Введение в Data Science
Занятие 9. Алгоритмы кластеризации

Николай Анохин Михаил Фирулик

26 апреля 2014 г.

Определение количества кластеров

Иерархическая кластеризация

DBSCAN

Задача кластеризации

Дано

- обучающая выборка $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$

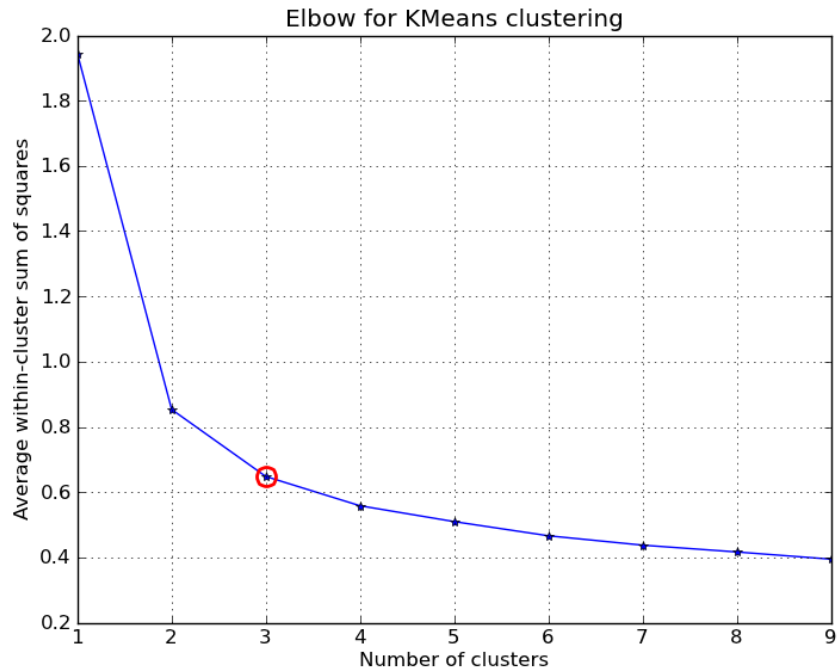
Задача

Разбить обучающую выборку на непересекающиеся множества (кластеры) так, чтобы объекты внутри одного кластера были близки, а объекты из разных кластеров отдалены

Выбор наилучшего K

Идея. Выбрать критерий качества кластеризации и построить его значение для $K = 1, 2, \ldots$

- средняя сумма квадратов расстояния до центроида
- средний диаметр кластера



Критерий Silhouette

Пусть дана кластеризация в K кластеров, и объект i попал в C_k

- $a(i)$ – среднее расстояние от i объекта до объектов из C_k
- $b(i) = min_{j \neq k} b_j(i)$, где $b_j(i)$ – среднее расстояние от i объекта до объектов из C_j

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

Средний silhouette для всех точек из $\mathbf{X}$ является критерием качества кластеризации.

Иерархическая кластеризация: идея метода

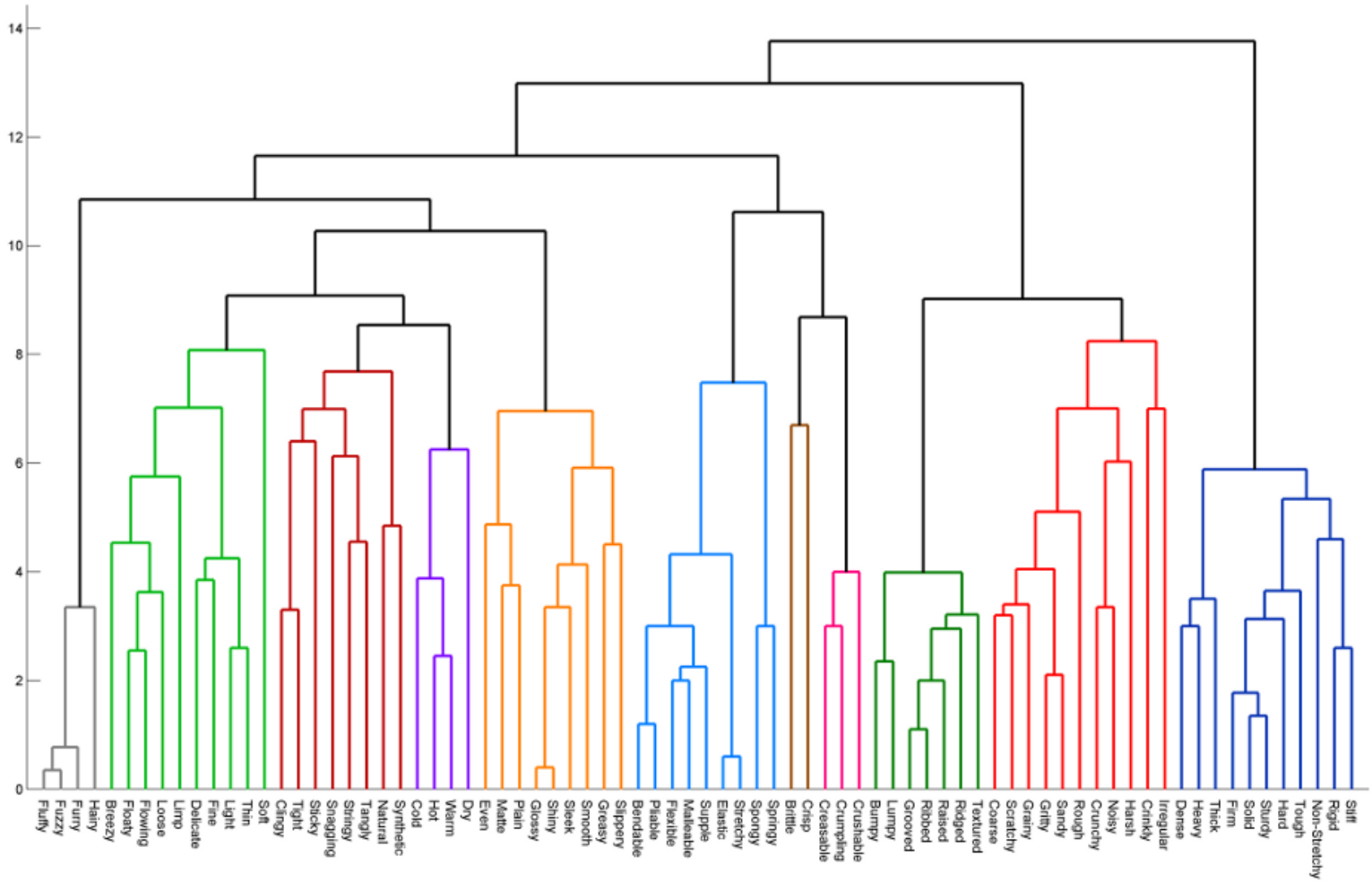
Agglomerative

1. начинаем с ситуации, когда каждый объект – отдельный кластер
2. на каждом шаге совмещаем два наиболее близких кластера
3. останавливаемся, когда получаем требуемое количество или единственный кластер

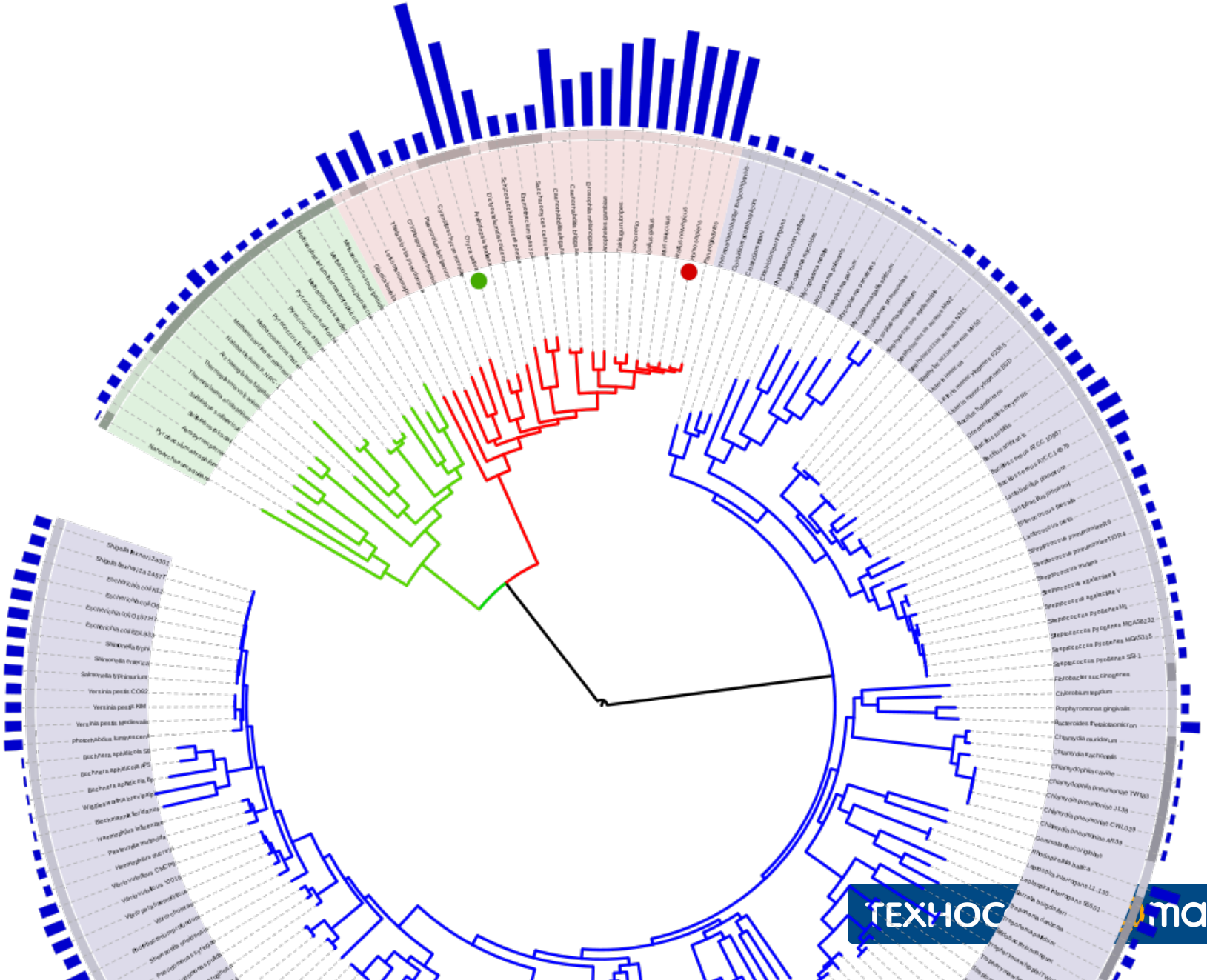
Divisive

1. начинаем с ситуации, когда все объекты составляют один кластер
2. на каждом шаге разделяем два один из кластеров пополам
3. останавливаемся, когда получаем требуемое количество или N кластеров

Дендрограмма



Радиальная дендрограмма



Агломеративный алгоритм

```
agglomerative(X, K):
  Инициализируем C_i ← x_i, C = N
  do
    Ищем ближайшие кластеры C_i и C_j
    Совмещаем ближайшие кластеры C_i и C_j
    C = C − 1
  until C = K or C = 1
  return C_1, ..., C_K
```

Алгоритмическая сложность: $O(n^2 \log n)$

Расстояние между кластерами

- single-linkage

$$d_{min}(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{x}' \in C_j} \|\mathbf{x} - \mathbf{x}'\|$$

- complete-linkage

$$d_{max}(C_i, C_j) = \max_{\mathbf{x} \in C_i, \mathbf{x}' \in C_j} \|\mathbf{x} - \mathbf{x}'\|$$

- average

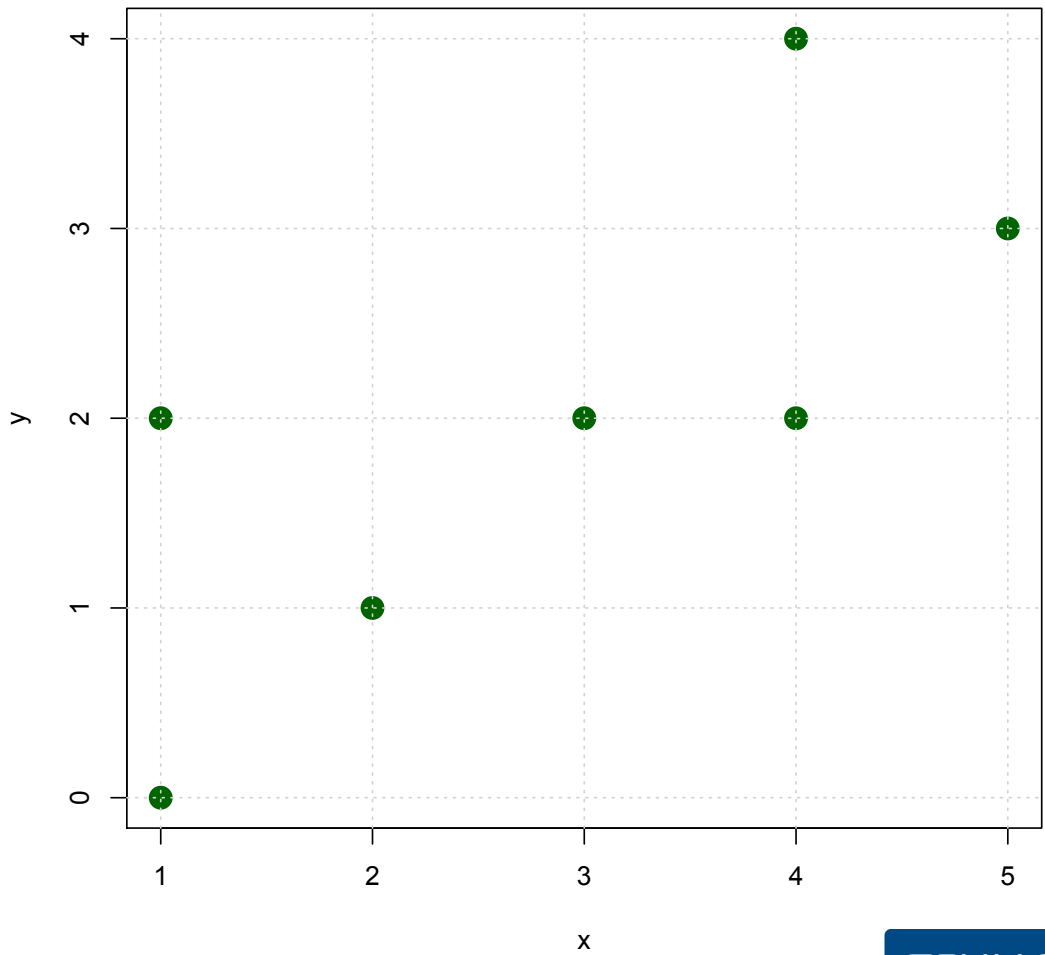
$$d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{\mathbf{x} \in C_i} \sum_{\mathbf{x}' \in C_j} \|\mathbf{x} - \mathbf{x}'\|$$

- mean

$$d_{mean}(C_i, C_j) = \|\mathbf{m}_i - \mathbf{m}_j\|$$

Задача

Кластеризовать данные иерархическим методом с использованием расстояний между кластерами d_{min} и d_{max}



Stepwise-optimal HC

Какой критерий мы оптимизируем?

```
swo(X, K):
  Инициализируем C_i ← x_i, C = N
  do
    Ищем C_i и C_j, после совмещения которых
      критерий оптимальности наиболее улучшится
    Совмещаем ближайшие кластеры C_i и C_j
    C = C − 1
  until C = K or C = 1
  return C_1, ..., C_K
```

d_{max} обеспечивает наименьшее увеличение диаметра кластера
d_e обеспечивает наименьшее увеличение квадратичного критерия

$$d_e(C_i, C_j) = \sqrt{\frac{n_i n_j}{n_i + n_j}} \|\mathbf{m}_i - \mathbf{m}_j\|$$

Неэвклидовы пространства

Проблема. Как измерить расстояние между кластерами, если невозможно определить центроид?

Идея. В каждом из кластеров выбрать “типичный” пример – clustroid.

Минимизируем

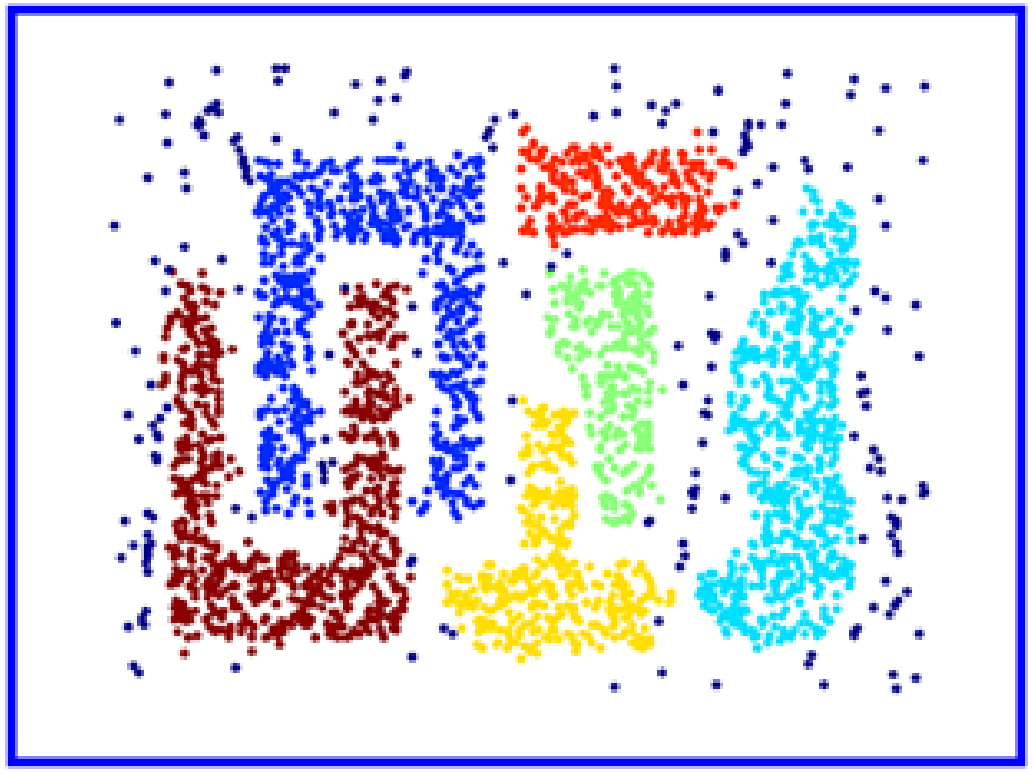
- сумму расстояний до других объектов в кластере
- сумму квадратов расстояний до других объектов в кластере
- максимальное расстояние до других объектов в кластере

Иерархическая кластеризация: итог

+ Несферические кластеры
+ Разнообразие критериев
+ Любые K из коробки
— Требует много ресурсов

DBSCAN: идея метода

- Кластеризация, основанная на плотности объектов
- Кластеры – участки высокой плотности, разделенные участками низкой плотности



Определения

Плотность

Количество объектов внутри сферы заданного радиуса ε

Core-объект

Объект **x** является core-объектом, если плотность вокруг него больше *min_pts*

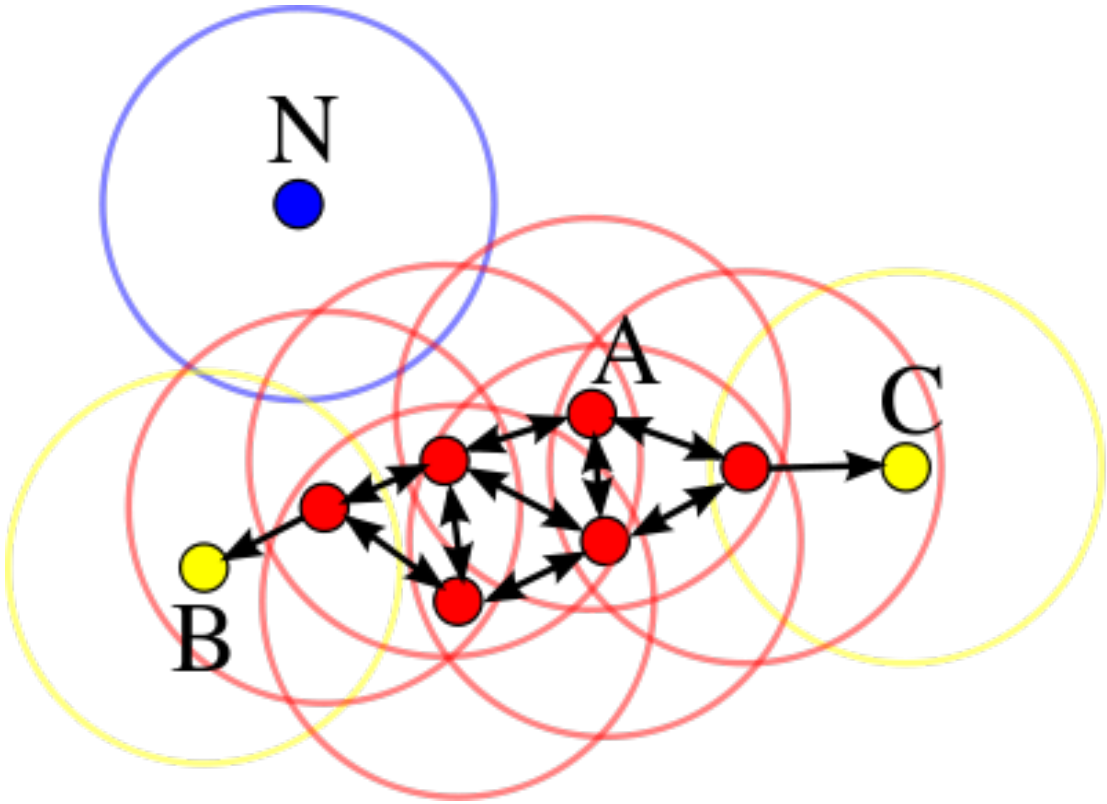
Граничный-объект

Объект **x** является граничным-объектом, если плотность вокруг него меньше *min_pts*, но он находится рядом с core-объектом

Шум

Объект **x** является шумом, если он не является ни core-объектом, ни граничным объектом

Виды объектов



DBSCAN 1

```
dbscan(X, ε, min_pts):
  for не посещенные P ∈ X:
    помечаем P как посещенный
    nbr = neigbors(P, ε)
    if len(nbr) < min_pts:
      помечаем P как шум
    else:
      C = create_cluster()
      expand_cluster(P, nbr, C, ε, min_pts)
      yield C
```

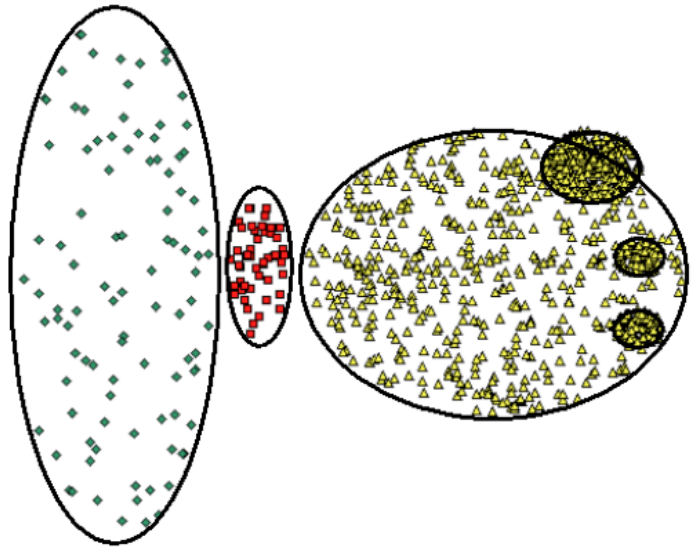
DBSCAN 2

```
expand_cluster(P, nbr, C, ε, min_pts):
  добавляем P в C
  for P' ∈ nbr:
    if P' не посещался:
      помечаем P' как посещенный
      nbr' = neighbors(P', ε)
      if len(nbr) ≥ min_pts:
    if P' не принадлежит ни одному кластеру:
      добавляем P' в C
```

Сложность: $O(n^2)$ или $O(n \log n)$ (R^* $Tree$)
Память: $O(n)$ или $O(n^2)$

DBSCAN: итог

- \+ не требует K
- \+ кластеры произвольной формы
- \+ учитывает выбросы
- — Не вполне детерминированный
- — Не работает при разных плотностях кластеров



Домашнее задание 2

Иерархическая кластеризация и DBCSAN

Реализовать алгоритм иерархической кластеризации и DBSCAN и протестировать на данных задачи модуля

Ключевые даты

- До 2014/05/03 00.00 выбрать ответственных
- До 2014/05/10 00.00 предоставить решения (после – половина очков)

На сегодня

1. Скачать ветку hier и запустить код для нескольких значений n
2. Реализовать метрики качества кластеризации
   - 2.1 Средней квадрат отклонения от центра
   - 2.2 Средний диаметр кластера
   - 2.3 Silhouette

   Позволяют ли эти метрики верно выбрать число кластеров?