Введение в анализ данных: Кластеризация

Юля Киселёва
juliakiseleva@yandex-team.ru
Школа анализа данных

План на сегодня

- Задача кластеризации
- Методы кластеризации
- Алгоритм k-means
- Алгоритм CURE

Задача кластеризации

Дано:

- набор точек,
- заданы правила для определения расстояния между точками.

Задача:

Сгруппировать точки в определенное число кластеров:

  - Члены кластера либо расположены близко друг к другу, либо похожи друг на друга
  - Члены разных классов не похожи

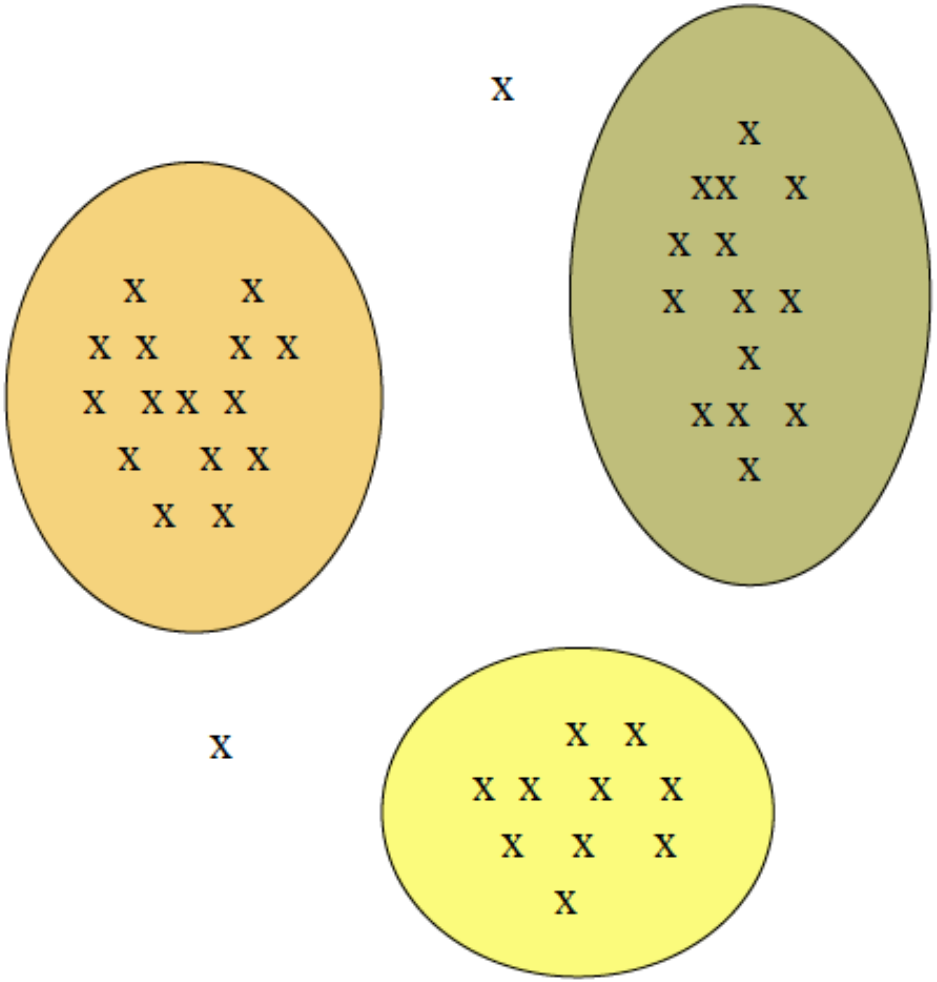
Задача кластеризации(2)

Обычно:

- Точки расположены в многомерном пространстве
- Похожесть определяется с использованием расстояний:

  Евклидово, косинусное, Jaccard similarity, edit distance,….

Кластеризация (пример)



Приложение: SkyCat

- **Дано:** каталог с описанием небесных объектов, есть измерения излучений каждого объекта
- **Задача:** Кластеризовать в похожие объекты, например галактики, ближайшие звезды, и др.

Приложение: Кластеризация фильмов

- Интуиция: Фильмы разделены на несколько категорий, и пользователи предпочитают несколько из них
- Представить фильм, как набор пользователей, которым он нравится (купили/выставили рейтинг)
- Похожие фильмы характеризуются похожим набором пользователей и наоборот

Приложение: Кластеризация фильмов (2)

Пространство для всех фильмов:

- Координатами являются уникальные пользователи
  - Фильм описывается набором из 0/1
  - Точка, характеризующая фильм в пространстве = $(x_1, x_2, ..., x_k)$, где x_i =1 если ***i*** –ый пользователь купил этот фильм

Cosine, Jaccard, Euclidean

- Выбор:

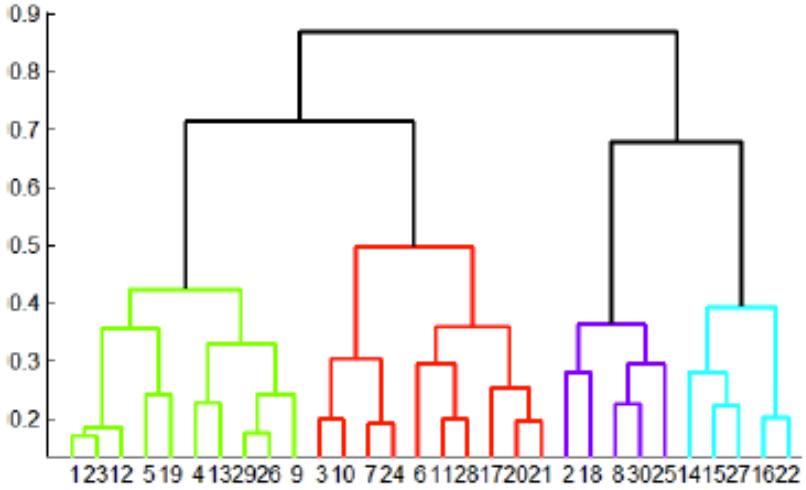
1. Набор как вектор: измерить похожесть с помощью косинусного расстояния
2. Набор как набор: измерить похожесть с помощью Jaccard Similarity
3. Набор как точки: измерить похожесть с помощью Евклидова расстояния

План на сегодня

- Задача кластеризации
- **Методы кластеризации**
- Алгоритм k-means
- Алгоритм CURE

Методы кластеризации

- Иерархический подход:
- Agglomerative (botton up):
  - Изначально каждый элемент – это отдельный кластер
  - Последовательно комбинируем ближайшие кластера в один
- Divisive (top down):
  - Начинает с одного кластера и последовательно разделяет его
- Point Assignment:
  - Изначально определяется набор кластеров
  - Точки попадают в ближайший кластер



Иерархическая кластеризация

- Основная операция:

  Последовательное объединение двух ближайших кластеров

- Три важных вопроса:

1. Как представить кластер, содержащий более одной точки?
2. Как определить ближайшие кластера?
3. Когда следует остановить объединение кластеров?

Иерархическая кластеризация (2)

- Ключевая задача: во время построения кластеров, нужно определить положение самого кластера.
- Необходимо, чтобы определить, расстояние между кластерами.
- Для евклидова случая: каждый кластер характеризуется

  центройдом = среднее всех точек
  - расстояние между кластерами = расстояние между их центройдами

Несколько определений

- Радиус – это максимальное расстояние между центройдом и всеми точками кластера
- Диаметр – это максимальное расстояние между двумя любыми точками в кластере

Что насчет неевклидова пространства?

- Объект в неевклидовом пространстве расстояние определяется на основе свойств точек, а не их положения в пространстве.
  - Не может быть среднего
- Нужно выбрать точку в кластере, и она должна символизировать центр кластера
- Решение 1: Clustroid = точка, которая ближе всех к другим точкам

Наиближайшая точка?

- Возможные определения для clustroid - это точка, которая характеризуется минимальным:

1. максимальное расстояние до других точек в кластере
2. Сумма расстояний до других точек в кластере
3. Сумма квадратов расстояний до других точек
   – Для расстояния ***d*** центра ***c*** и кластера ***C***

$$\min_{c} \sum_{x \in C} d(x,c)^2$$

Определение «похожести» кластеров

- Решение2: межкластерное расстояние = минимум среди расстояний между точками из разных кластеров

- Решение 3: Рассмотрим понятие «сплоченности» кластеров
  - Объединяем кластеры, объедения которых «сплочены» больше.

Сплоченность кластеров

- Решение 1: использовать диаметр кластеров
- Решение 2: использовать среднюю длину между точками в кластере
- Решение 3: применить подход, основанный на плотности: подсчитать диаметр кластера или среднюю длину между точками и разделить на количество точек в кластере

План на сегодня

- Задача кластеризации
- Методы кластеризации
- **Алгоритм k-means**
- Алгоритм CURE

Алгоритм k-means

- Используем евклидово пространство/ расстояние
- Выбираем ***K*** – число кластеров
- Выбираем ***K*** точек, которые будут отражать ***K*** кластеров

Пример выбора ***K*** точек: рандомно выбираем первую точку, затем выбираем ***k-1*** точек, каждую по возможности дальше от другой

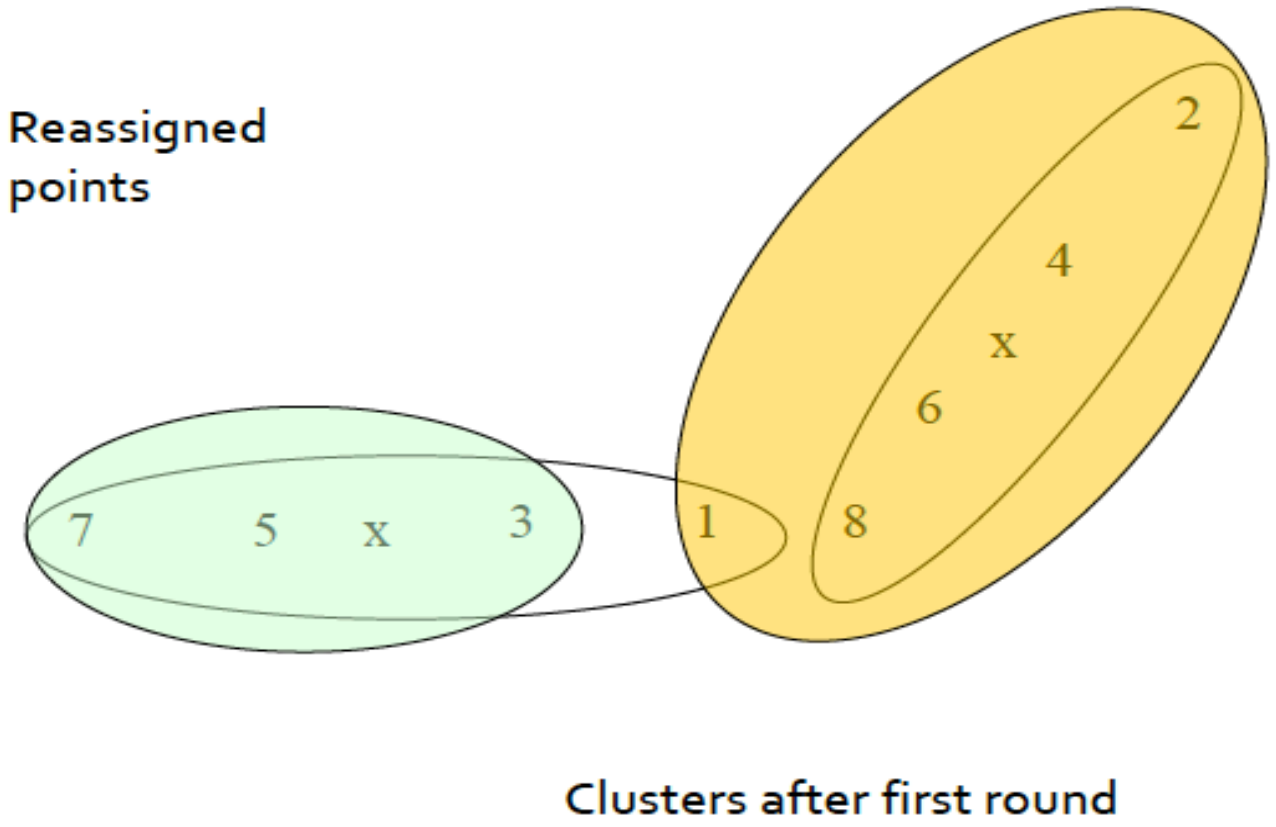
Получение кластеров

1. Каждая выбранная точка является центройдом, выбираем ближайшие к ней точки и помещаем их в кластер.

2. После того, как все точки помещены в ***к*** кластеров, пересчитываем центройды для каждого кластера

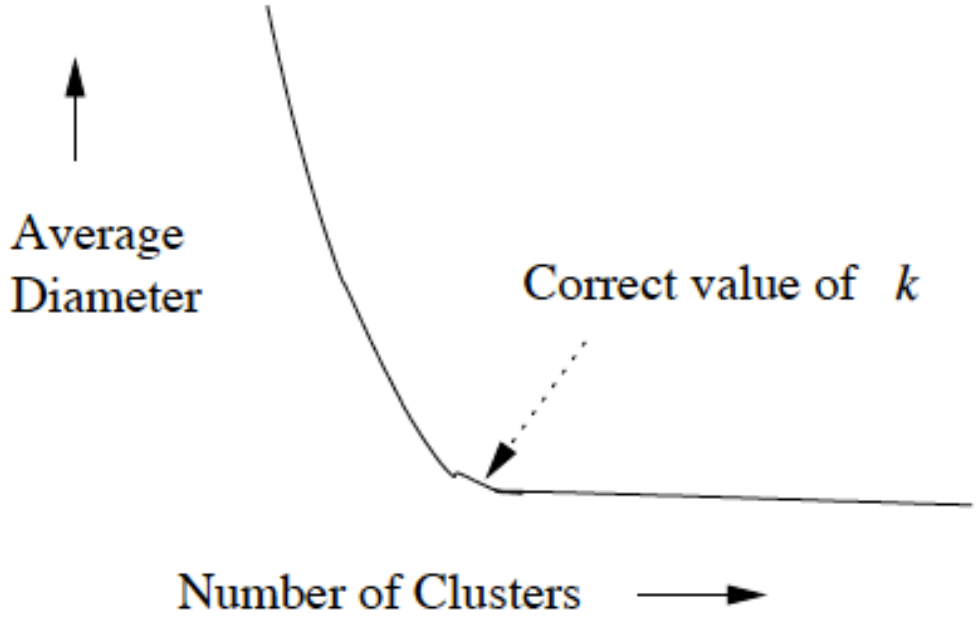
3. Далее: перемещаем все точки к ближайшему к ним центройду.

Пример: перемещения точек в другой кластер



Определение k

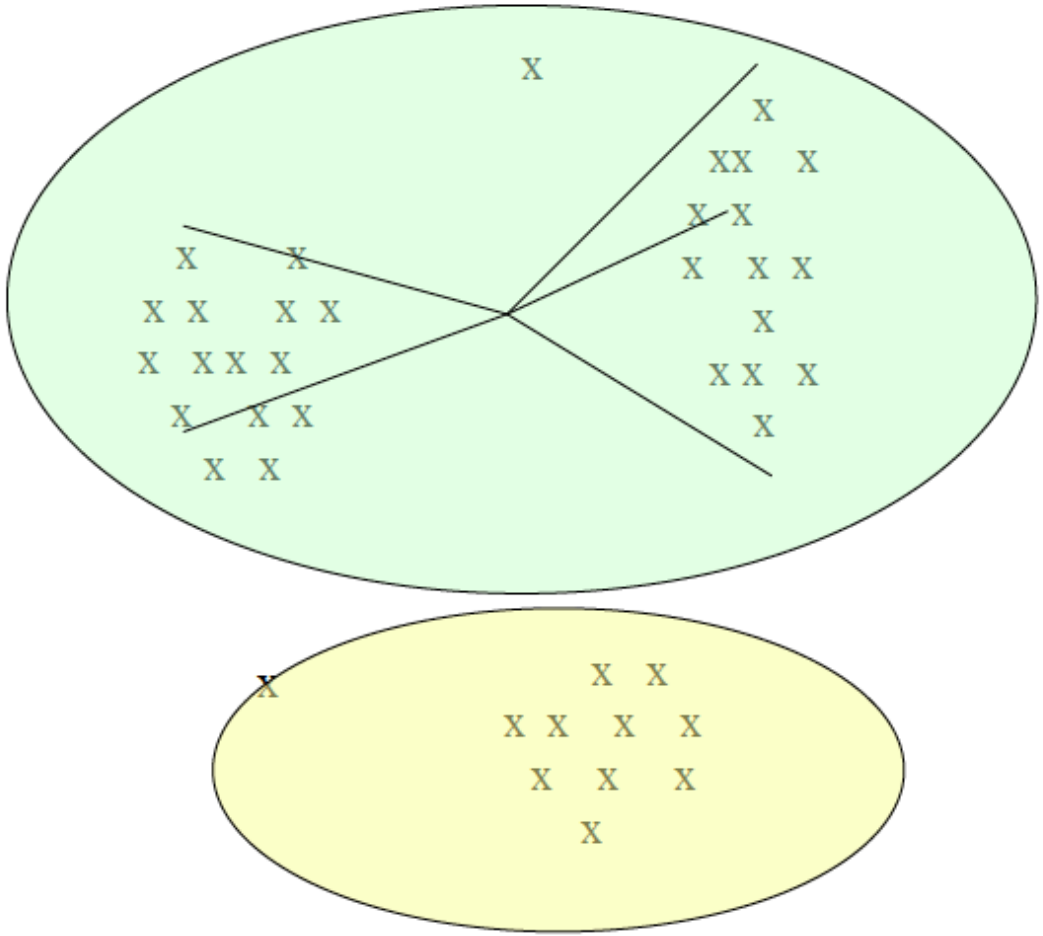
Средний диаметр (или иной мера диффузности) увеличивается быстро, как только количество кластеров падает ниже истинного числа, присутствующего в данных



Пример: выбор *K*

Слишком мало кластеров:

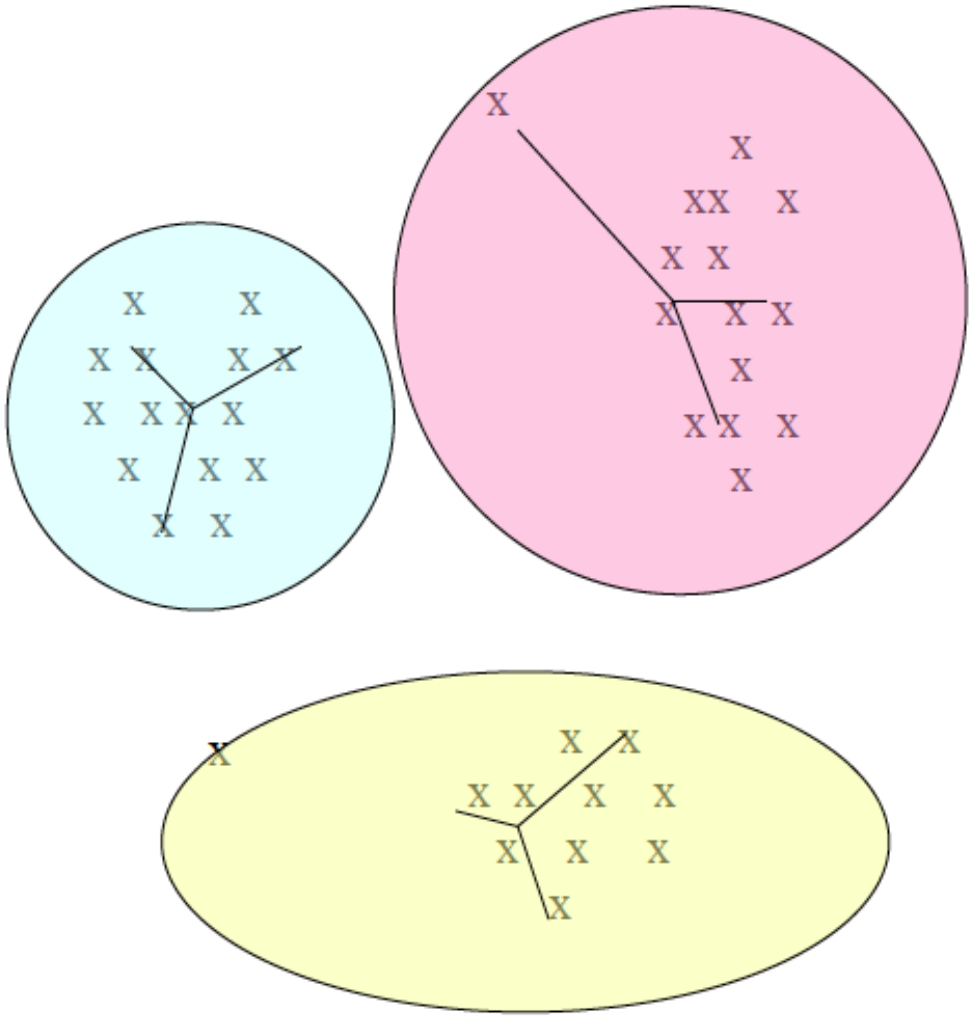
Много длинных Расстояний до центройда



Пример: выбор *K*

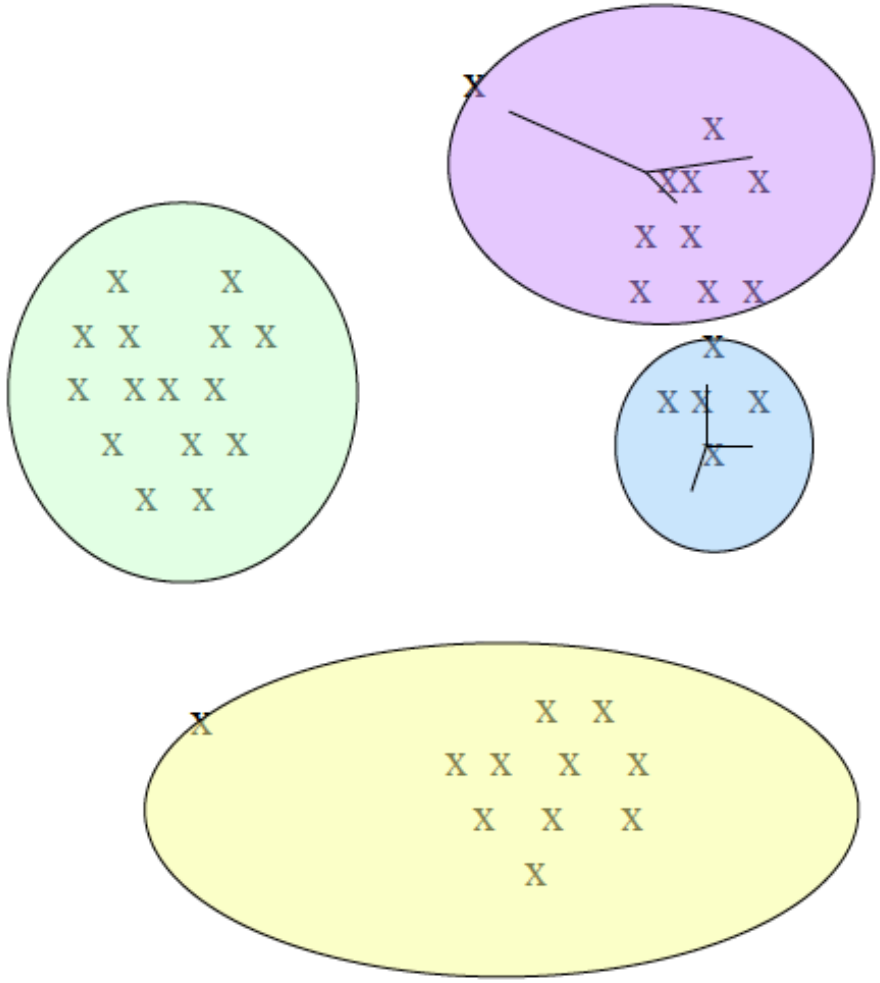
То что нужно:

Расстояние более короткие



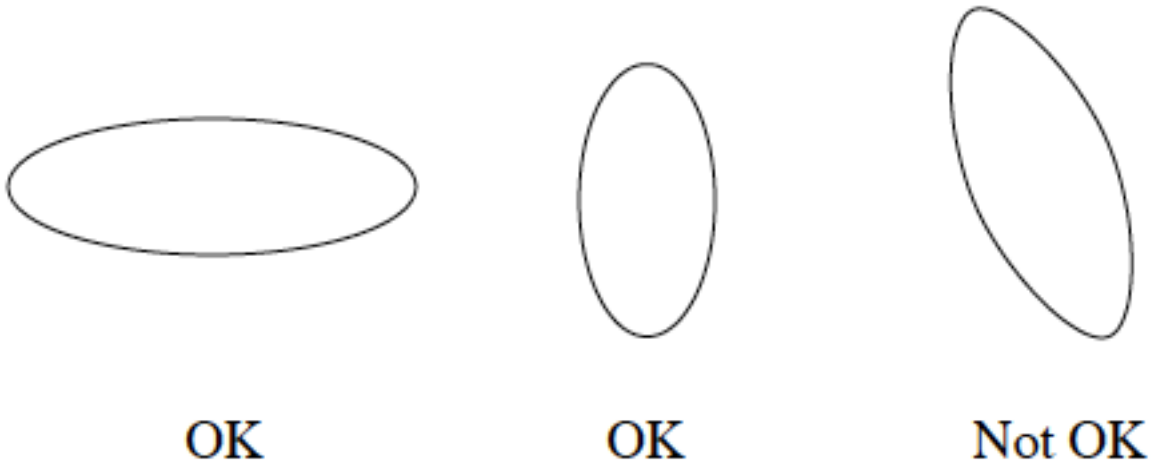
Пример: выбор *K*

Слишком много:
совсем небольшое
улучшение
средней длины



BFR Алгоритм

- BFR[Bradley-Fayyad-Reina] – это вариант алгоритма k-means, который был спроектирован для работы с большими объемами данных
- Предполагается, что кластеры распределены относительно центройда и имеют определенную форму:



План на сегодня

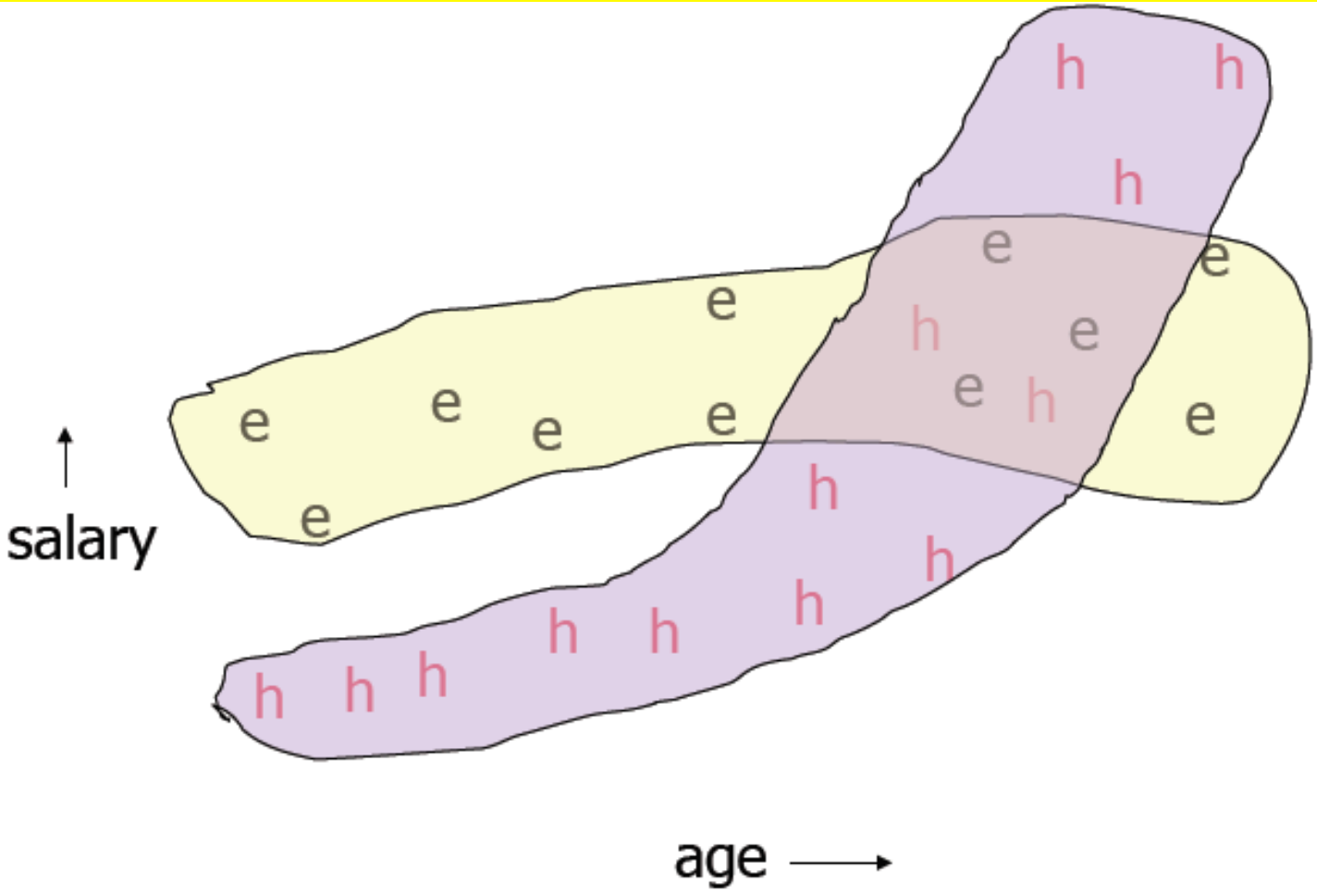
- Задача кластеризации
- Методы кластеризации
- Алгоритм k-means
- **Алгоритм CURE**

CURE Алгоритм

CURE = **C**lustering **U**sing **RE**presentatives

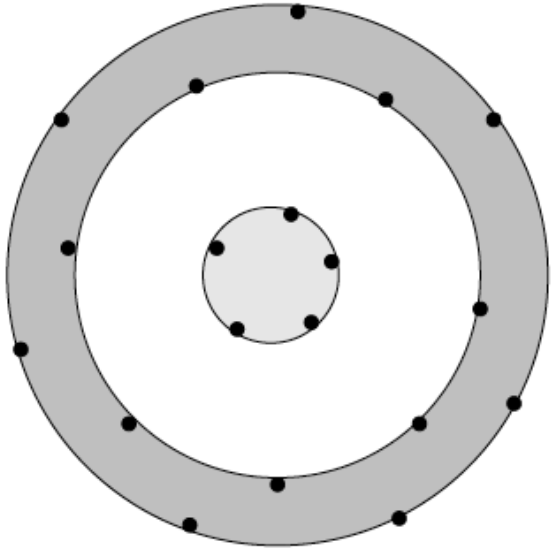
- Евклидово пространство
- Не заботится о форме кластеров
- Кластер представляется коллекцией репрезентативных точек

Пример: зарплата в Стэндфордском Университете



CURE Алгоритм

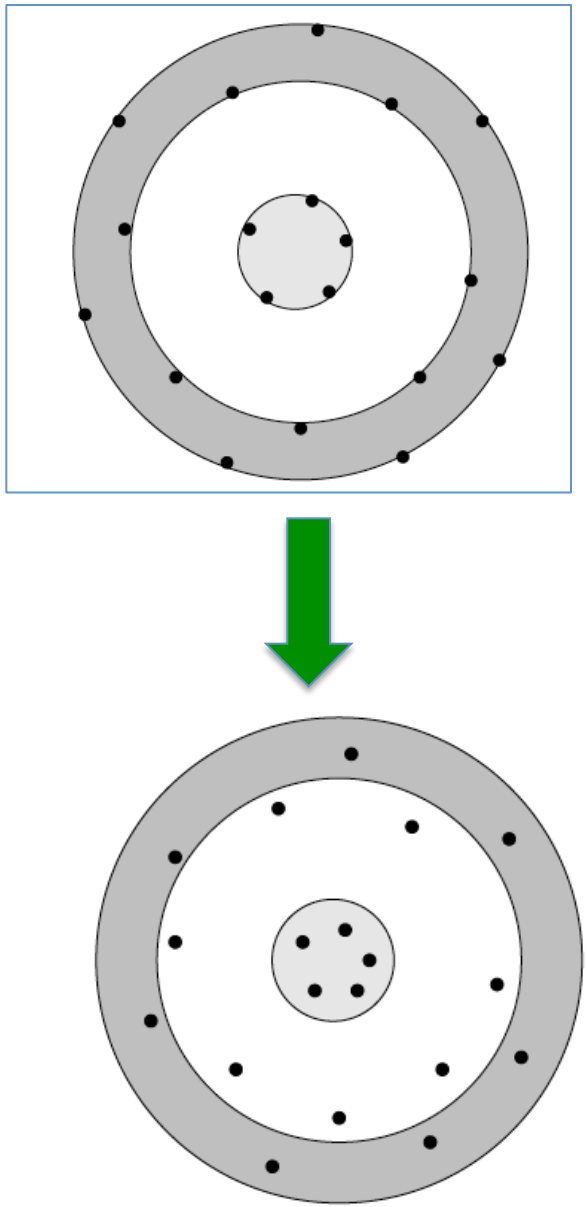
1. Случайным образом выбираем набор точек, которые помещаются в память
2. Кластеризуем этот набор помощью иерархического метода – группируем наиболее близкие точки
3. Для каждого кластера выбираем набор точек (представителей), которые удалены друг от друга насколько это возможно



CURE Алгоритм (2)

4. Из набора нужно выбрать представителей, перемещая их (скажем) 20% в сторону центра тяжести кластера
5. Затем обходим каждую точку ***p*** и перемещаем ее в ближайший кластер.

Определение: «Ближайшим» к ***p*** называется кластер, который содержит большее число ближайших к ***p*** точек



Метрики для оценки

- ***C-index*** (Dalrymple-Alford, 1970)
- ***Gamma*** (Baker & Hubert, 1975)
- ***Adjusted ratio of clustering*** (Roenker et al., 1971)
- ***D-index*** (Dalrymple-Alford, 1970)
- ***Modified ratio of repetition*** *(*Bower, Lesgold, and Tieman, 1969)
- ***Dunn's index*** (Dunn, 1973)
- **Variations of Dunn's index** (Bezdek and Pal, 1998)
- **Jagota index** (Arun Jagota 2003)
- ***Strict separation*** (based on Balacan, Blum, and Vempala, 2008)
- And many more...

Оценка (1)

- Jagota предложил метрику, которая отражает однородность кластера:

$$Q = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} d(\boldsymbol{x}, \mu_i)$$

- где $|\boldsymbol{C_i}|$ - это число элементов в кластере *l*
- *Q* будет маленьким, если (в среднем) точки в кластере близки друг к другу

Gamma

- За d(+) обозначим число раз, когда две точки, которые были кластеризованы вместе в кластер С имели расстояние большее, чем другие две точки не помещенные в один кластер
- За d(-) обозначим противоположный результат

$$\gamma = \frac{d(+) - d(-)}{d(+) + d(-)}$$

Резюме

- Познакомились с задачей кластеризации
- Ввели несколько определений
- k-means
- CURE
- Ввели методы оценки