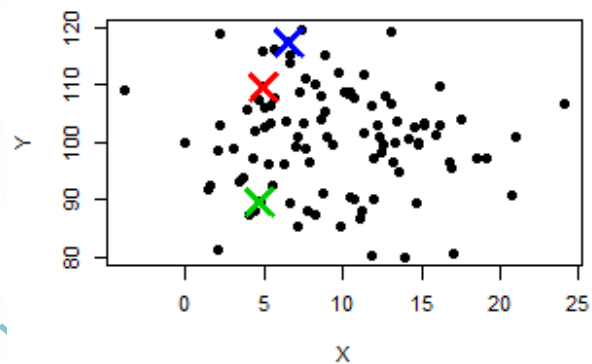
(e)



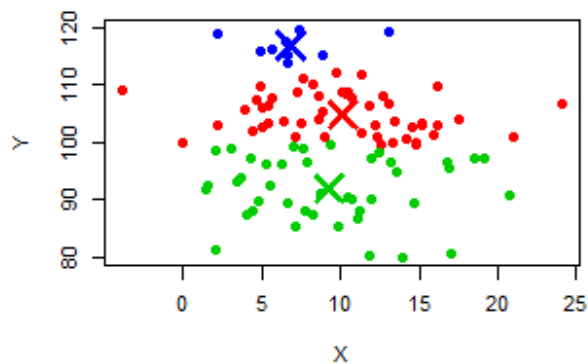
(f)

# K-MEANS (ТРИ КЛАСТЕРА)

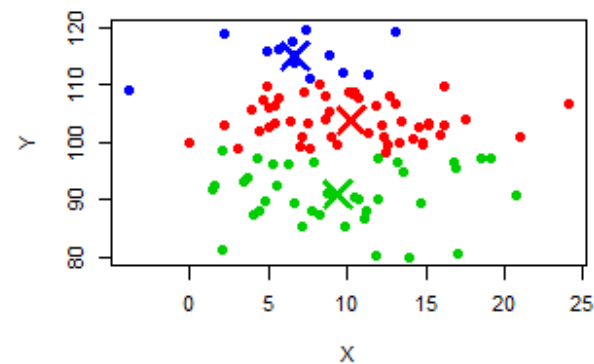
Iteration 1



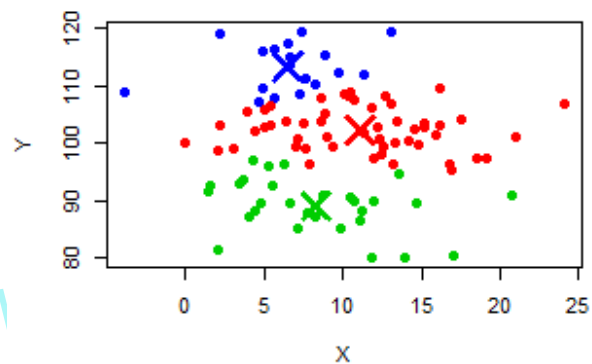
Iteration 2



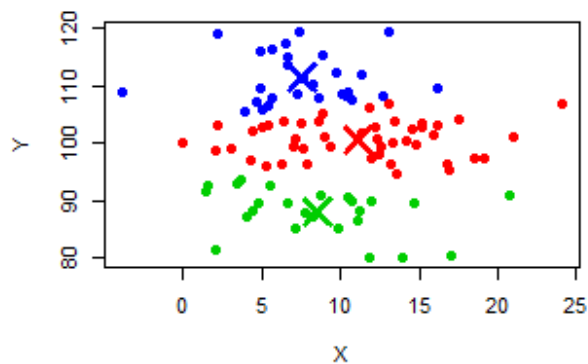
Iteration 3



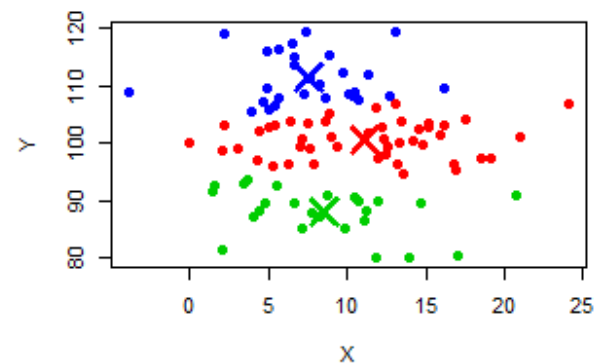
Iteration 6



Iteration 9



Converged!



# K-MEANS ДЛЯ СЖАТИЯ ИЗОБРАЖЕНИЙ



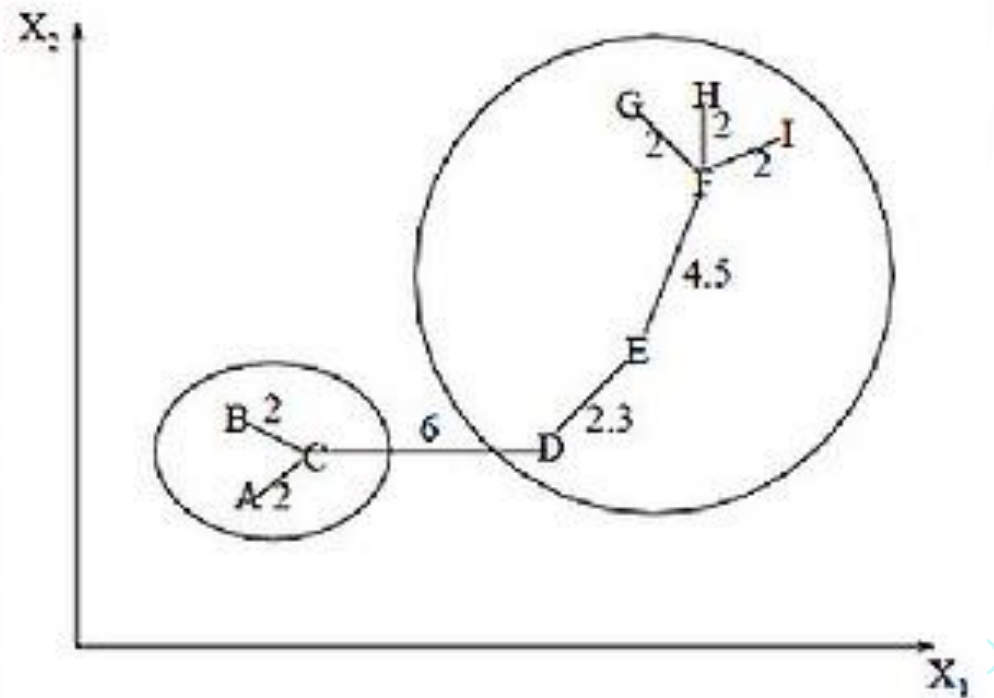
# ГРАФОВЫЕ МЕТОДЫ КЛАСТЕРИЗАЦИИ

- выборка представляется в виде графа, где в вершинах стоят объекты, а на рёбрах – расстояния между ними

Алгоритм выделения связных компонент:

1) из графа удаляются все ребра, для которых расстояния больше некоторого значения  $R$

2) Кластеры – объекты, попадающие в одну компоненту связности



# ИЕРАРХИЧЕСКАЯ КЛАСТЕРИЗАЦИЯ

Иерархия кластеров:

- на верхнем уровне – один большой кластер
- на нижнем уровне -  $l$  кластеров, каждый из которых состоит из одного объекта



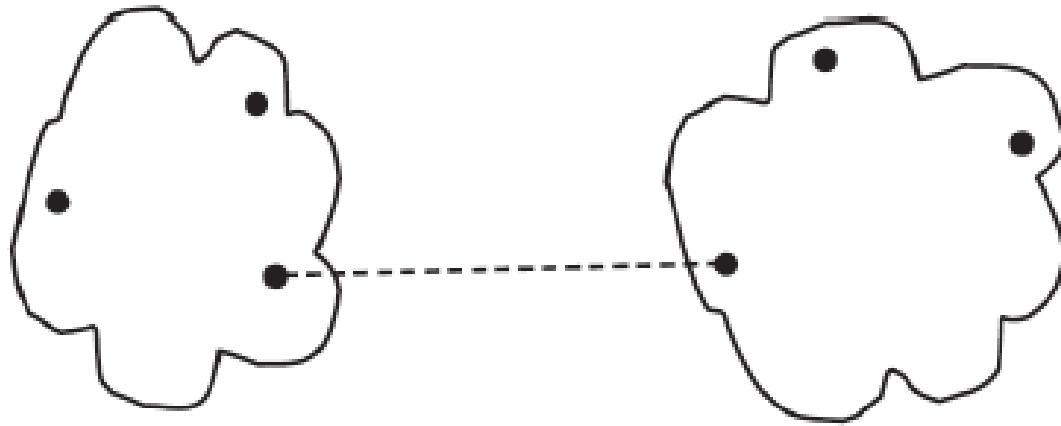
# ИЕРАРХИЧЕСКАЯ КЛАСТЕРИЗАЦИЯ

Алгоритм Ланса-Уильямса:

- первый шаг: один кластер = один объект
- на каждом следующем шаге объединяем два наиболее похожих кластера (по некоторой метрике  $d$ ) с предыдущего шага

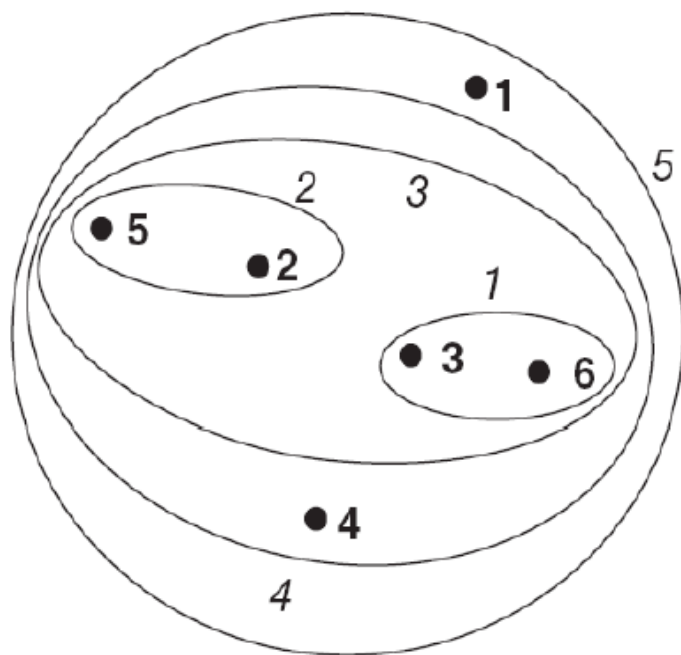
# ПРИМЕР РАССТОЯНИЯ МЕЖДУ КЛАСТЕРАМИ

Расстояние  $d$  между кластерами  $W$  и  $S$  – расстояние до ближайшего соседа:

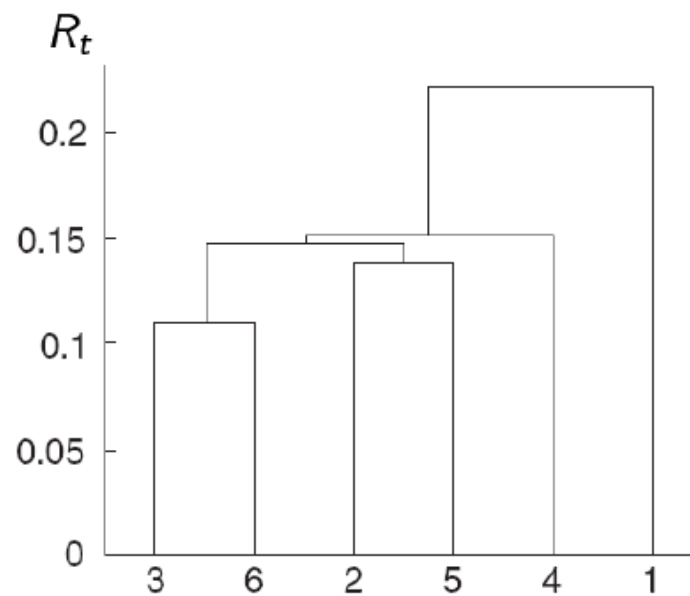


# ПРИМЕР

Диаграмма вложения

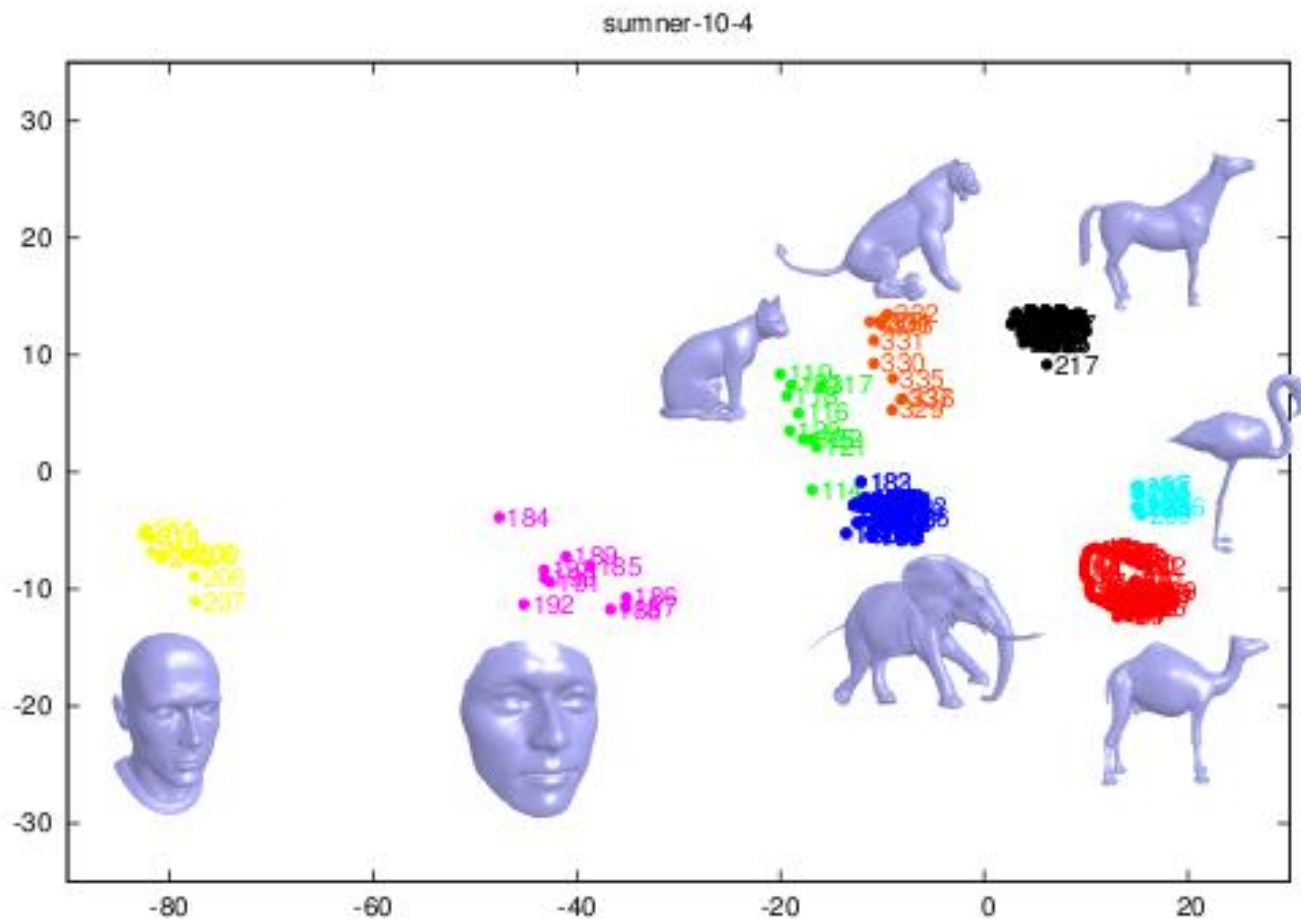


Дендрограмма



# ВИЗУАЛИЗАЦИЯ

Задача визуализации состоит в отображении объектов в 2х- или 3хмерное пространство с сохранением отношений между ними.



# MULTIDIMENSIONAL SCALING (MDS)

Идея метода – минимизация квадратов отклонений между исходными и новыми попарными расстояниями:

$$\sum_{i \neq j}^l (\rho(x_i, x_j) - \rho(z_i, z_j))^2 \rightarrow \min_{z_1, \dots, z_l}$$

- достаточно уметь вычислять расстояние  $\rho$  между объектами для применения метода

# TSNE

*t-SNE – t-distributed stochastic neighbor embedding*

- *При проекции нам важно не сохранение расстояний между объектами, а сохранение пропорций:*

$$\rho(x_1, x_2) = \alpha \rho(x_1, x_3) \Rightarrow \rho(z_1, z_2) = \alpha \rho(z_1, z_3)$$

# БЛИЗОСТЬ ОБЪЕКТОВ В ИСХОДНОМ ПРОСТРАНСТВЕ

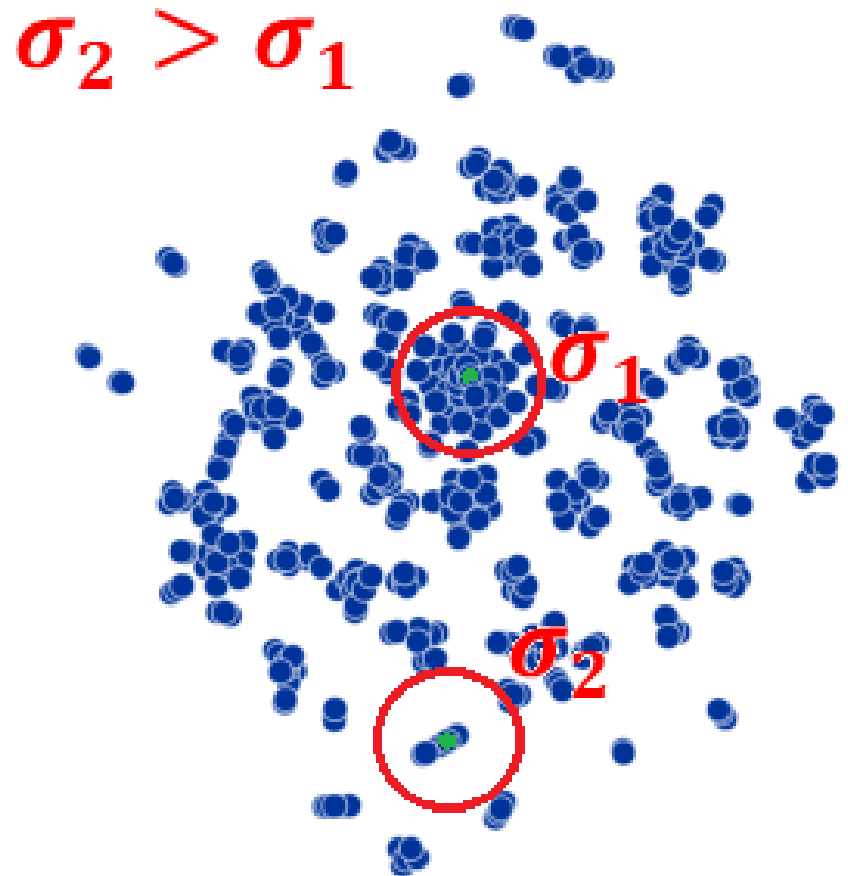
$$p(i|j) = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_j^2)}{\sum_{k \neq j} \exp(-\|x_k - x_j\|^2 / 2\sigma_j^2)}$$

(затем симметризуем  $p(i|j)$ )

- объекты из окрестности  $x_j$

приближаются нормальным  
распределением

- чем кучнее объекты  
из этой окрестности,  
тем меньше берётся  
значение  $\sigma_j^2$



# БЛИЗОСТЬ ОБЪЕКТОВ В НОВОМ ПРОСТРАНСТВЕ

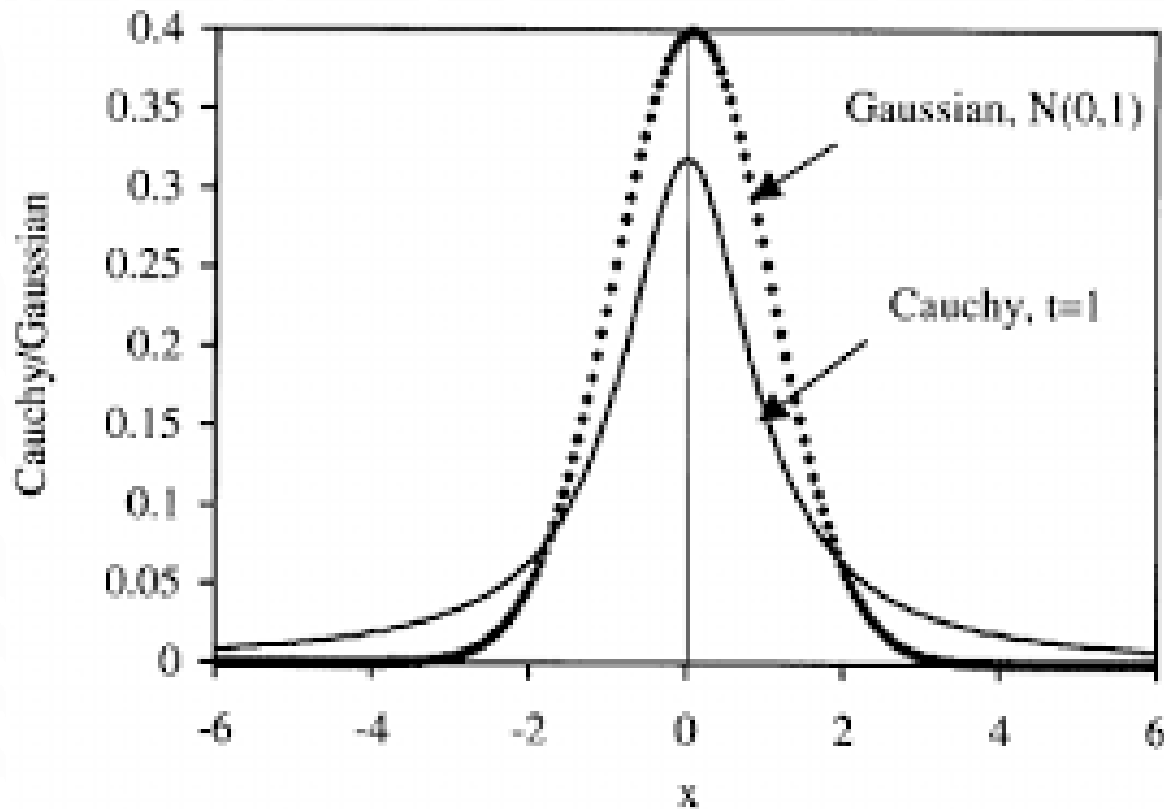
- *В пространстве большой размерности можно разместить несколько объектов так, чтобы расстояния между ними были малы, но сохранить это свойство в низкоразмерном пространстве довольно сложно.*
- Будем измерять сходство объектов в новом пространстве с помощью распределения Коши, так как оно не так сильно штрафует за увеличение расстояний между объектами:

$$q_{ij} = \frac{\left(1 + \left\|z_i - z_j\right\|^2\right)^{-1}}{\sum_{k \neq j} \left(1 + \left\|z_k - z_j\right\|^2\right)^{-1}}$$



# НОРМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ И РАСПРЕДЕЛЕНИЕ КОШИ

- Будем измерять сходство объектов в новом пространстве с помощью распределения Коши, так как оно не так сильно штрафует за увеличение расстояний между объектами:



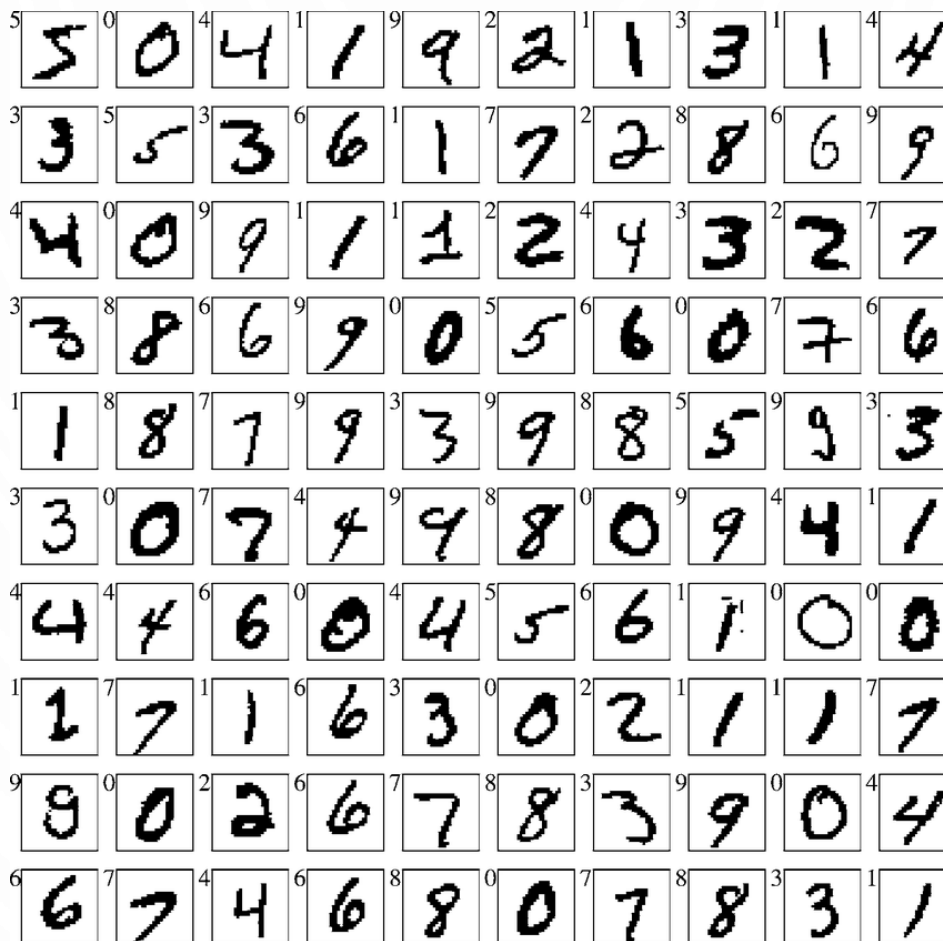
# ОБУЧЕНИЕ TSNE

- Для построения проекций  $z_i$  объектов  $x_i$  будем минимизировать расстояние между исходным и полученным распределениями (минимизируем дивергенцию Кульбака-Лейблера).

$$KL(p||q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \rightarrow \min_{z_1, \dots, z_l}$$

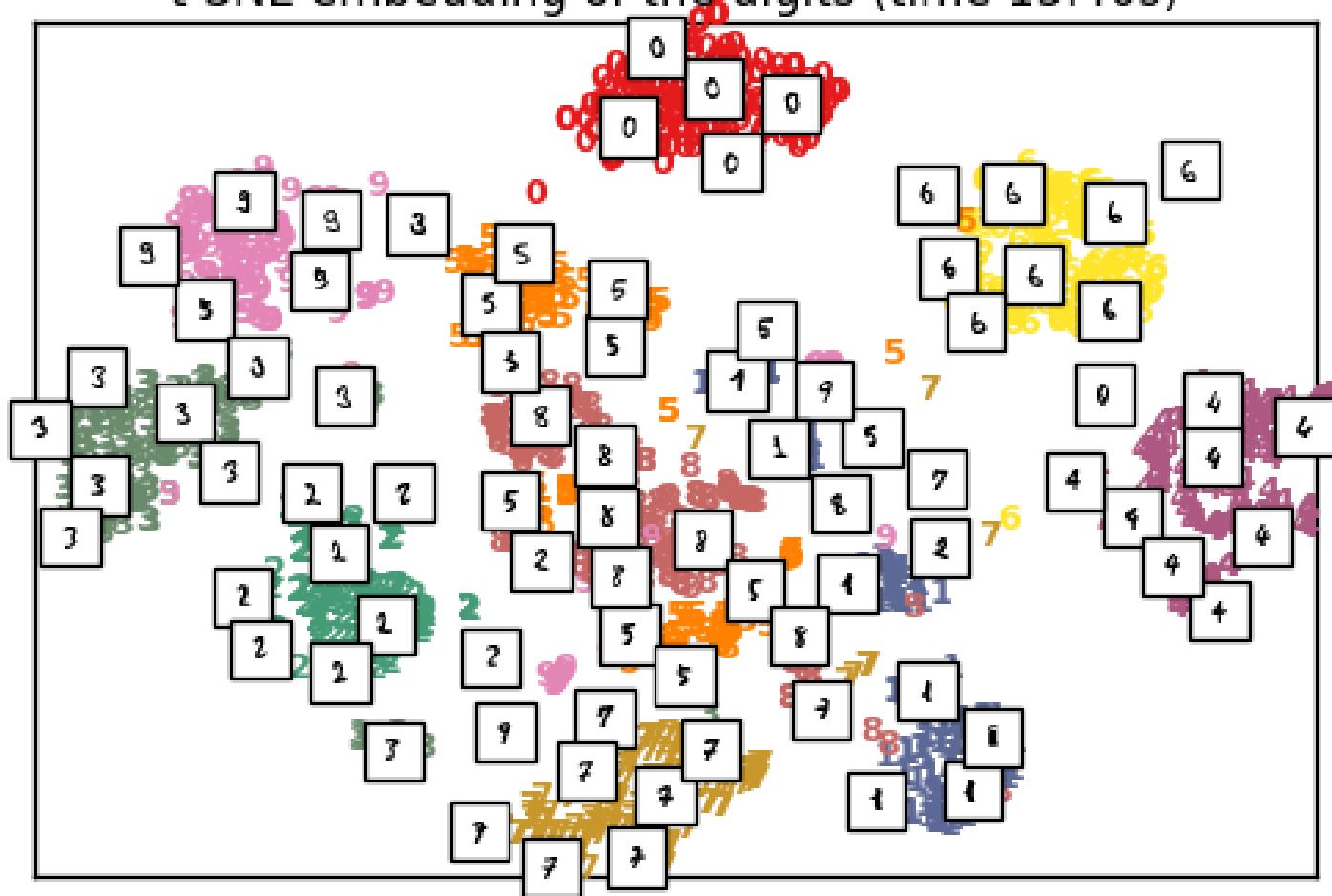
# TSNE (ПРИМЕР)

- MNIST – датасет из различных написаний десятичных цифр, где каждая картинка размера 28x28.



# TSNE (ПРИМЕР)

t-SNE embedding of the digits (time 13.40s)



# ВИЗУАЛИЗАЦИЯ PCA И TSNE

**<http://projector.tensorflow.org/>**

The background features a light gray pattern of concentric circles. In the four corners, there are decorative circuit-like lines in dark blue and light teal, with small circles at the end of the lines.

# ВНЕШНИЕ МЕТРИКИ КАЧЕСТВА КЛАСТЕРИЗАЦИИ

# ADJUSTED RAND INDEX (ARI)

- Предполагается, что известны истинные метки объектов.

*Мера зависит не от самих значений меток, а от разбиения выборки на кластеры.*

- $a$  – число пар объектов с одинаковыми метками и находящихся в одном кластере,  $b$  – число пар объектов с различными метками, но находящихся в одном кластере,  $N$  – число объектов в выборке

$$RI = \frac{2(a + b)}{N(N - 1)}$$

$RI$  – доля объектов, для которых исходное и полученное разбиения согласованы. Выражает похожесть двух различных разбиений выборки.

# ADJUSTED RAND INDEX (ARI)

Затем  $RI$  нормируется так, чтобы величина всегда принимала значения из отрезка  $[-1; 1]$  независимо от числа объектов  $N$  и числа кластеров, получается  $ARI$ .

- $ARI > 0$  – разбиения похожи ( $ARI = 1$  – совпадают)
- $ARI \approx 0$  – случайные разбиения
- $ARI < 0$  – непохожие разбиения



# ADJUSTED MUTUAL INFORMATION (AMI)

Метрика похожа на ARI.

Индекс  $MI$  – это взаимная информация

([https://en.wikipedia.org/wiki/Mutual\\_information](https://en.wikipedia.org/wiki/Mutual_information))

для двух разбиений выборки на кластеры.

- Взаимная информация измеряет долю информации, общей для обоих разбиений: насколько информация об одном из них уменьшает неопределенность относительно другого.
- $AMI \in [0; 1]$  - чем ближе к 1, тем более похожи разбиения.

# ГОМОГЕННОСТЬ, ПОЛНОТА, V-МЕРА

Пусть  $H$  – энтропия распределения распределения. Тогда

$$h = 1 - \frac{H(C|K)}{H(C)}, c = 1 - \frac{H(K|C)}{H(K)},$$

где  $K$  – результат кластеризации,  $C$  – истинное разбиение выборки на классы.

- $h$  (гомогенность) измеряет, насколько каждый кластер состоит из объектов одного класса
- $c$  (полнота) измеряет, насколько объекты одного класса относятся к одному кластеру

# ГОМОГЕННОСТЬ, ПОЛНОТА, V-МЕРА

- Гомогенность и полнота принимают значения из отрезка  $[0; 1]$ . Большие значения соответствуют более точной кластеризации.

*Эти метрики не нормализованы (как ARI и AMI), т.е. они зависят от числа кластеров!*

- *При большом числе кластеров и малом числе объектов лучше использовать ARI и AMI*
- *При более 1000 объектов и числе кластеров меньше 10 проблема не так сильно выражена, поэтому её можно игнорировать.*

# ГОМОГЕННОСТЬ, ПОЛНОТА, V-МЕРА

V-мера – учитывает и гомогенность и полноту, это их среднее гармоническое:

$$v = \frac{2hc}{h + c}$$

*V-мера показывает, насколько два разбиения схожи между собой.*

# СИЛУЭТ (SILHOUETTE)

*Не требует знания истинных меток! (значит, это внутренняя метрика качества кластеризации)*

- Пусть  $a$  – среднее расстояние от объекта до всех объектов из того же кластера,  $b$  – среднее расстояние от объекта до объектов из ближайшего (не содержащего объект) кластера. Тогда *силуэт данного объекта*:

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

- *Силуэт выборки ( $S$ ) – средняя величина силуэта по объектам.*

*Силуэт показывает, насколько среднее расстояние до объектов своего кластера отличается от среднего расстояния до объектов других кластеров.*

# СИЛУЭТ (SILHOUETTE)

$$S \in [-1; 1].$$

- $S$  близкий к -1 – плохие (разрозненные) кластеризации
- $S \approx 0$  – кластеры накладываются друг на друга
- $S$  близкий к 1 – четко выраженные кластеры

*С помощью силуэта можно выбирать число кластеров  $k$  (если оно заранее неизвестно) – выбирается  $k$ , для которого метрика максимальна.*

- Силуэт зависит от формы кластеров и достигает больших значений на более выпуклых кластерах.

# ПОИСК АНОМАЛИЙ

# ISOLATION FOREST

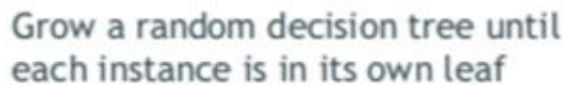
- Строим лес, состоящий из  $N$  деревьев. Каждый признак и порог выбираем случайно. Останавливаемся, когда в вершине 1 объект или когда построили дерево максимальной глубины.

Идея: чем сильнее объект отличается от большинства, тем раньше он будет отделен от основной выборки случайными разбиениями => выбросы – объекты, которые оказались на небольшой глубине.



# ISOLATION FOREST

Идея: чем сильнее объект отличается от большинства, тем раньше он будет отделен от основной выборки случайными разбиениями => выбросы – объекты, которые оказались на небольшой глубине.



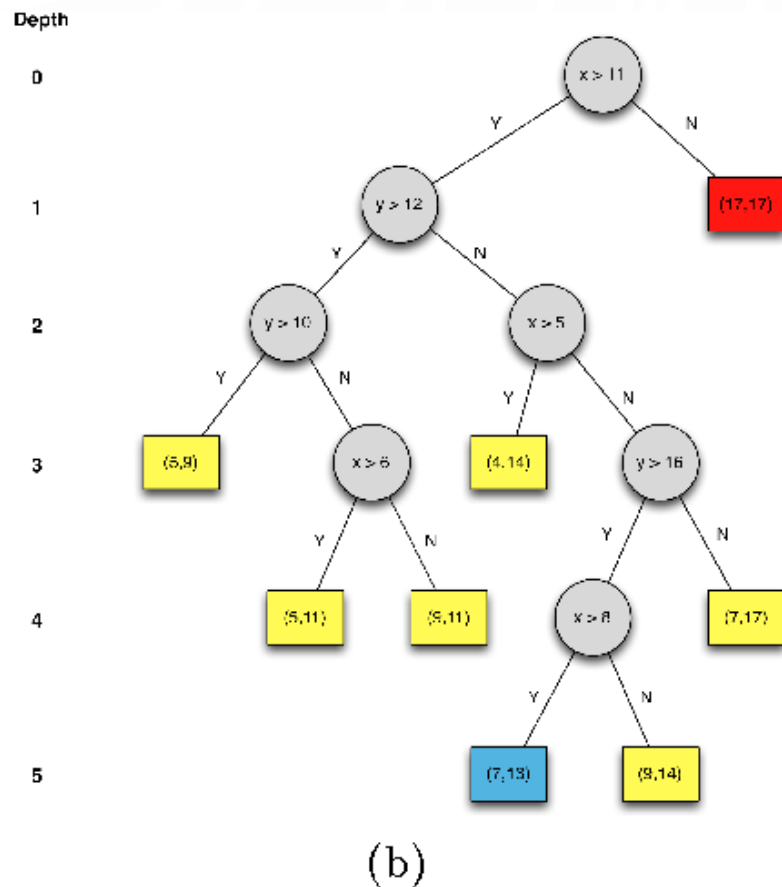
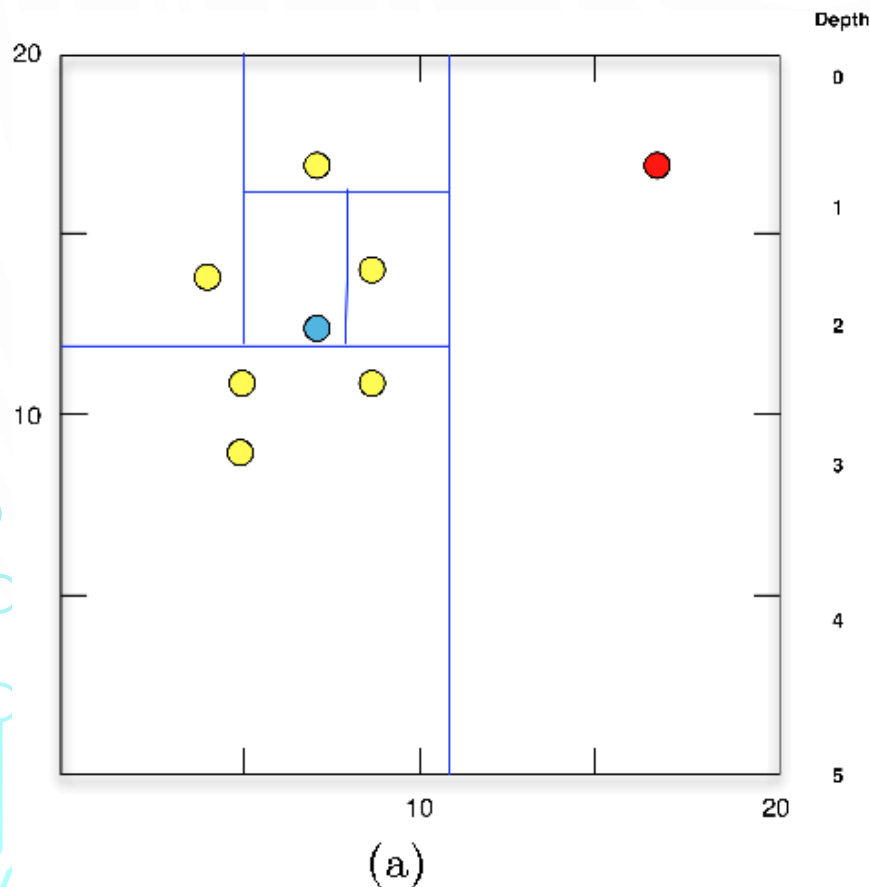
Depth

“hard” to isolate

Now repeat the process several times and use average Depth to compute anomaly score: 0 (similar) -> 1 (dissimilar)

# ISOLATION FOREST

Идея: чем сильнее объект отличается от большинства, тем раньше он будет отделен от основной выборки случайными разбиениями => выбросы – объекты, которые оказались на небольшой глубине.



# ISOLATION FOREST

- Если объект единственный в листе, то его оценка аномальности в дереве – это глубина листа  $h_n(x) = k$ .
- Если в лист попало  $m$  объектов, то оценка аномальности:

$$h_n(x) = k + c(m),$$

где  $c(m)$  - средняя длина пути от корня до листа.

- Оценка аномальности объекта в Isolation Forest:

$$a(x) = 2^{-\frac{a}{b}},$$

где  $a = \frac{1}{N} \sum h_n(x)$  – средняя глубина,

$b = c(l)$  – средняя длина пути в дереве, построенном по выборке размера  $l$ .

