Машинное обучение

Лекция 4. Методы кластеризации

Катя Тузова

Дендрограмма

Может ли так случиться, что дендрограмма имеет самопересечения?

Свойство монотонности

Кластеризация монотонна, если на каждом шаге расстояние ρ между объединяемыми кластерами не уменьшается.

$\rho_2 \leq \rho_3 \leq \cdots \leq \rho_l$

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

Найти:
Y – множество кластеров
$a : X \to Y$ – 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

$$\frac{\sum\limits_{a(x_i)=a(x_j)} \rho(x_i,x_j)}{\sum\limits_{a(x_i)=a(x_j)} 1} \to \min$$

- Максимизировать среднее межкластерное расстояние

$$\frac{\sum\limits_{a(x_i)\neq a(x_j)} \rho(x_i,x_j)}{\sum\limits_{a(x_i)\neq a(x_j)} 1} \to \max$$

Методы кластеризации

- Иерархические
- Графовые
- Статистические

Графовые алгоритмы

Какие есть две очевидные идеи?

Графовые алгоритмы

Очевидные:

- Выделение связных компонент
- Минимальное покрывающее дерево

Выделение связных 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 связных 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

$U = X, C = 0$

while $U \neq 0$:

выбрать случайную точку x_0

Повторять пока x_0 не стабилизируется:

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

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

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

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

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

μ_i – ближайший к x_i центр кластера

Метод k-средних

Инициализировать центры k кластеров

Пока c_i не перестанет меняться:

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

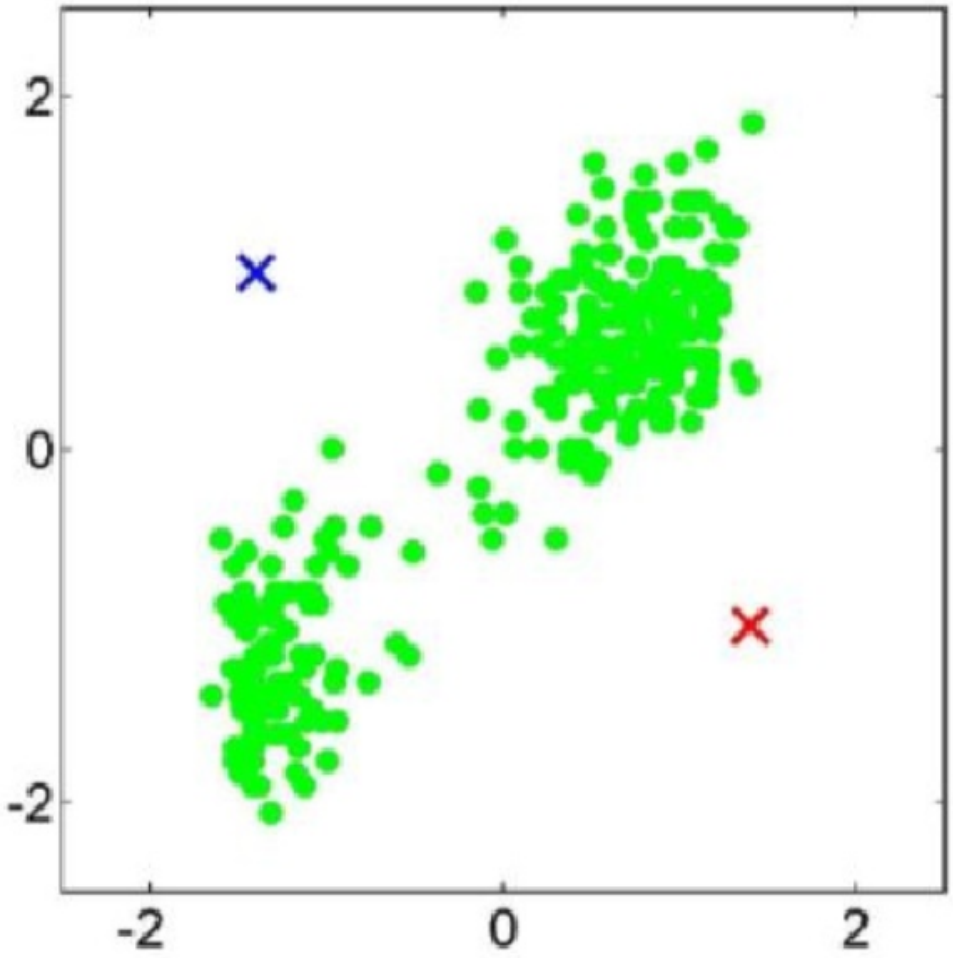
$$\mu_c = \frac{\sum\limits_{c_i = c} f_j(x_i)}{\sum\limits_{c_i = c} 1} \qquad j = 1, \ldots, n,\ c \in C$$

μ_c – новое положение центров кластеров

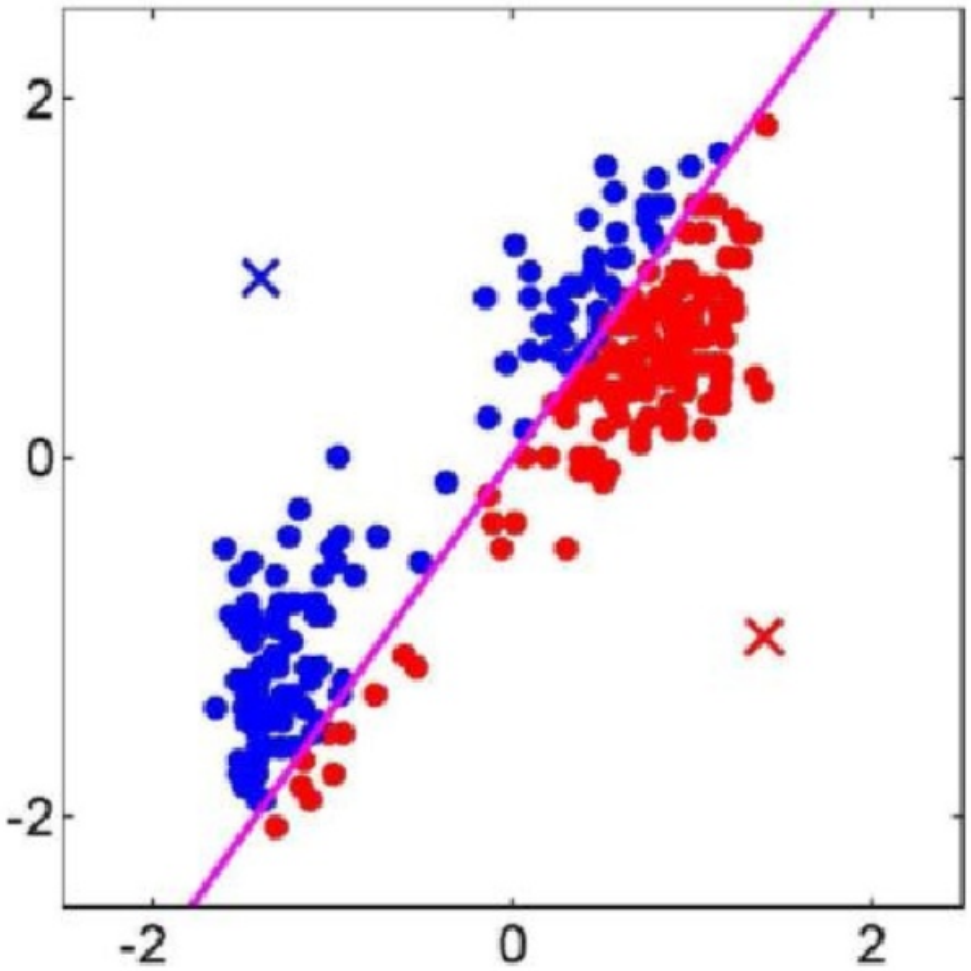
c_i – принадлежность x_i к кластеру

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

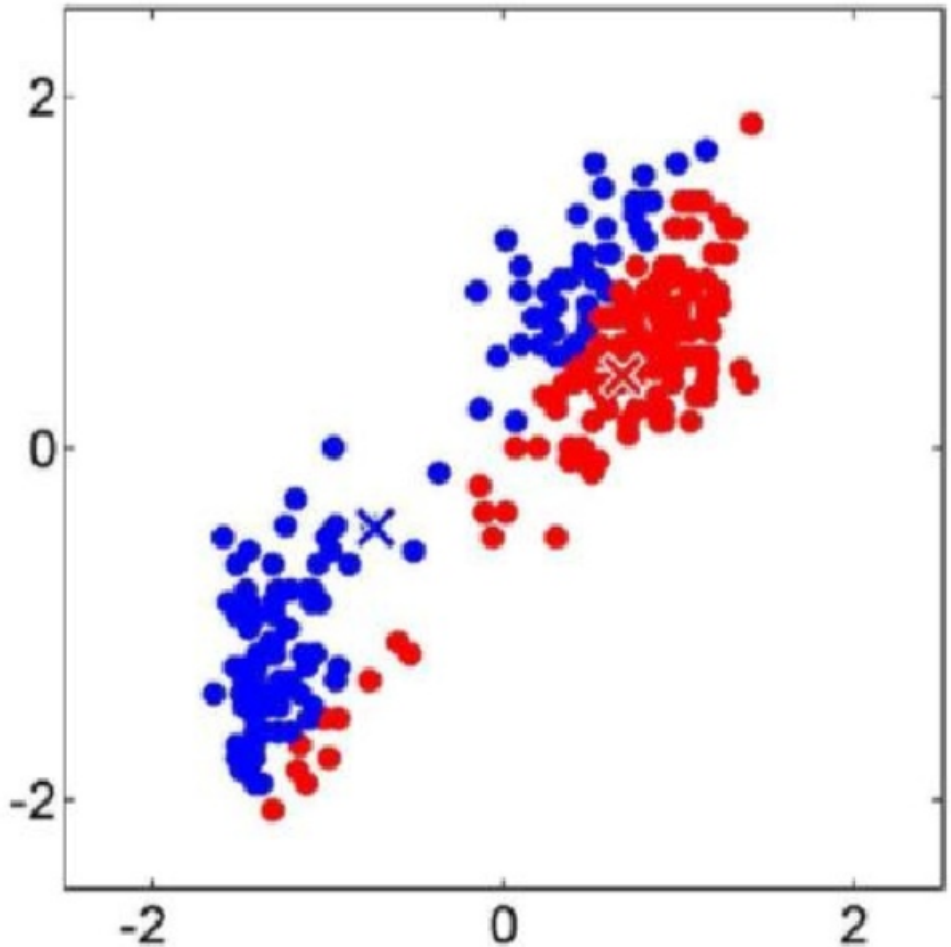
Метод k-средних



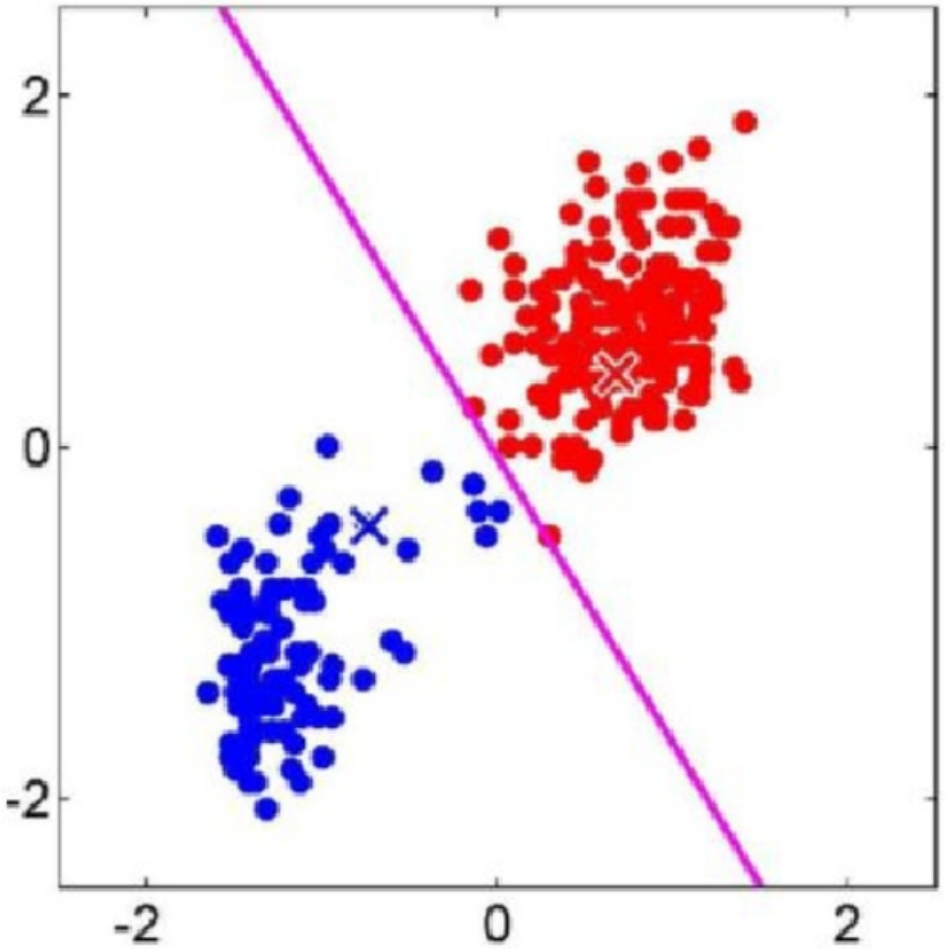
Метод k-средних



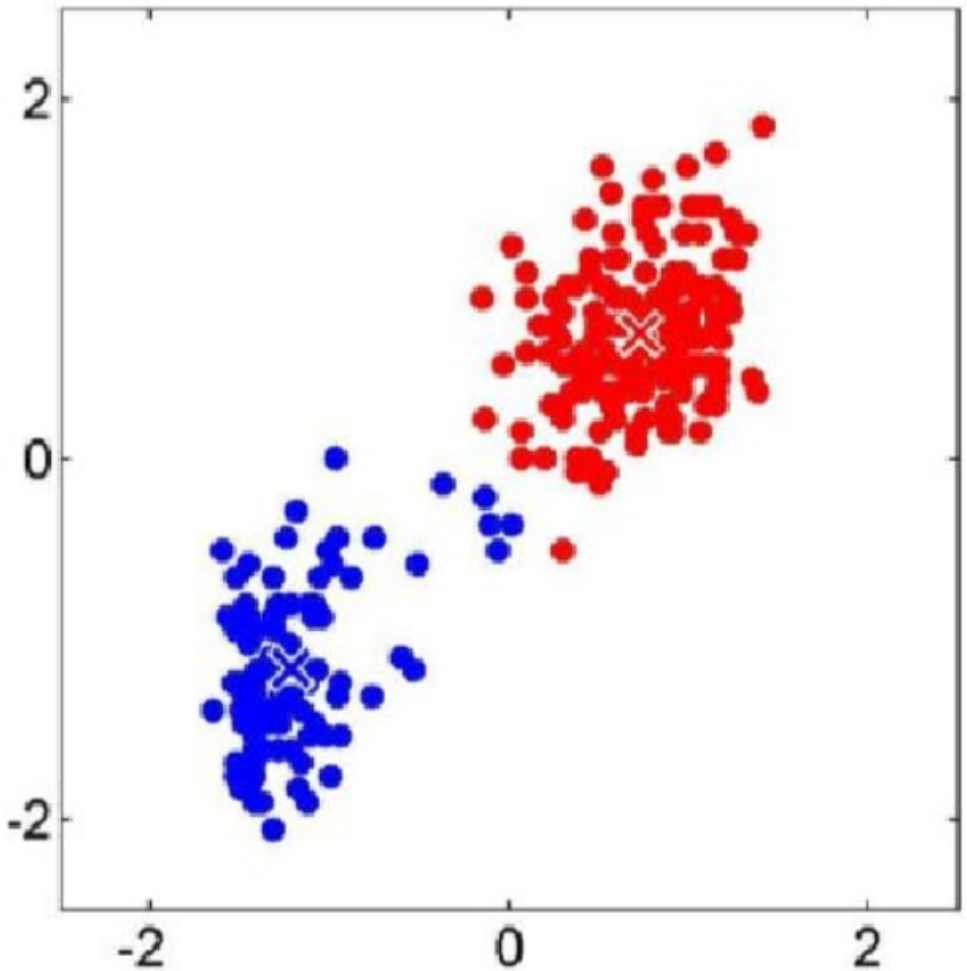
Метод k-средних



Метод k-средних



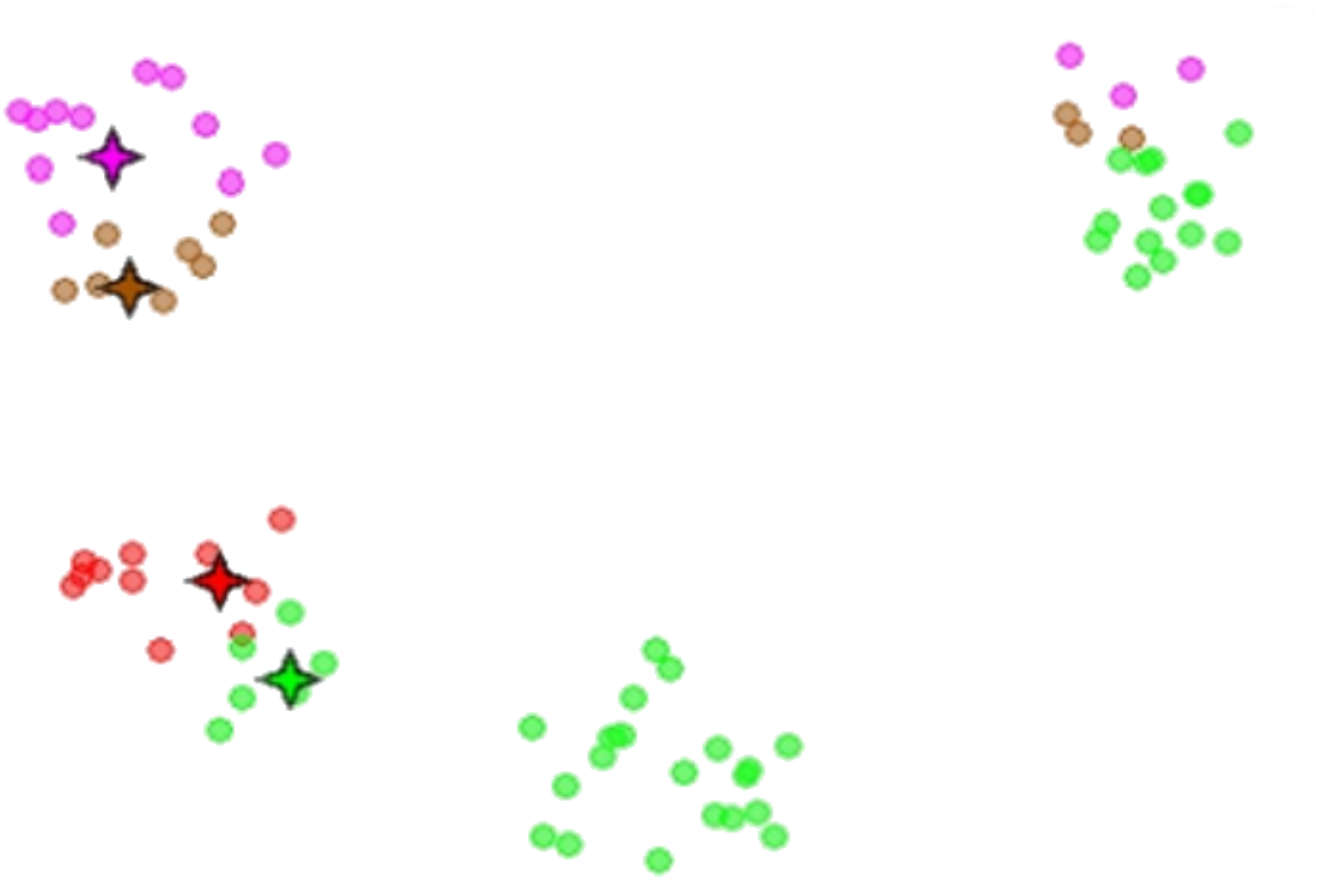
Метод k-средних



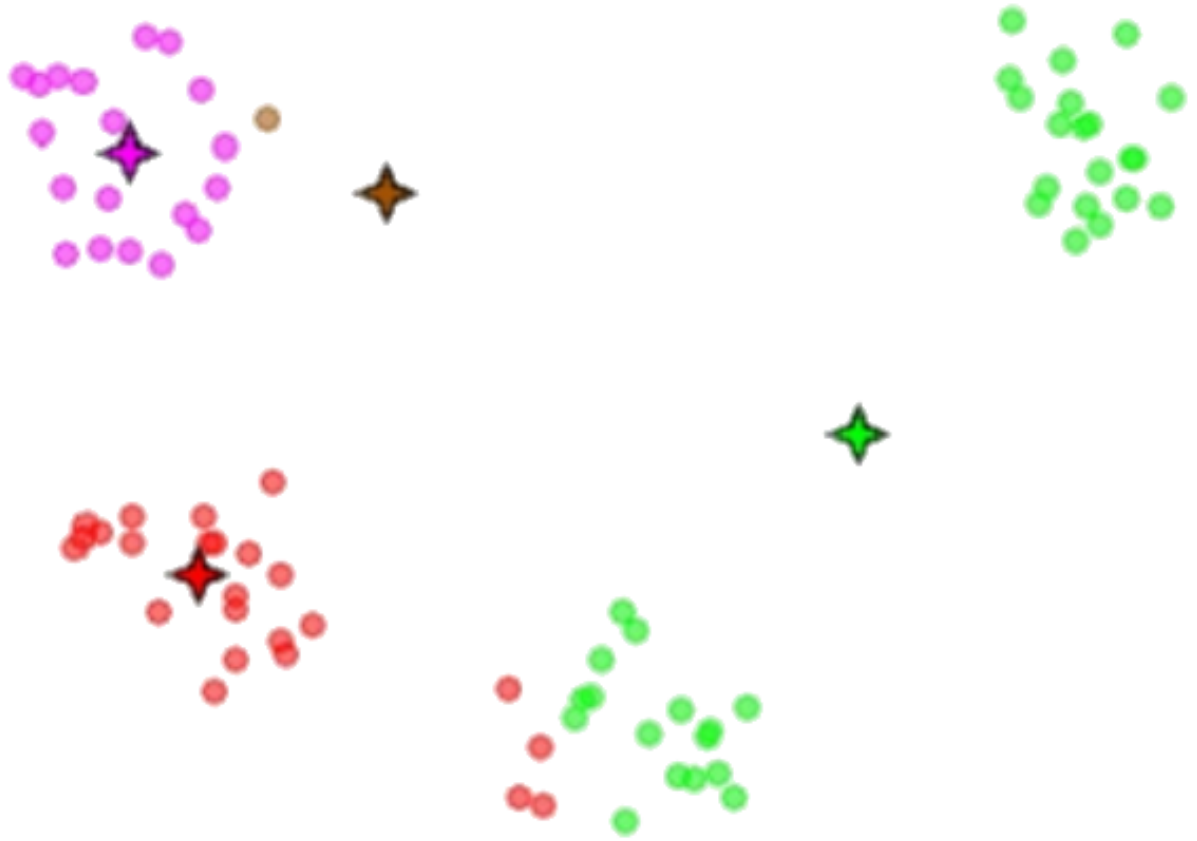
Особенности метода k-средних

- Чувствительность к начальному выбору μ_c
- Необходимость задавать k

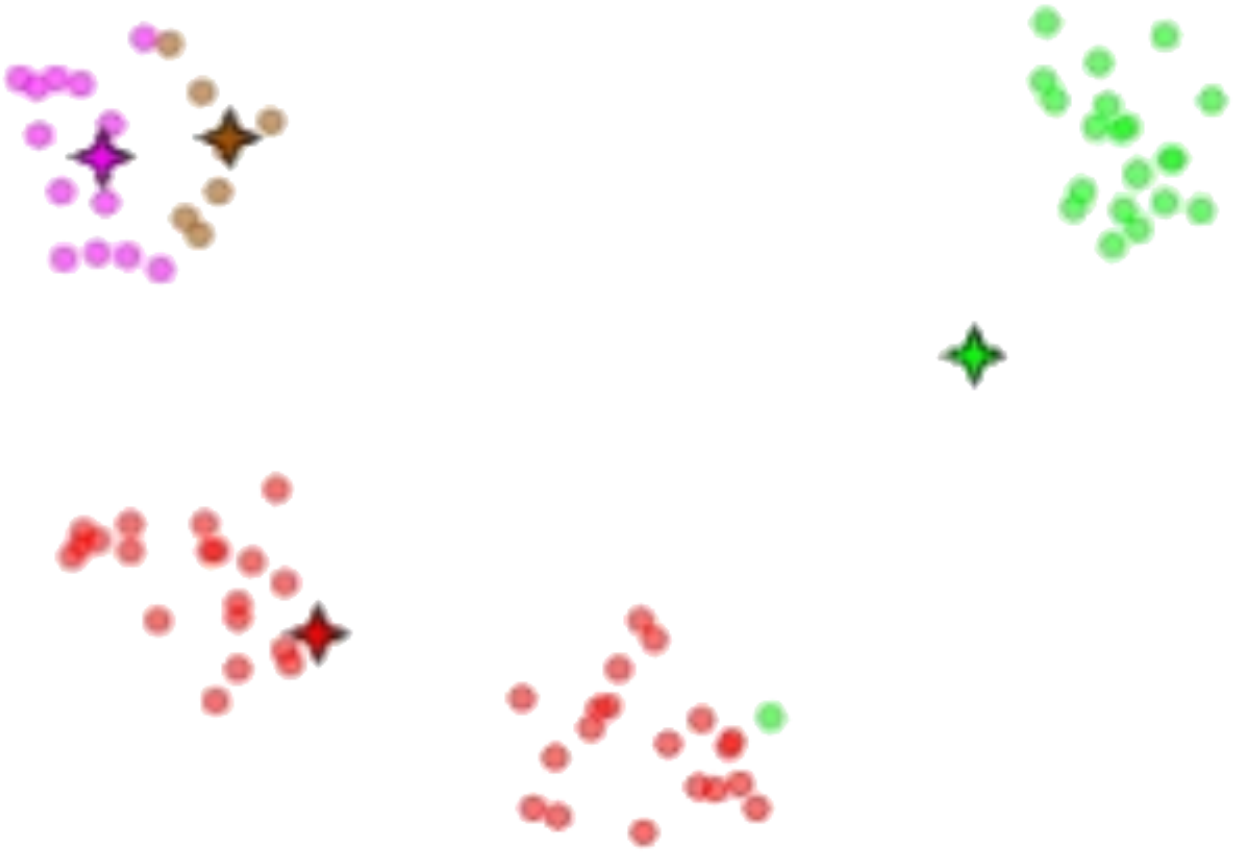
Чувствительность к начальному выбору μ_c



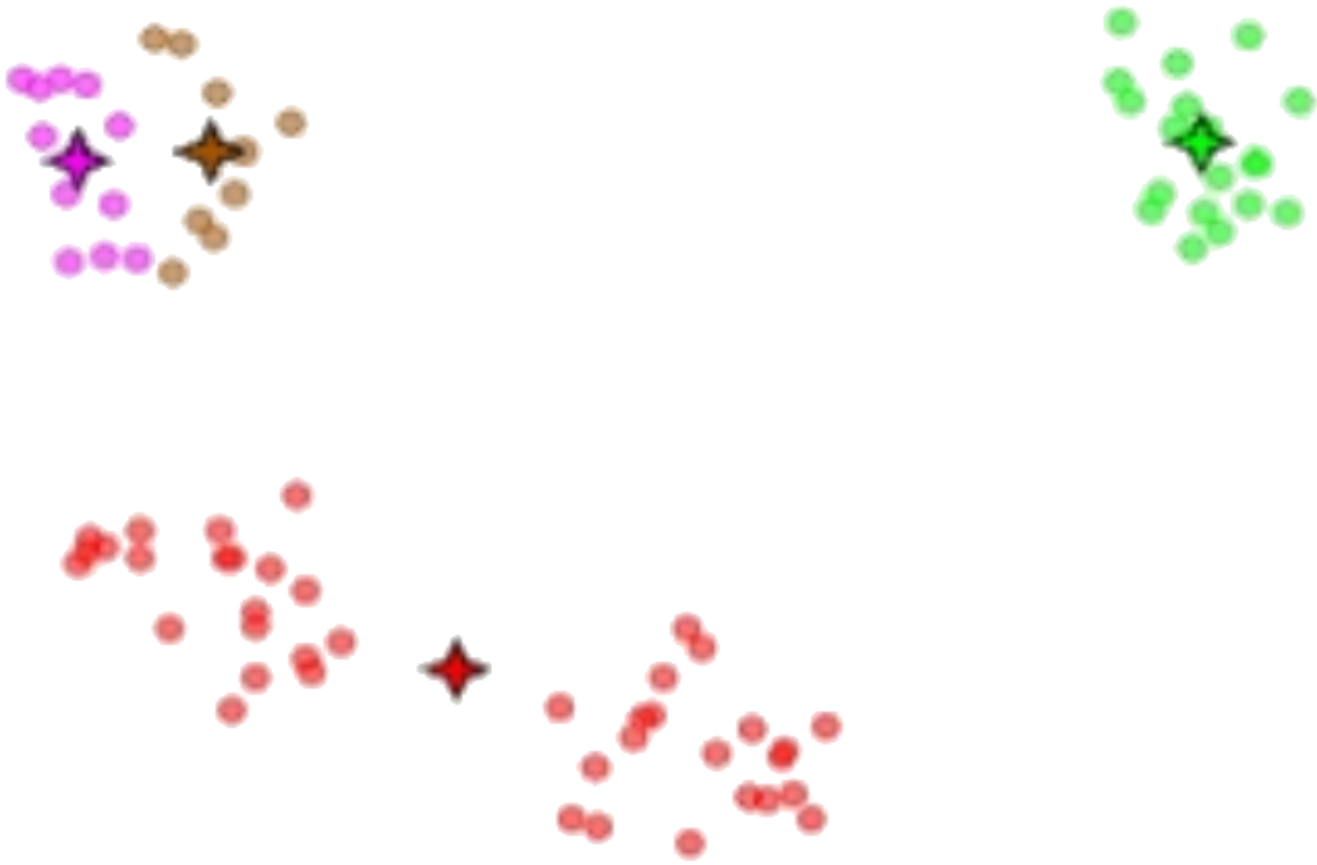
Чувствительность к начальному выбору μ_c



Чувствительность к начальному выбору μ_c



Чувствительность к начальному выбору μ_c



Необходимость задавать k

Устранение недостатков

Устранение недостатков

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

k-means++

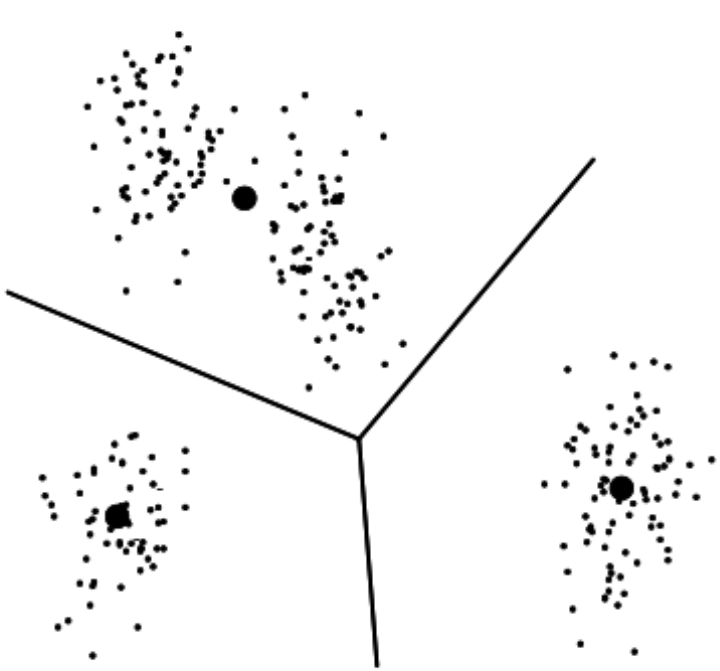
- Выбрать первый центроид случайным образом
- Для каждой точки найти значение квадрата расстояния до ближайшего центроида.
- Выбрать из этих точек следующий центроид так, чтобы вероятность выбора точки была пропорциональна вычисленному для неё квадрату расстояния

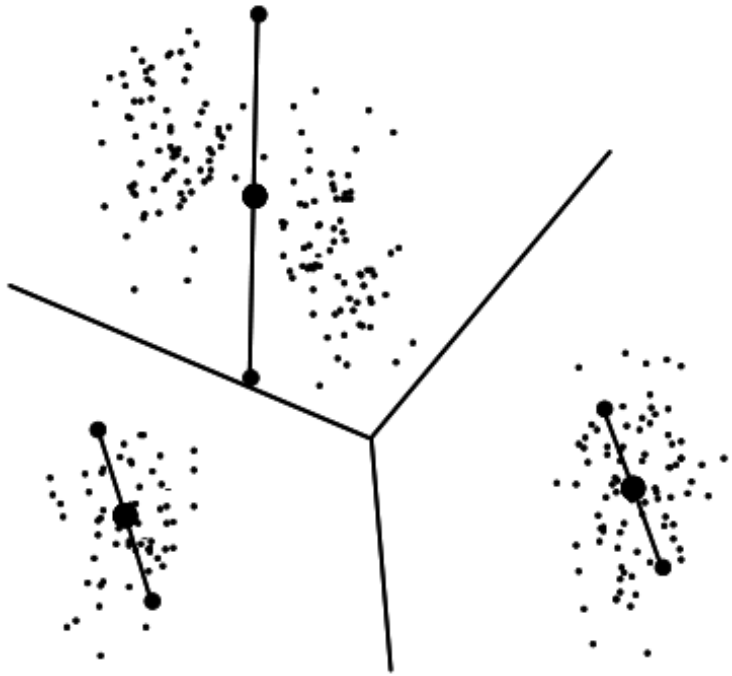
X-means

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





X-means

Как проверить, что кластеризация улучшилась?

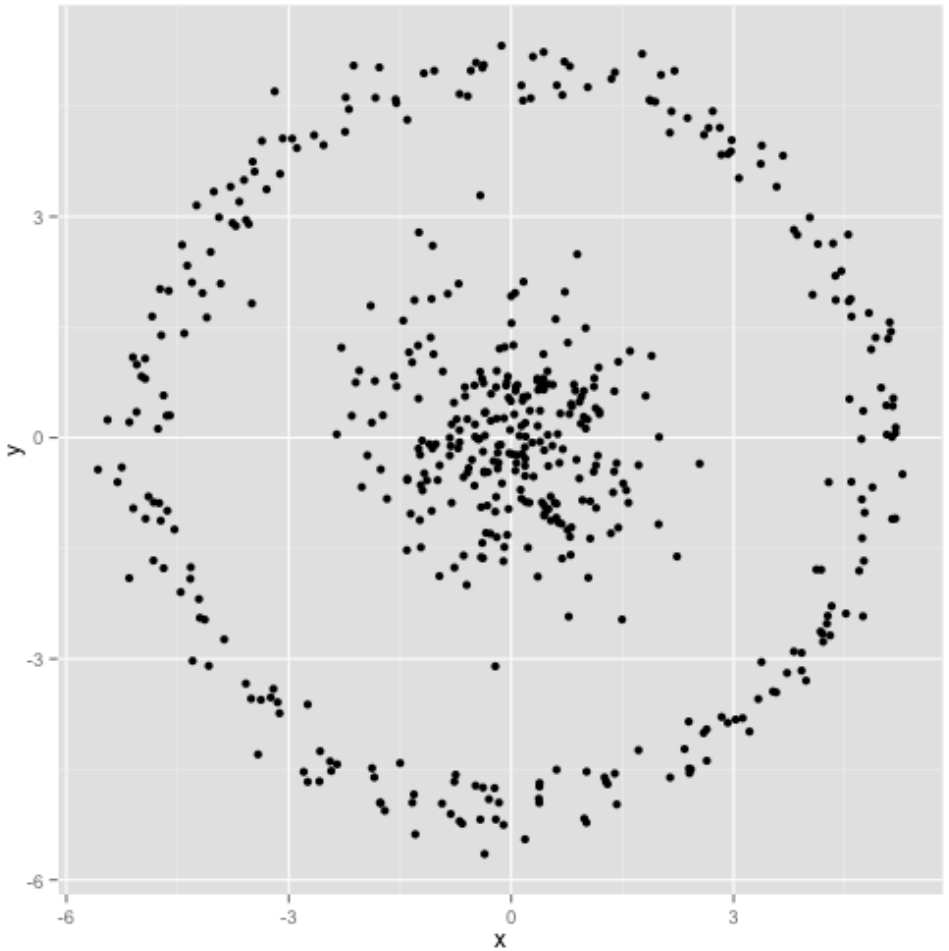
Байесовский информационный критерий

$$BIC_j = L_j(X) + \frac{d}{2}\log(n)$$

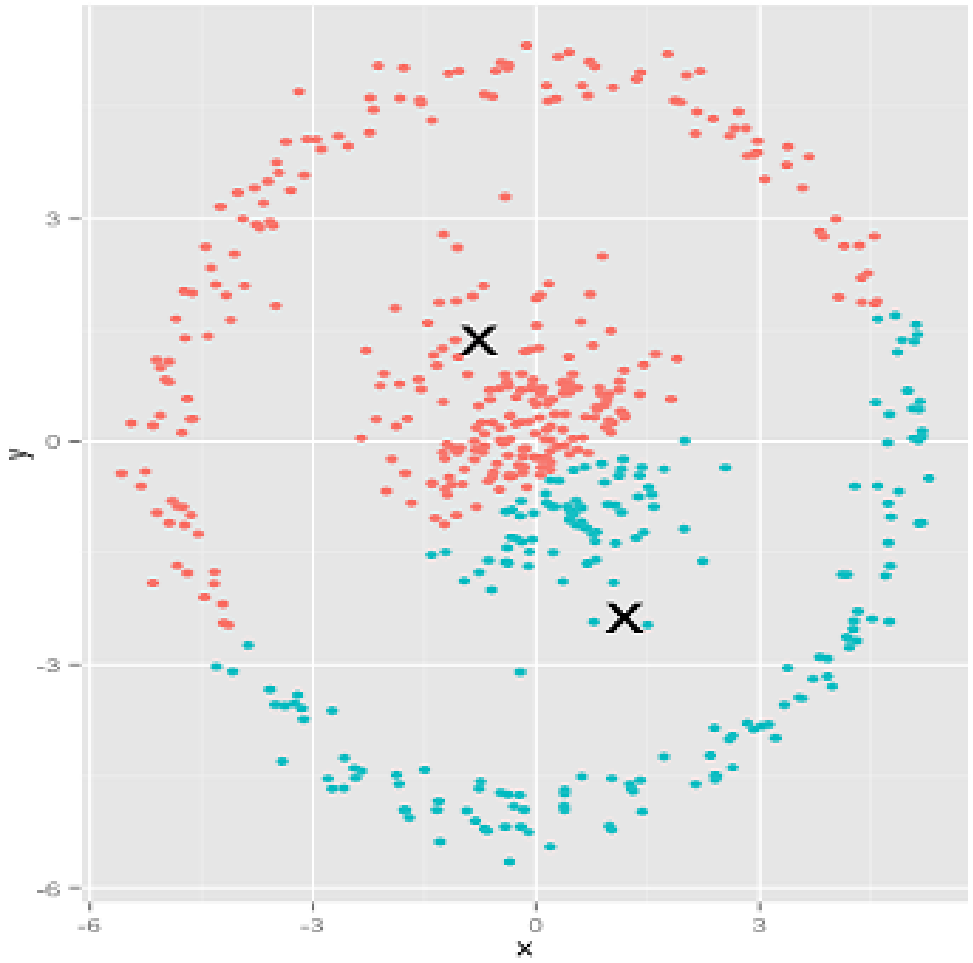
L_j – логарифмическая функция правдоподобия для j-й модели
d – длина вектора параметров
n – количество объектов в выборке

Недостатки k-means

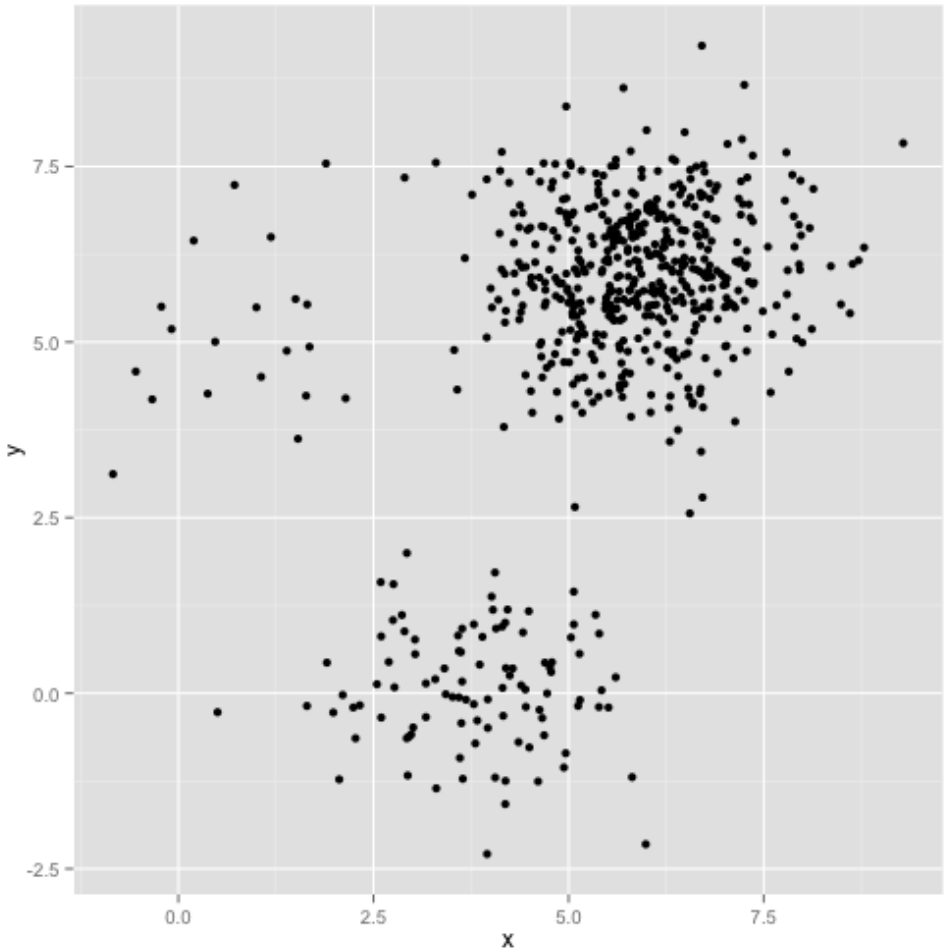
"Не сферические данные"



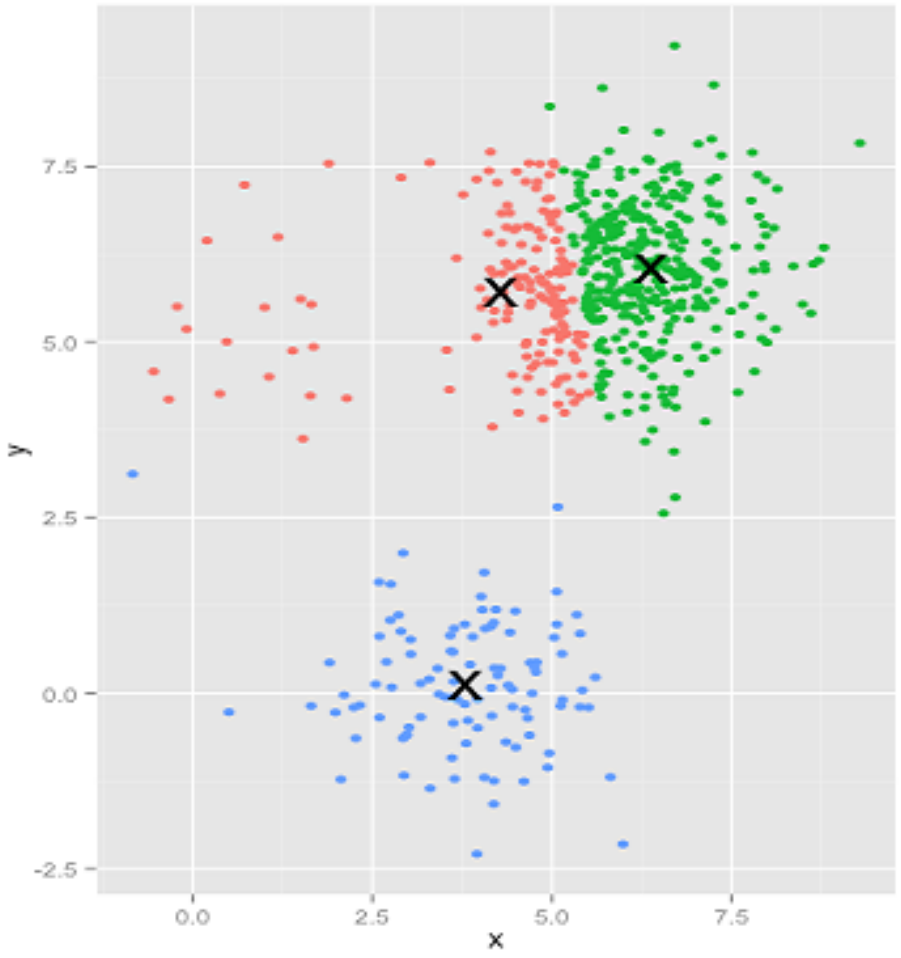
"Не сферические данные"



Разноразмерные кластеры



Разноразмерные кластеры



На следующей лекции

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