

# Машинное обучение

## Лекция 4. Методы кластеризации

Катя Тузова

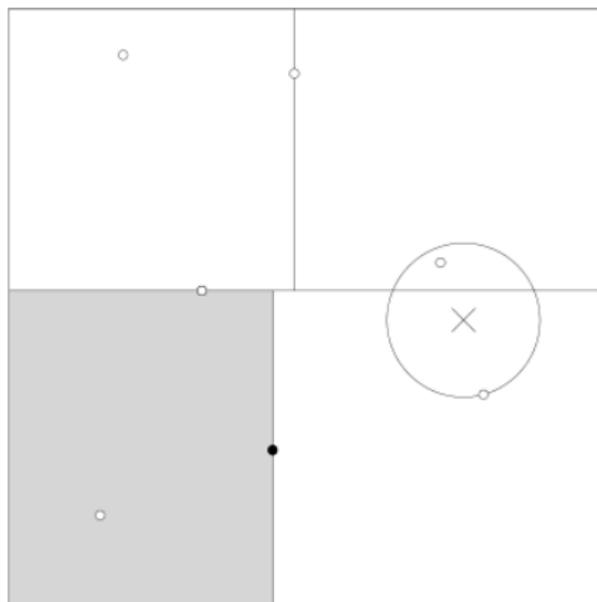
Какая в худшем случае сложность поиска в kd-дереве из  $n$  элементов?  
Какой случай является “худшим”?

# Разбор летучки

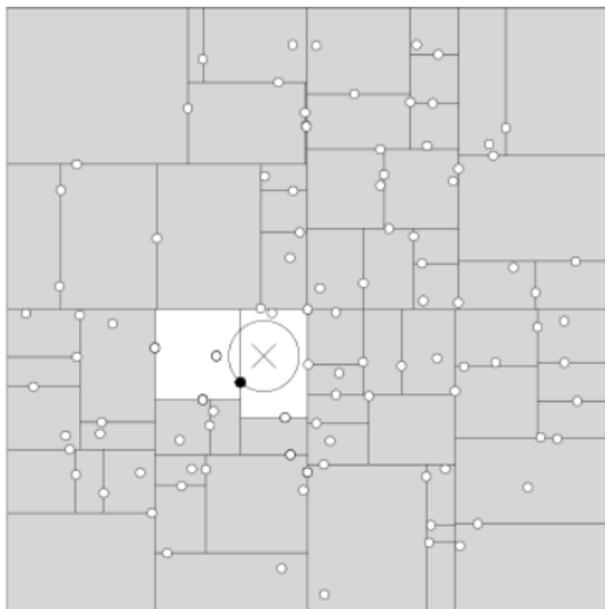
$O(n)$

На каждом шаге сфера радиуса  $R$  пересекает разделяющую поверхность.

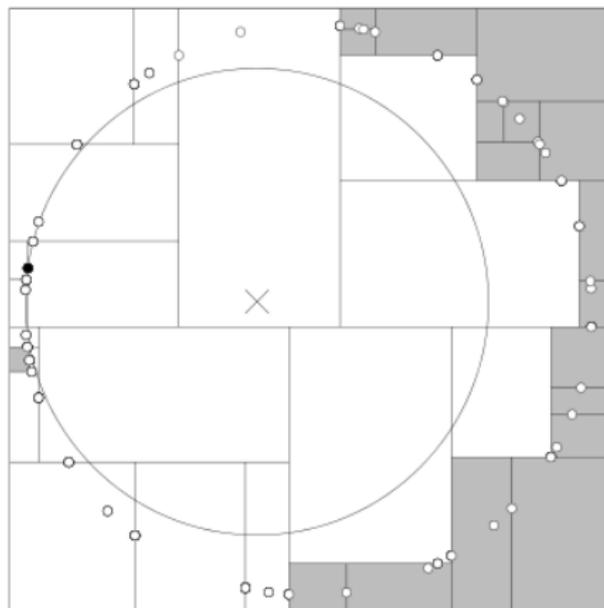
# Разбор летучки



# Разбор летучки



# Разбор летучки



Что важнее для алгоритма классификации: точность или полнота? Почему?

$$\text{Precision} = \frac{TP}{TP + FP}$$

- $TP$  — это количество элементов, которые классификатор верно отнёс к классу  $c$ ,
- $FP$  — количество элементов, которые классификатор неверно отнёс к классу  $c$ ,

$$\text{Precision} = \frac{TP}{TP + FP}$$

Характеризует, сколько полученных от классификатора положительных ответов являются правильными. Чем больше точность, тем меньше число ложных попаданий.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Что остается неизвестным?

$$\text{Precision} = \frac{TP}{TP + FP}$$

Мера точности не дает представления о том, все ли правильные ответы вернул классификатор.

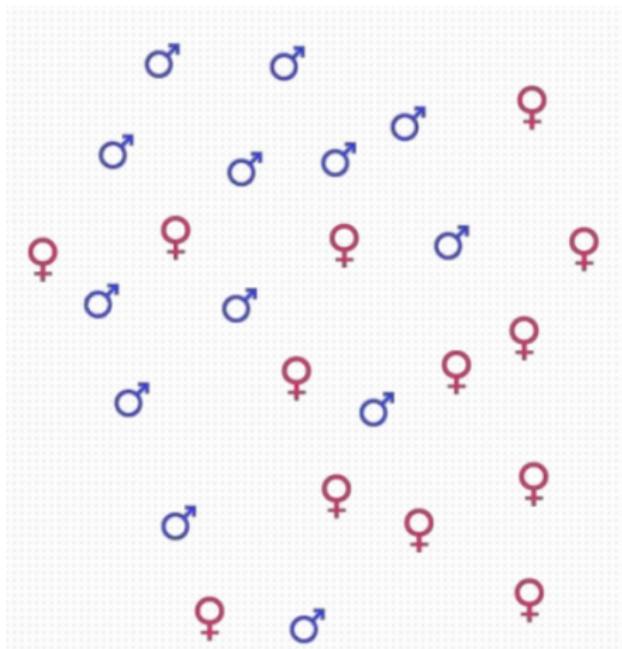
$$\text{Recall} = \frac{TP}{TP + FN}$$

- $TP$  — это количество элементов, которые классификатор верно отнёс к классу  $c$ ,
- $FN$  — количество элементов, которые классификатор неверно отнёс к классу, отличному от  $c$ .

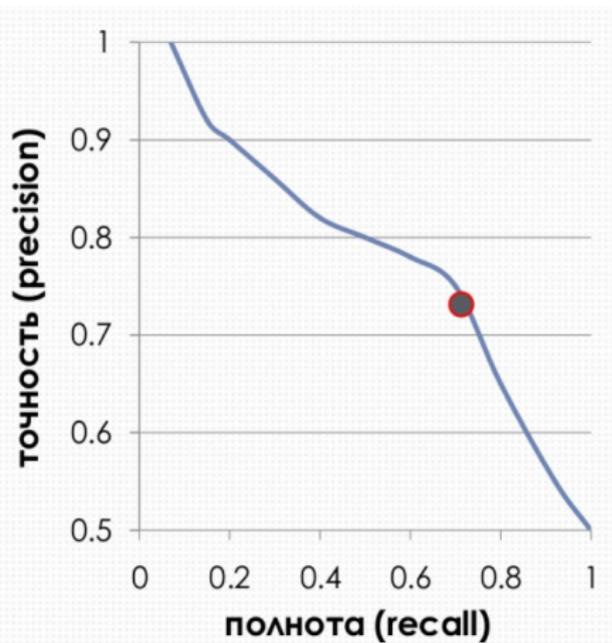
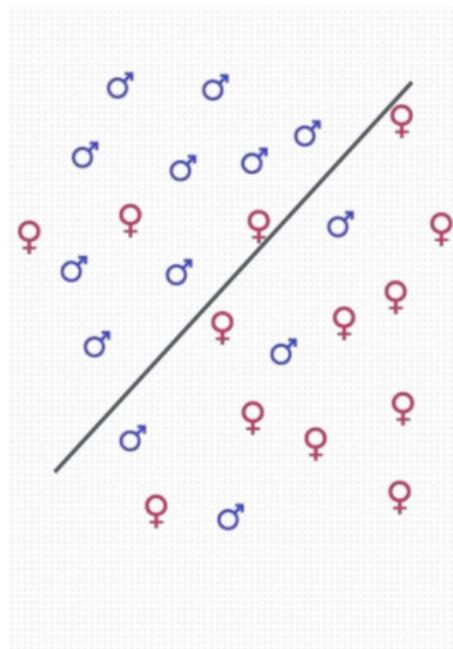
$$\text{Recall} = \frac{TP}{TP + FN}$$

Мера полноты характеризует способность классификатора «угадывать» как можно большее число положительных ответов из ожидаемых.

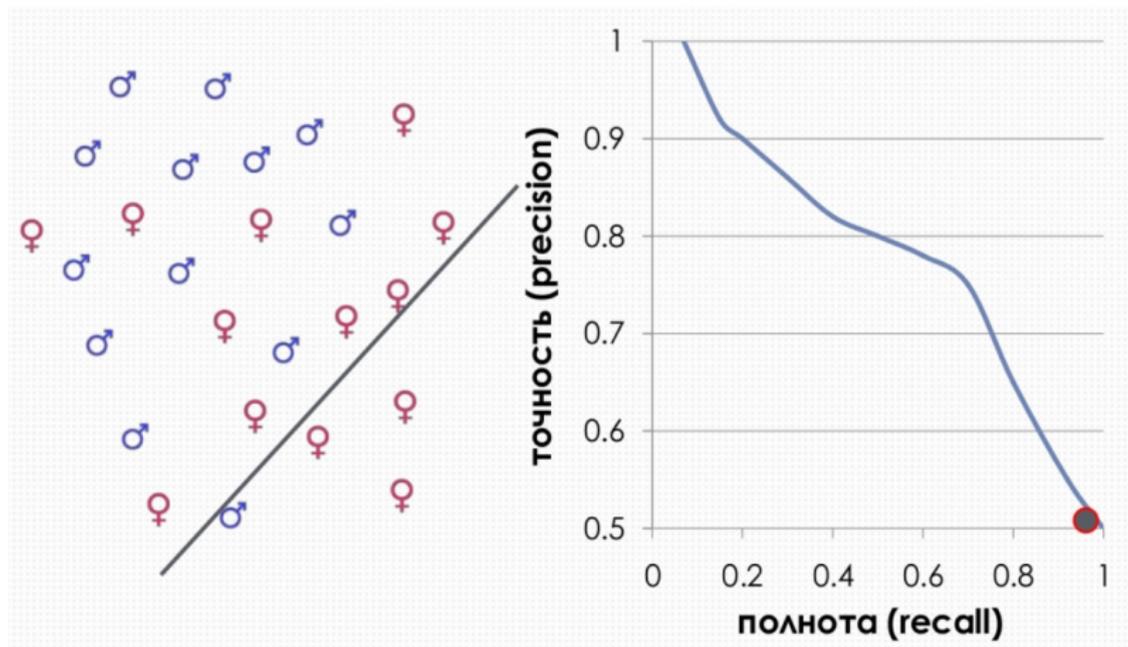
# Разбор летучки



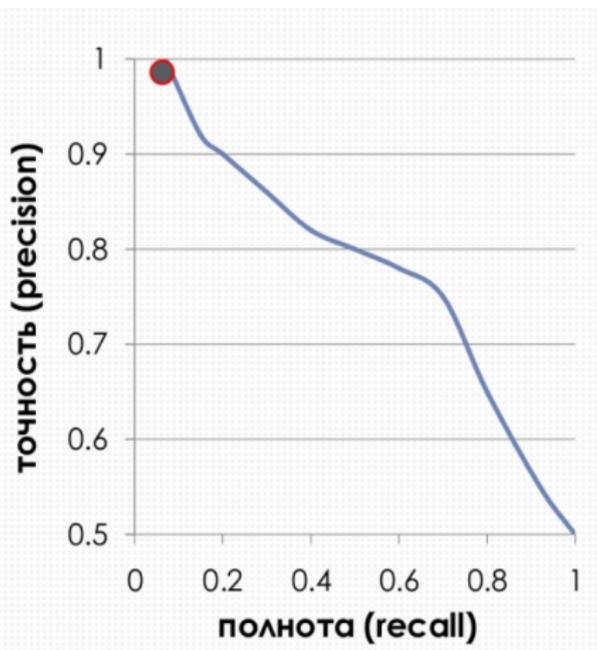
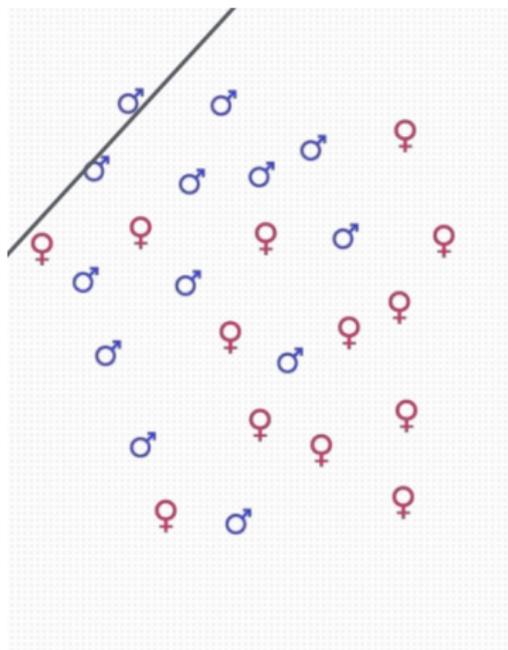
# Разбор летучки



# Разбор летучки



# Разбор летучки



# Разбор летучки

$$F_1 = 2 \frac{PR}{P+R}$$

F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю.

Кратко опишите основную идею locality-sensitive hashing.

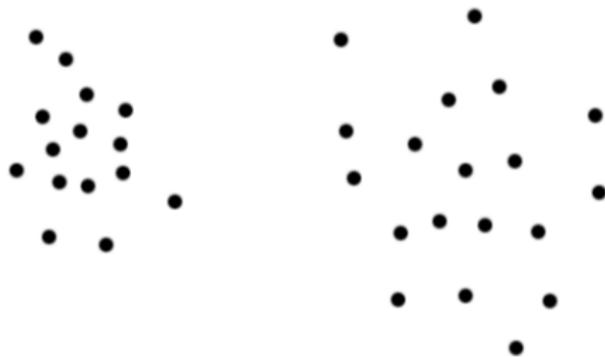
Найти такую хэш-функцию  $h$ , что:

- Если объекты  $C_1$  и  $C_2$  похожи, то с большой вероятностью  $h(C_1) == h(C_2)$
- Иначе – с большой вероятностью  $h(C_1) \neq h(C_2)$

Приведите пример кластерной структуры на которой алгоритм Ланса-Уильямса с параметрами, соответствующими групповому среднему, будет работать плохо.

# Разбор летучки

Сгущения



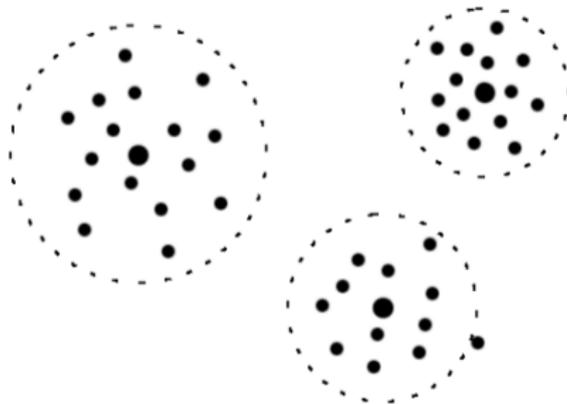
# Разбор летучки

Ленты



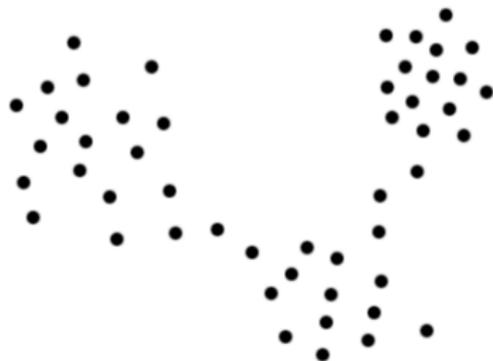
# Разбор летучки

С центром



# Разбор летучки

С перемычками



# Разбор летучки

На фоне



# Разбор летучки

Перекрывающиеся



Соответствует ли дендрограмма кластеризации точек  $a$ ,  $b$ ,  $c$ ,  $d$  алгоритмом Ланса-Уильямса с параметрами, соответствующими групповому среднему?  
Если нет, то приведите вариант, который считаете правильным.

# Дендрограмма

Может ли так случиться, что дендрограмма имеет самопересечения?

# Свойство монотонности

Кластеризация монотонна, если на каждом шаге расстояние  $\rho$  между объединяемыми кластерами не уменьшается.

$$\rho_2 \leq \rho_3 \leq \dots \leq \rho_l$$

# Постановка задачи кластеризации

Кластеризация – задача разделения объектов одной природы на несколько групп так, чтобы объекты в одной группе обладали одним и тем же свойством.

Кластеризация – это обучение без учителя.

# Постановка задачи кластеризации

$X$  – пространство объектов

$\rho : X \times X \rightarrow [0, \infty)$  – функция расстояния между объектами

Найти:

$Y$  – множество кластеров

$a : X \rightarrow Y$  – алгоритм кластеризации

# Степени свободы в постановке задачи

- Критерий качества кластеризации
- Число кластеров неизвестно заранее
- Результат кластеризации существенно зависит от метрики

# Цели кластеризации

- Сократить объём хранимых данных
- Выделить нетипичные объекты
- Упростить дальнейшую обработку данных
- Построить иерархию множества объектов

# Оценка качества кластеризации

- Минимизировать среднее внутрикластерное расстояние

$$\frac{\sum_{a(x_i)=a(x_j)} \rho(x_i, x_j)}{\sum_{a(x_i)=a(x_j)} 1} \rightarrow \min$$

- Максимизировать среднее межкластерное расстояние

$$\frac{\sum_{a(x_i) \neq a(x_j)} \rho(x_i, x_j)}{\sum_{a(x_i) \neq a(x_j)} 1} \rightarrow \max$$

# Методы кластеризации

- Иерархические
- Графовые
- Статистические

Какие есть две очевидные идеи?

Очевидные:

- Выделение связных компонент
- Минимальное покрывающее дерево

# Выделение связанных компонент

- Рисуем полный граф с весами, равными расстоянию между объектами
- Выбираем лимит расстояния  $r$  и выкидываем все ребра длиннее  $r$
- Компоненты связности полученного графа – наши кластеры

# Выделение связанных компонент

Как искать компоненты связности?

# Минимальное покрывающее дерево

Минимальное остовное дерево – дерево, содержащее все вершины графа и имеющее минимальный суммарный вес ребер.

Как найти?

# Минимальное покрывающее дерево

Как использовать минимальное остовное дерево для разбиения на кластеры?

# Минимальное покрывающее дерево

Строим минимальное остовное дерево, а потом выкидываем из него ребра максимального веса.

Сколько ребер выбросим – столько кластеров получим.

# Статистические алгоритмы

# Алгоритм FOREL

Идея:

- Выделить все точки выборки  $x_i$ , попадающие внутрь сферы  $\rho(x_i, x_0) \leq R$
- Перенести  $x_0$  в центр тяжести выделенных точек
- Повторять пока  $x_0$  не стабилизируется

# Алгоритм FOREL

Input:  $X, R$

$U = X, C = \emptyset$

while  $U \neq \emptyset$ :

    выбрать случайную точку  $x_0$

    Повторять пока  $x_0$  не стабилизируется:

$$c = \{x \in X \mid \rho(x, x_0) < R\}$$

$$x_0 = \frac{1}{|c|} \sum_{x \in c} x$$

$$U = U \setminus c, C = C \cup \{c\}$$

# Алгоритм FOREL

- + Наглядность
- + Сходимость
- Зависимость от выбора  $x_0$
- Плохо работает, если изначальная выборка плохо делится на кластеры

# Метод $k$ -средних

Идея:

минимизировать меру ошибки

$$E(X, C) = \sum_{i=1}^n \|x_i - \mu_i\|^2$$

$\mu_i$  – ближайший к  $x_i$  центр кластера

# Метод $k$ -средних

Инициализировать центры  $k$  кластеров

Пока  $c_i$  не перестанет меняться:

$$c_i = \arg \min_{c \in C} \rho(x_i, \mu_c) \quad i = 1, \dots, l$$

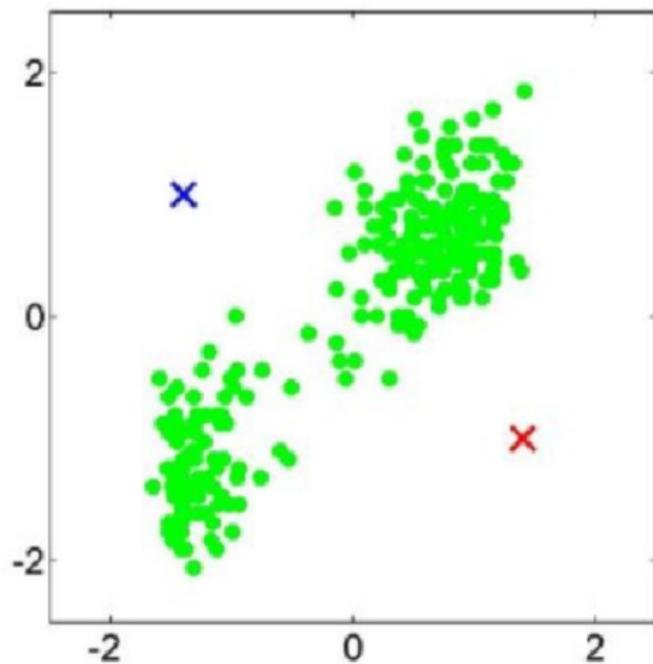
$$\mu_c = \frac{\sum_{c_i=c} f_j(x_i)}{\sum_{c_i=c} 1} \quad j = 1, \dots, n, c \in C$$

$\mu_c$  – новое положение центров кластеров

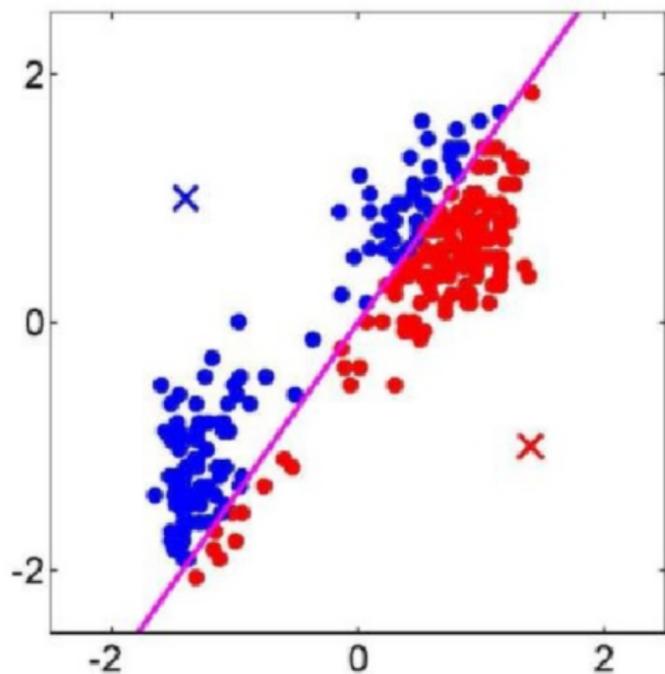
$c_i$  – принадлежность  $x_i$  к кластеру

$\rho(x_i, \mu_c)$  – расстояние от  $x_i$  до центра кластера  $\mu_c$

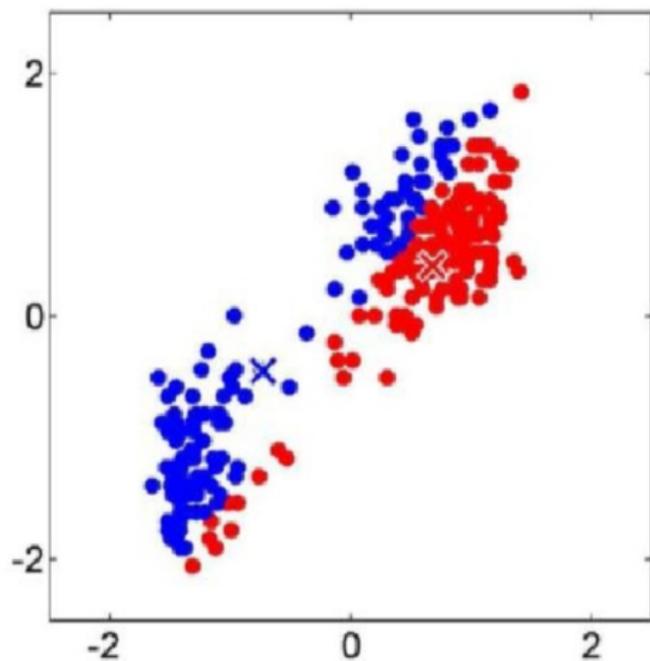
# Метод $k$ -средних



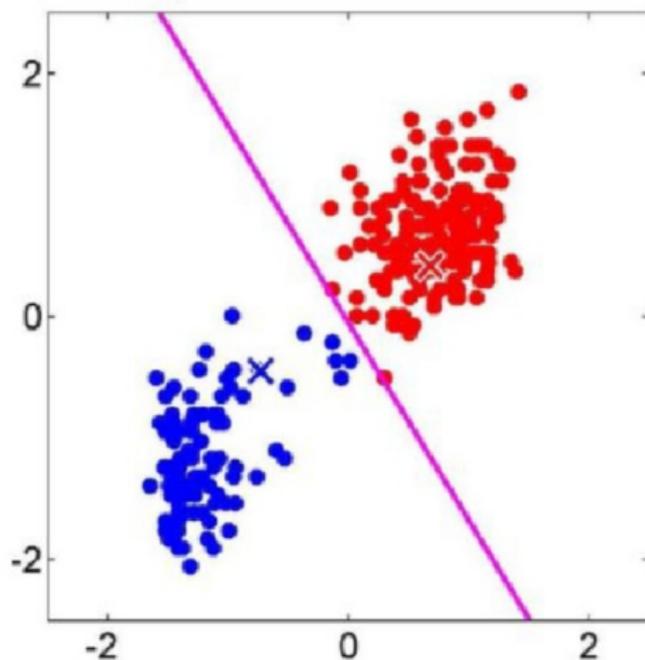
# Метод $k$ -средних



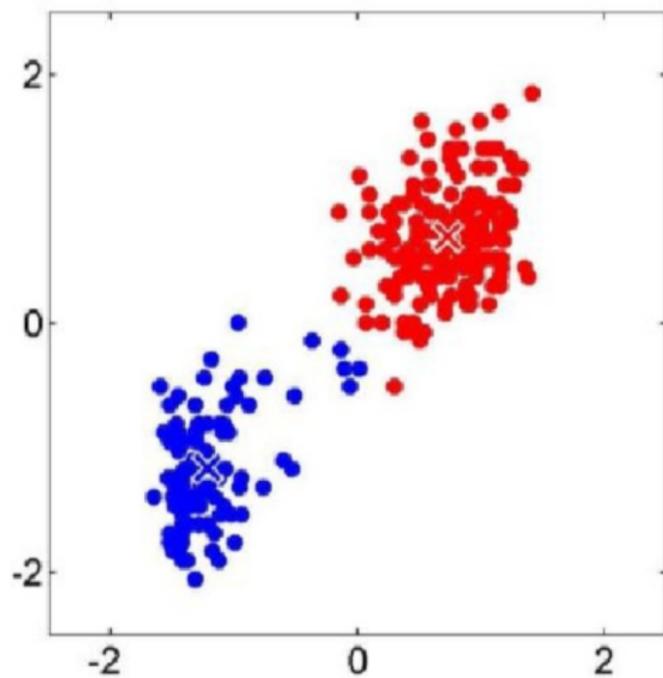
# Метод $k$ -средних



# Метод $k$ -средних



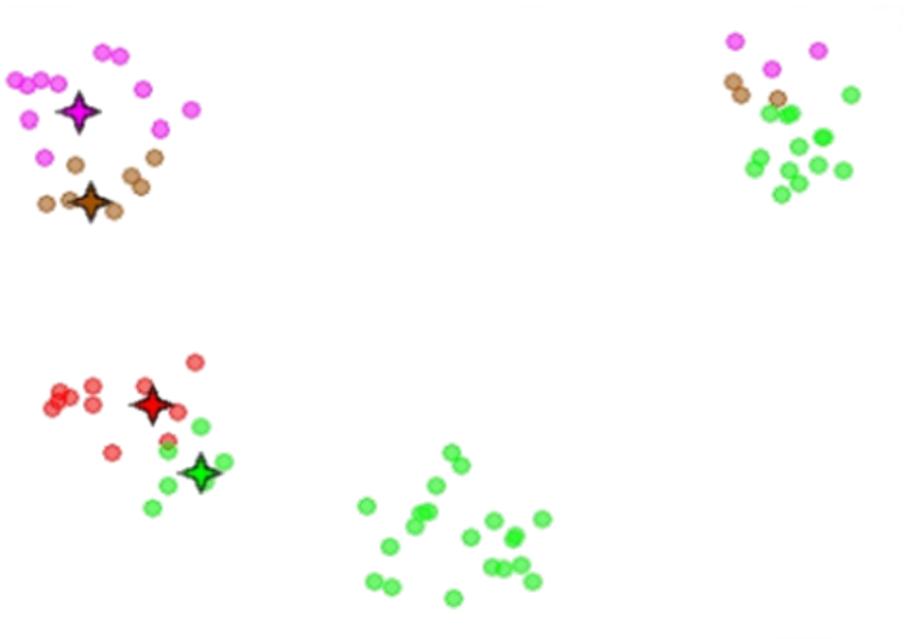
# Метод $k$ -средних



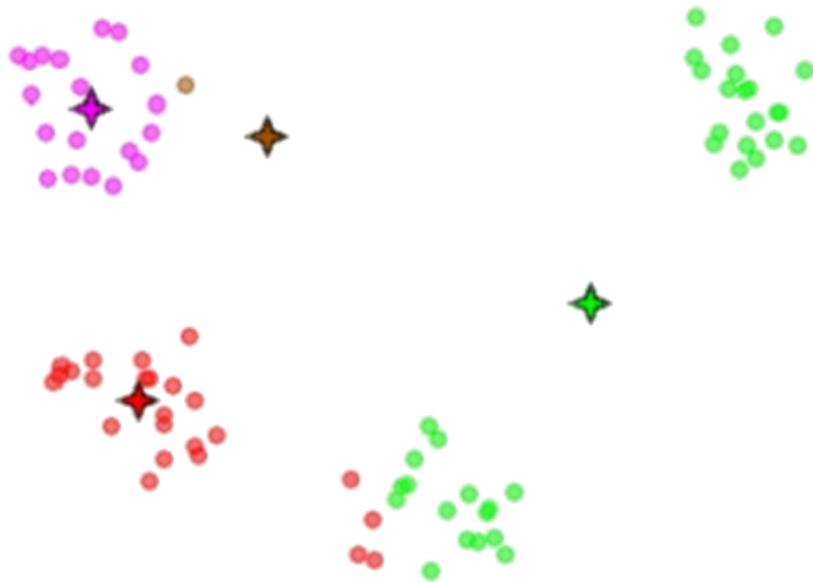
# Особенности метода $k$ -средних

- Чувствительность к начальному выбору  $\mu_c$
- Необходимость задавать  $k$

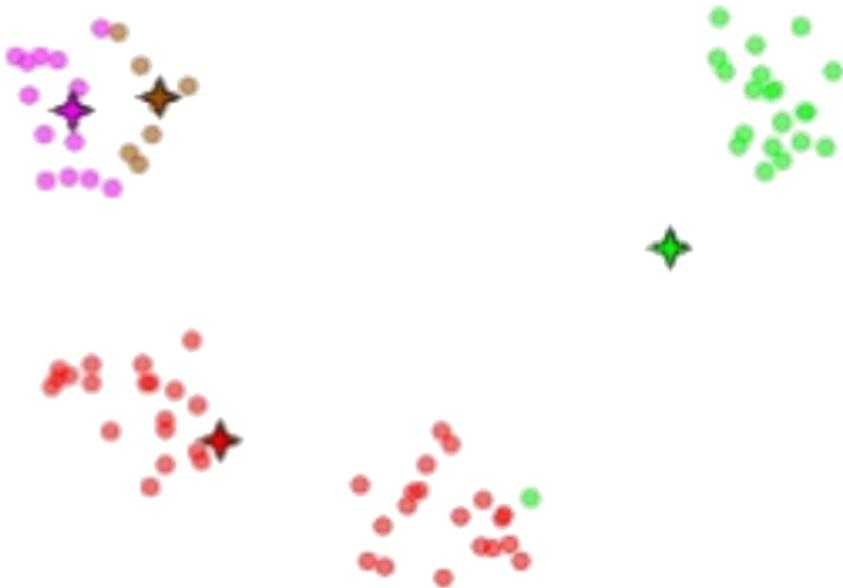
# Чувствительность к начальному выбору $\mu_c$



# Чувствительность к начальному выбору $\mu_c$



# Чувствительность к начальному выбору $\mu_c$



# Чувствительность к начальному выбору $\mu_c$



# Необходимость задавать $k$



# Устранение недостатков

# Устранение недостатков

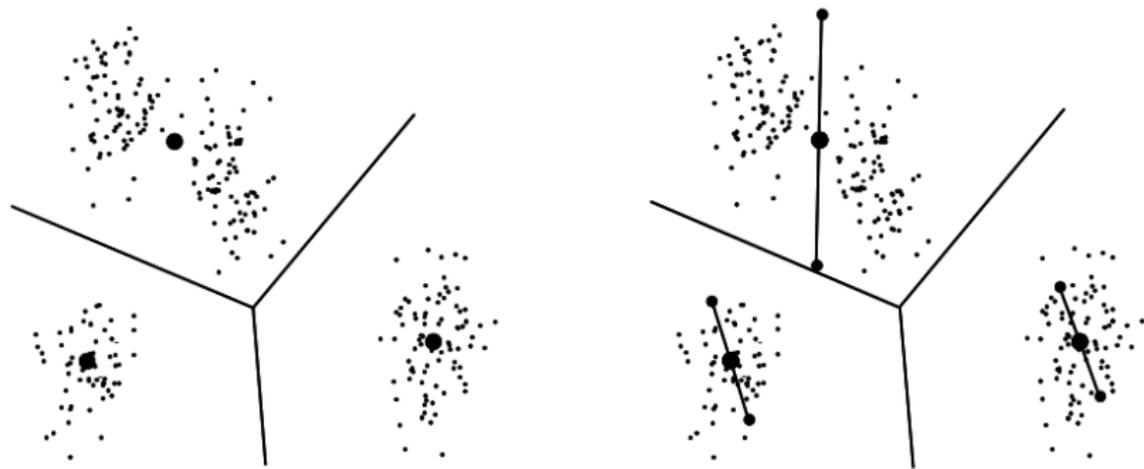
- Несколько случайных кластеризаций
- Постепенное наращивание числа  $k$
- Использование k-means++

- Выбрать первый центроид случайным образом
- Для каждой точки найти значение квадрата расстояния до ближайшего центроида.
- Выбрать из этих точек следующий центроид так, чтобы вероятность выбора точки была пропорциональна вычисленному для неё квадрату расстояния

Идея:

- Получать на вход не  $k$ , а диапазон, в котором может находиться  $k$ .
- Запустить  $k$ -means на самом маленьком значении из диапазона.
- Разбивать пополам полученные кластеры и проверять, не улучшилась ли кластеризация.

# X-means



Как проверить, что кластеризация улучшилась?

# Байесовский информационный критерий

$$BIC = E(X, C) + kd \log(n)$$

$E$  – внутрикластерная сумма квадратов расстояний

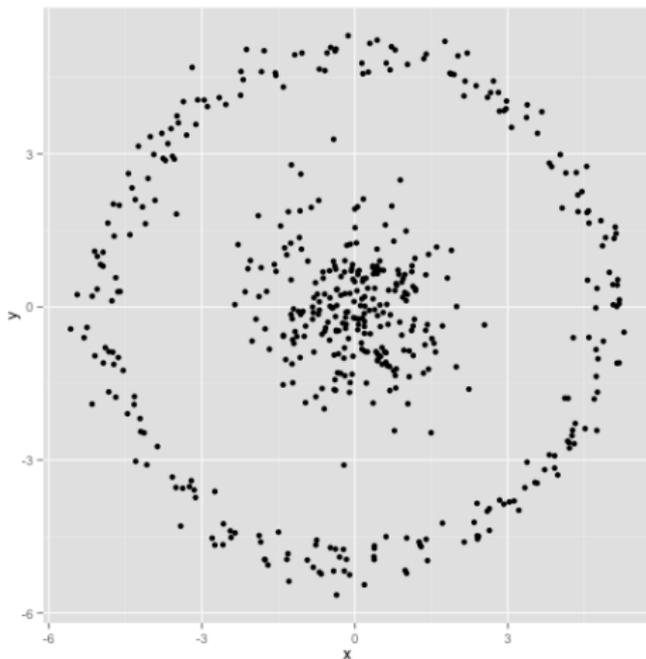
$k$  – количество кластеров

$d$  – длина вектора фич

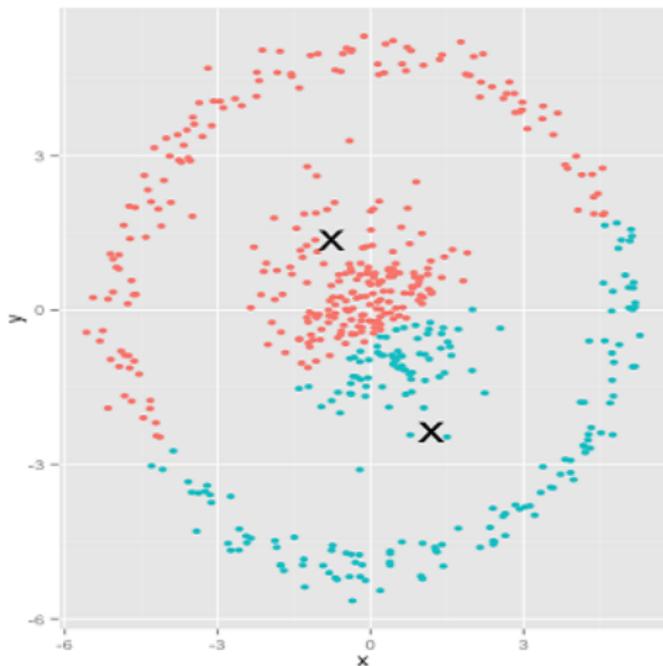
$n$  – количество объектов в выборке

# Недостатки k-means

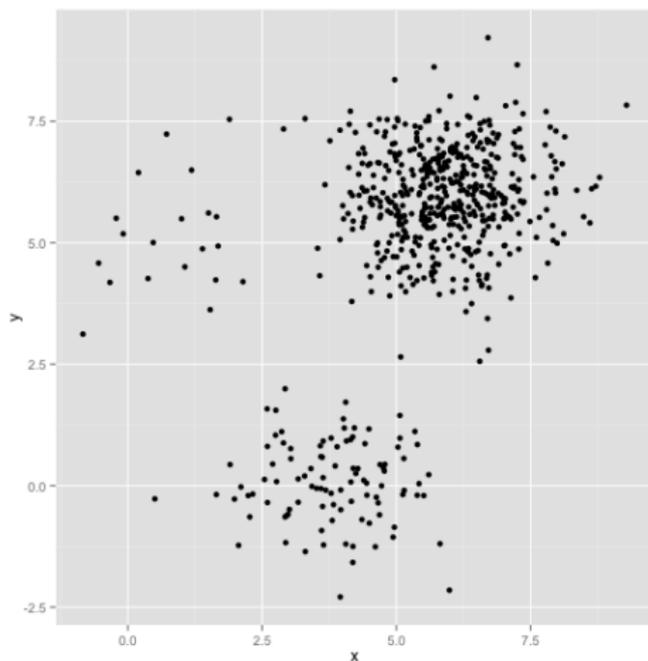
# "Не сферические данные"



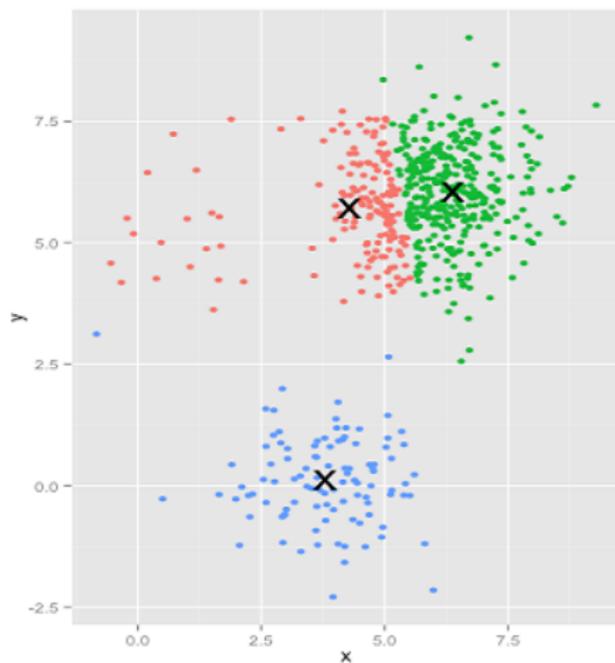
# "Не сферические данные"



# Разноразмерные кластеры



# Разноразмерные кластеры



# На следующей лекции

- Линейные методы классификации
- Минимизация эмпирического риска
- Метод градиентного спуска
- Принцип максимума правдоподобия