Методы кластеризации

Игнатов Д., Кашницкий Ю.

Национальный исследовательский университет Высшая школа экономики
Департамент анализа данных и искусственного интеллекта

29 апреля 2016

План лекции

План лекции

Формулировка задачи

Cluster — гроздь, сгусток, пучок (*англ.*)
Основная задача кластерного анализа — разбиение исходного набора объектов на различающиеся между собой подмножества объектов, состоящие из близких элементов.

Кластерные структуры

- Разбиения
- Иерархии
- Нечеткие разбиения
- Бикластеры
- Смеси распределений

Применение кластеризации

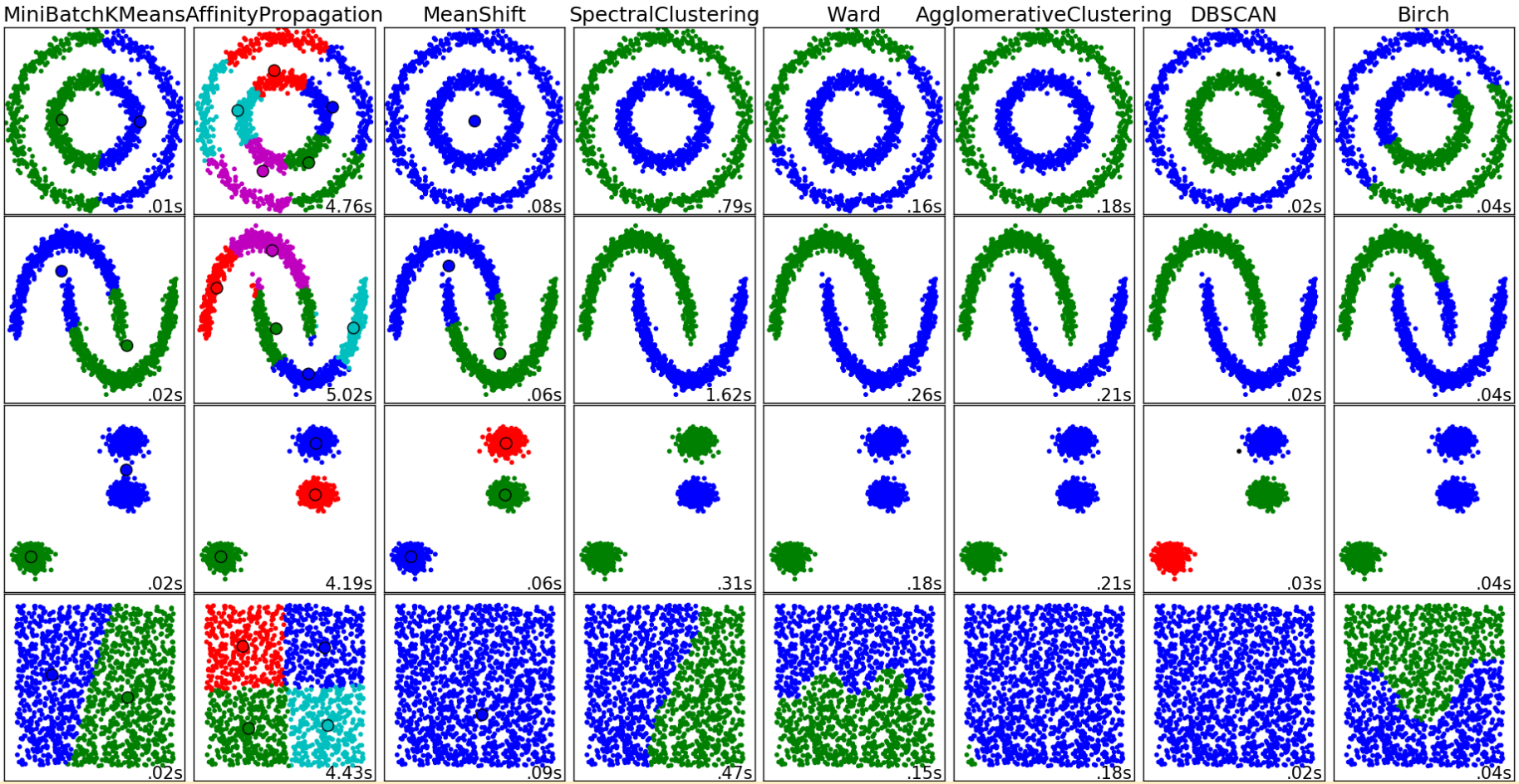
- Биология и медицина
  - Анализ экспрессии генов
  - Кластеризация томограмм
- Науки об обществе и человеке
  - Социология и антропология
  - Психология
- Технические системы
  - Телеметрия
  - Сегментация изображений
- Маркетинг
  - Сегментация потребителей
  - Анализ поведения групп
- Анализ текстов
- Социальные сети
  - Поиск сообществ

Классификация методов кластеризации

- Методы построения разбиений (Partitioning methods)
- Иерархические методы (Hierarchical methods)
- Методы, основанные на плотности (Density-based methods)
- Непараметрические методы (Non-parametric methods)
- Сеточные методы кластеризации (Grid-based methods)
- Мультимодальная кластеризация и бикластеризация (Multimodal clustering)

Сравнение методов кластеризации

Сравнение на игрушечных данных. Scikit-Learn®



Методы построения разбиений

- Выдача попарно-непересекающихся кластеров сферической формы
- Основано на расстоянии между объектами
- Кластер характеризуется центроидом — центром масс (k-means) или одним из объектов (k-medoids)

Иерархические методы

- Выдача иерархической структуры кластеров
- Последовательное объединение одноэлементных кластеров (агломеративное или снизу вверх) или разбиение тривиального кластера (все множество объектов) на несколько мелких (дивизимная или сверху вниз)

План лекции

Примеры «мер различия» объектов

Объекты $x \in \mathbb{R}^m$ представляются в виде матрицы «объект-признак»

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \Longleftrightarrow \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^m \\ x_2^1 & x_2^2 & \cdots & x_2^m \\ \cdots & \cdots & \cdots & \cdots \\ x_n^1 & x_n^m & \cdots & x_n^m \end{bmatrix}$$

Метрика Минковского

$$d(x,y) = \left[\sum_{i=1}^{m} |x^i - y^i|^p\right]^{\frac{1}{p}}$$

Косинусное расстояние

$$d(x,y) = 1 - \frac{\langle x,y\rangle}{\sqrt{\langle x,x\rangle}\sqrt{\langle y,y\rangle}}$$

Расстояние Хэмминга

$$d(x,y) = \frac{1}{m}\sum_{i=1}^{m}[x^i \neq y^i]$$

Метод k-средних

Метод k-средних является итеративным алгоритмом разбиения множества объектов на k классов.
Центр масс кластера (внутрикластерное среднее по каждому признаку) C_j называется **центроидом** и вычисляется как

$$c_j = \frac{1}{|C_j|} \sum_{i \in C_j} x_i$$

Целевая функция алгоритма есть сумма расстояний между объектами и центроидами классов, к которым они принадлежат

$$J(C) = \sum_{j=1}^{k} \sum_{i \in C_j} d(x_i, c_j)^2$$

Метод k-средних

Этапы алгоритма

Вход: Данные, k — параметр

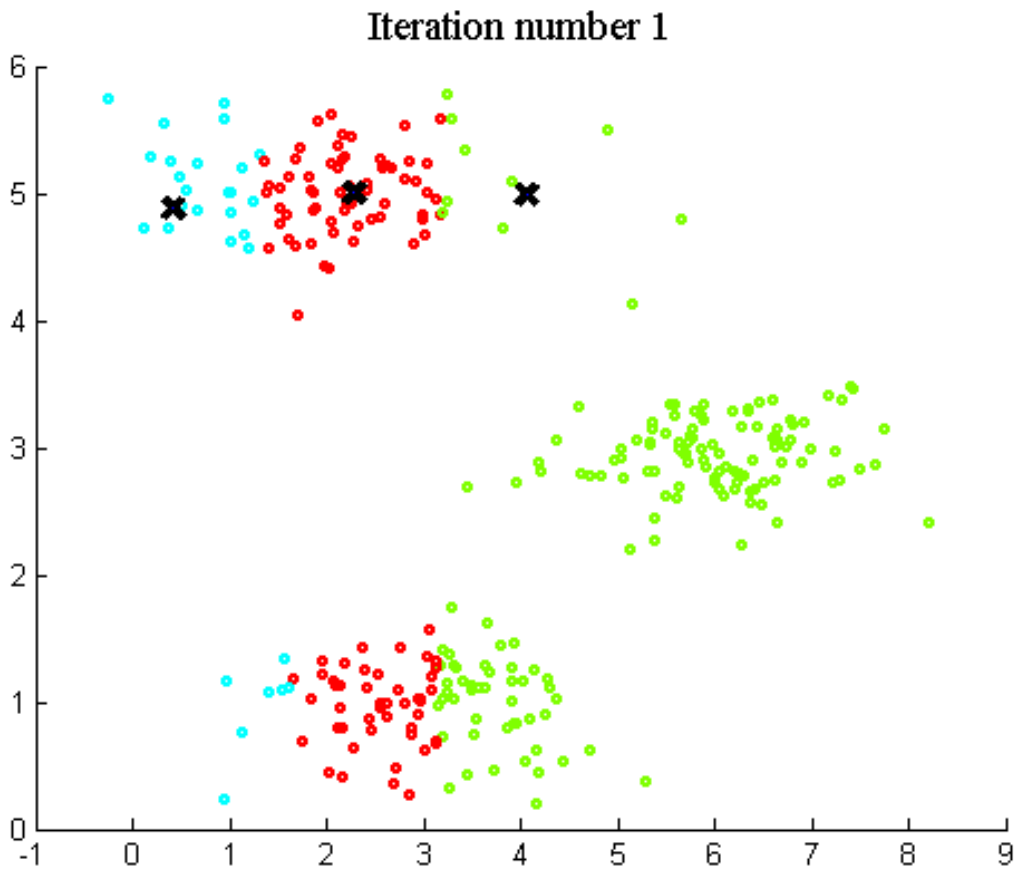
Выход: Разбиение, состоящее из k кластеров

* * *

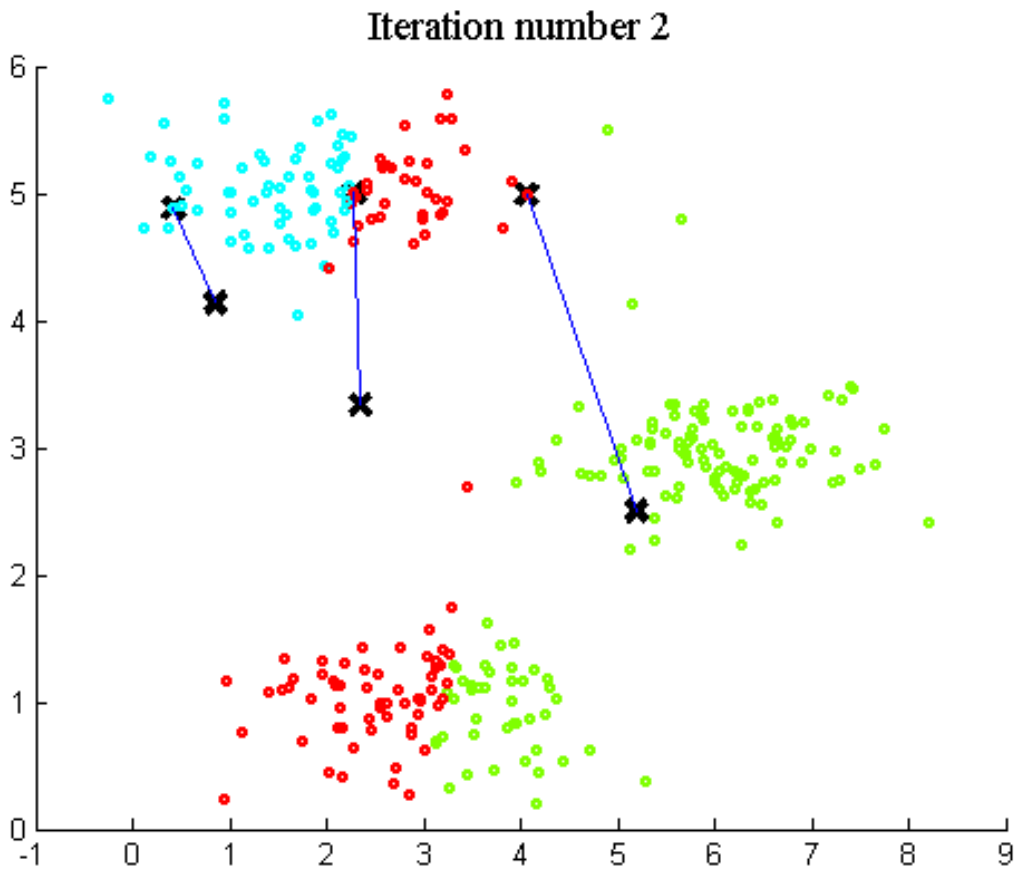
1. Инициализация: Назначение k точек в качестве начальных центроидов.
2. Обновление кластеров: При заданных k центроидах, каждый объект приписывается к ближайшему центроиду. Объекты, приписанные к центроиду c_j $(j = 1 \dots k)$, образуют кластер C_j.
3. Обновление центроидов: Для каждого кластера C_j вычисляется центр масс, который объявляется новым центроидом.

Итеративный процесс 2-3 продолжается до тех пор, пока получаемые кластеры изменяются.

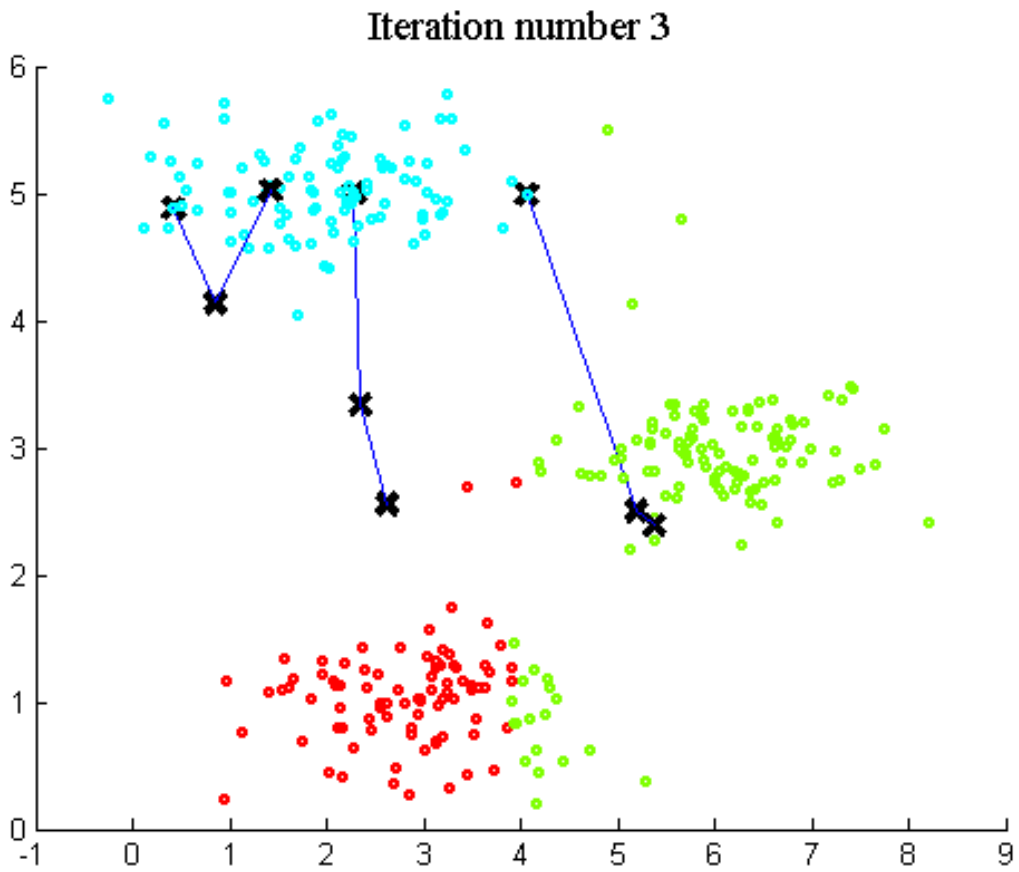
k-means. Пример



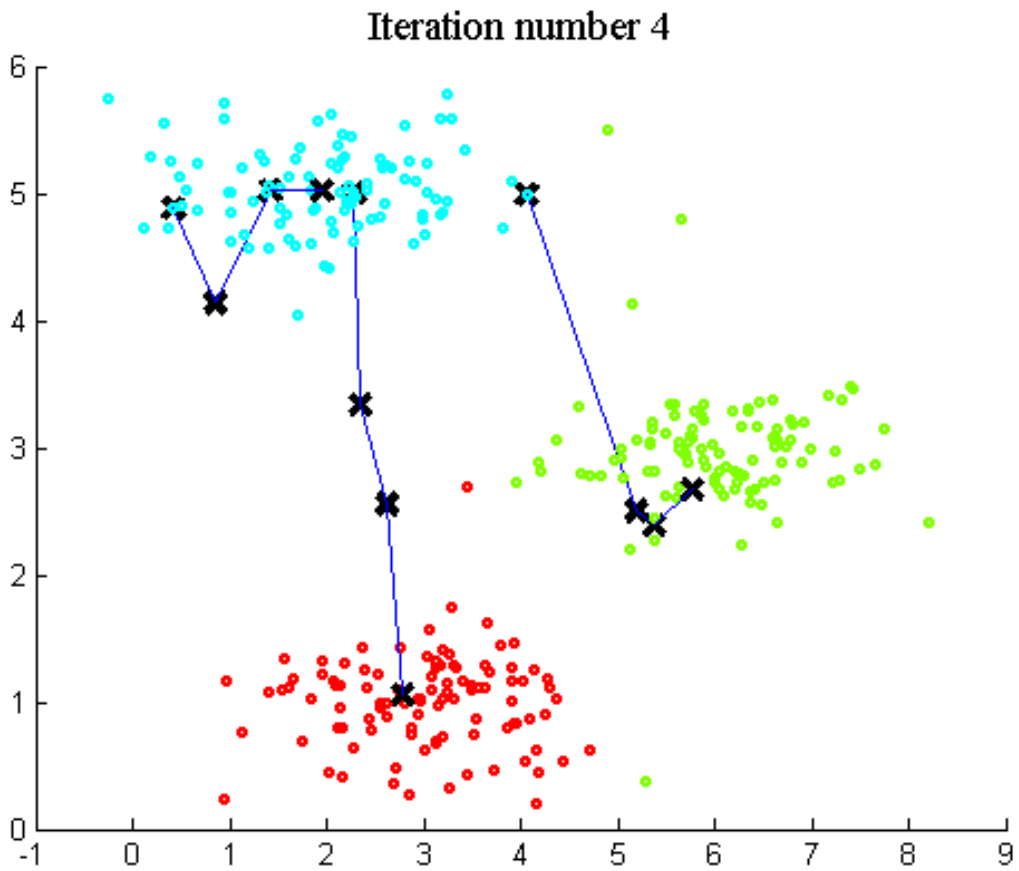
k-means. Пример



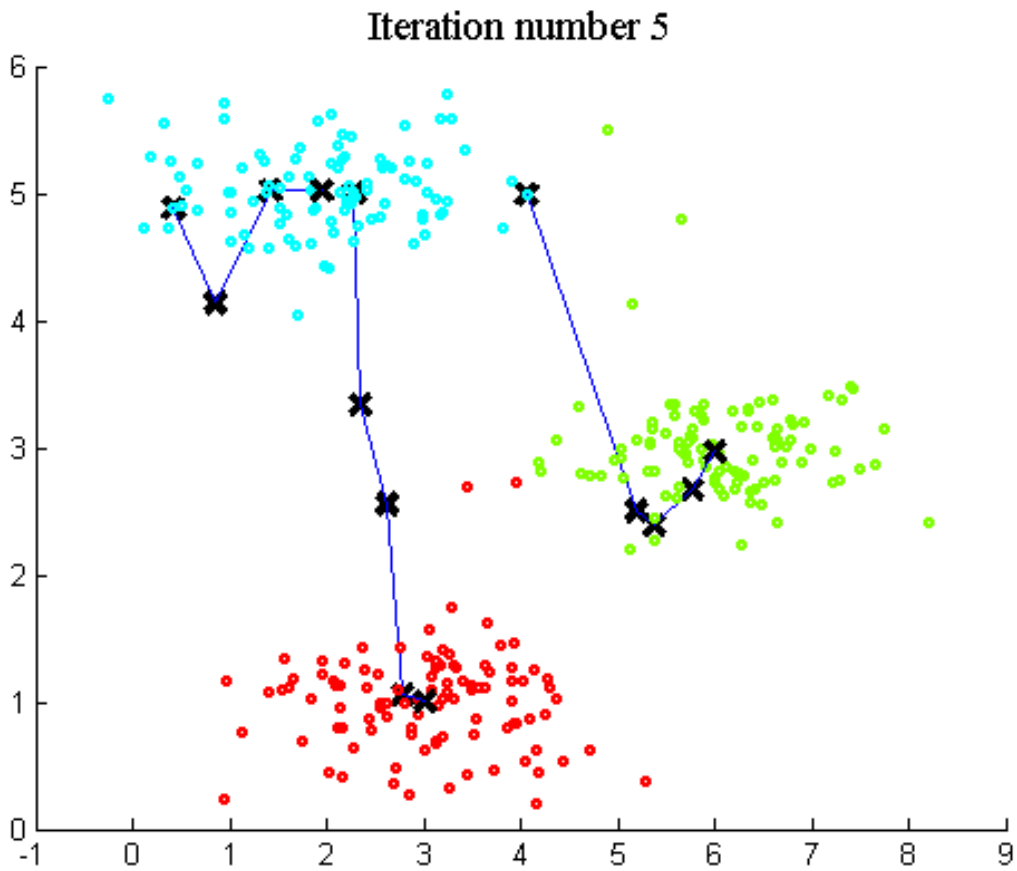
k-means. Пример



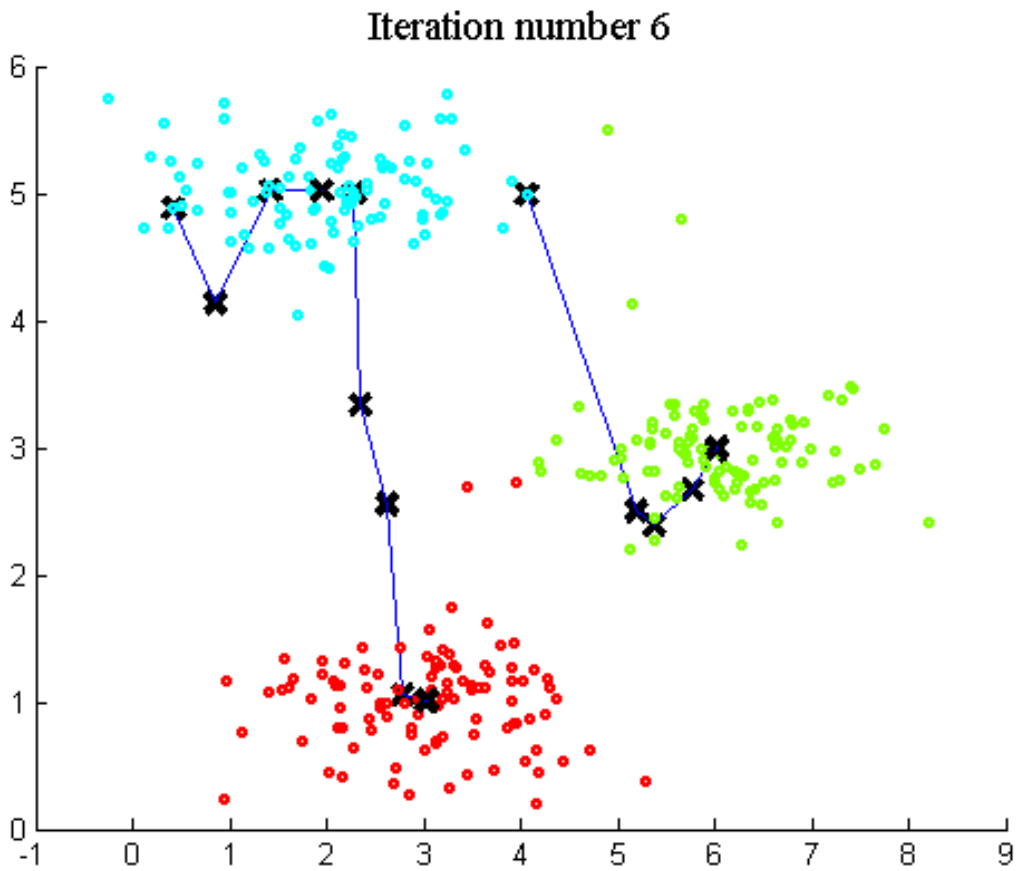
k-means. Пример



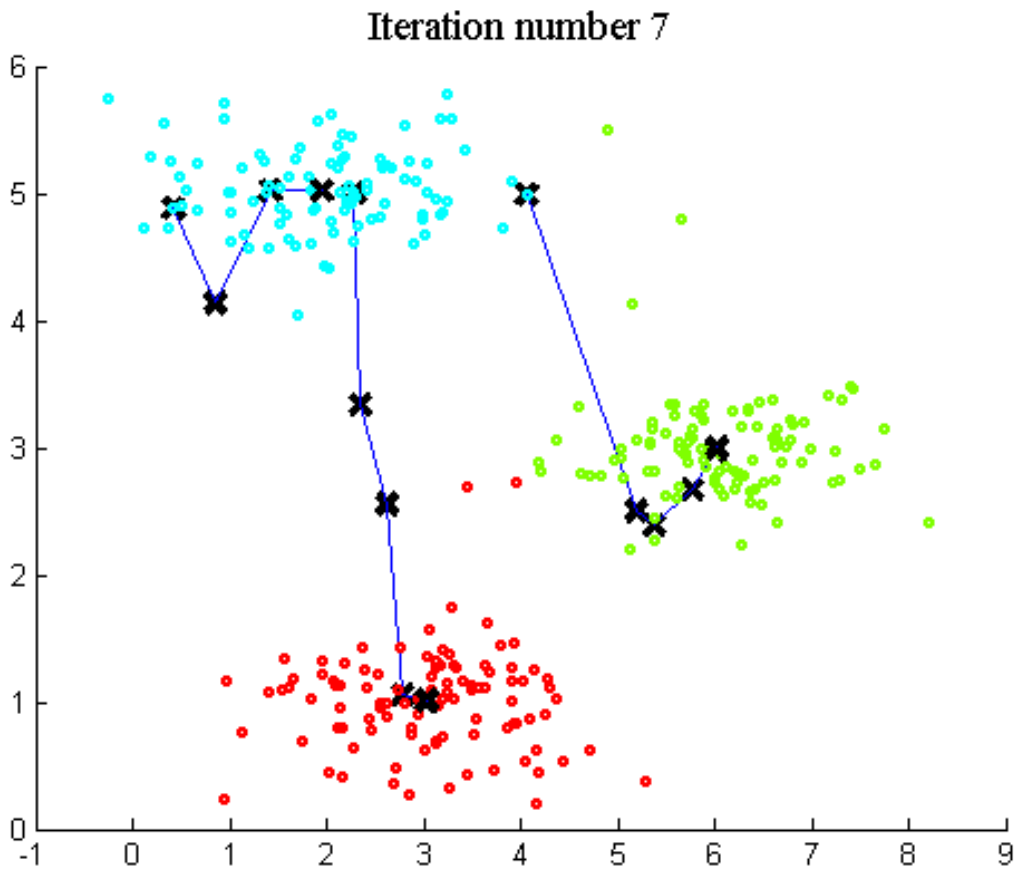
k-means. Пример



k-means. Пример



k-means. Пример



Оценки качества / числа классов

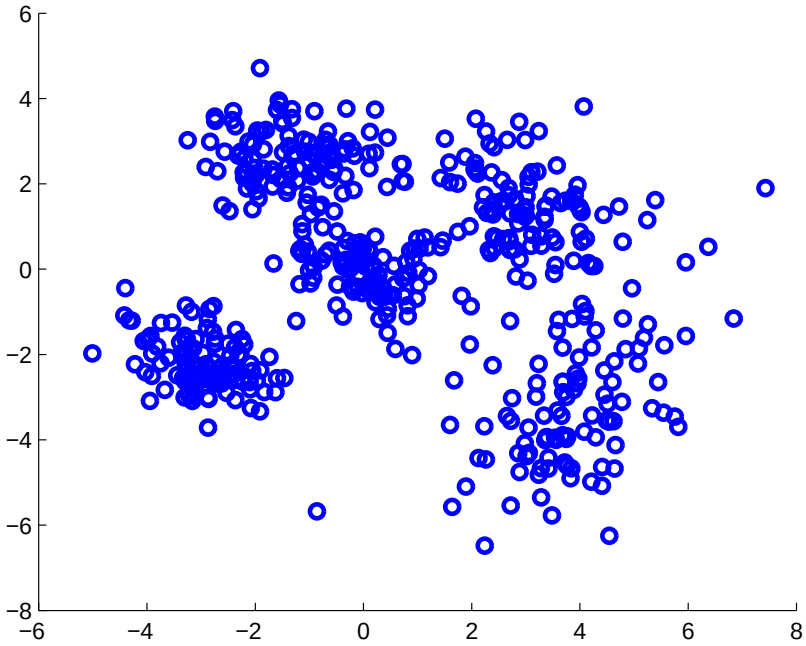
Метод локтя (elbow method)

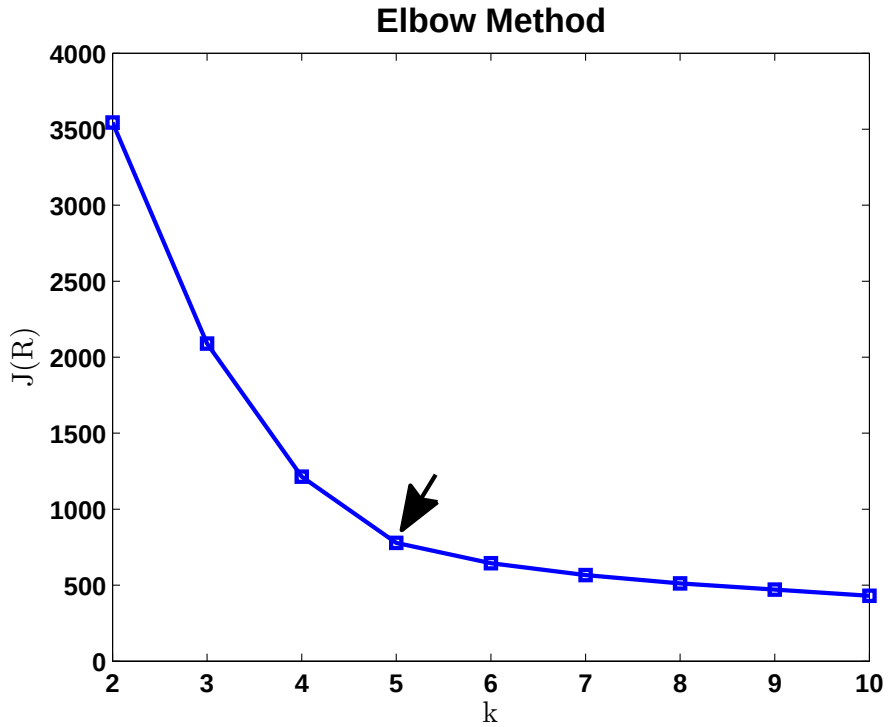
Каждому k ставится в соответствие значение функционална $J(C)$.
Принятие решения о количестве кластеров заключается в том, что нужно найти такую точку k', начиная с которой значения функционала $J(C)$ падают «не слишком быстро».
То есть соотношение невелико:

$$D(k) = \frac{|J(k) - J(k+1)|}{|J(k-1) - J(k)|}$$

Оценки качества / числа классов

Метод локтя (elbow method)





Оценки качества / числа классов

Силуэт кластеров

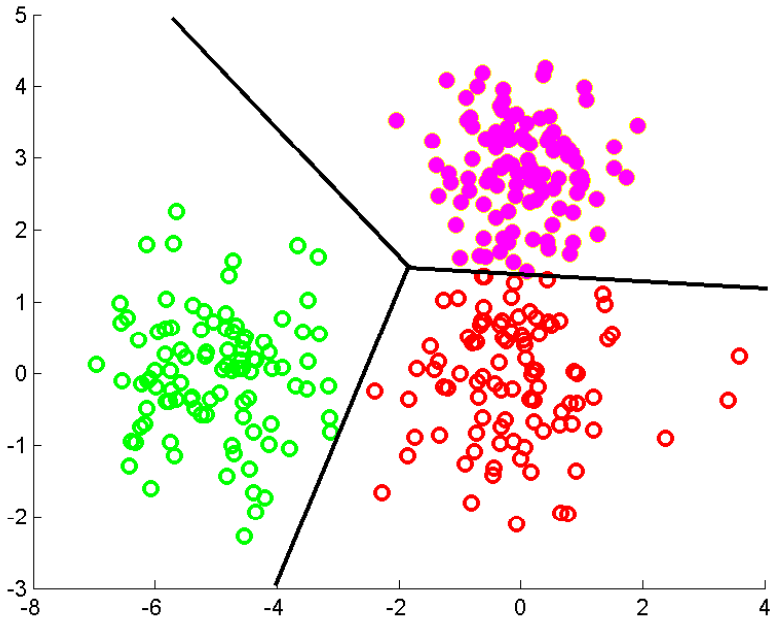
Силуэтом кластера C_h называют функцию

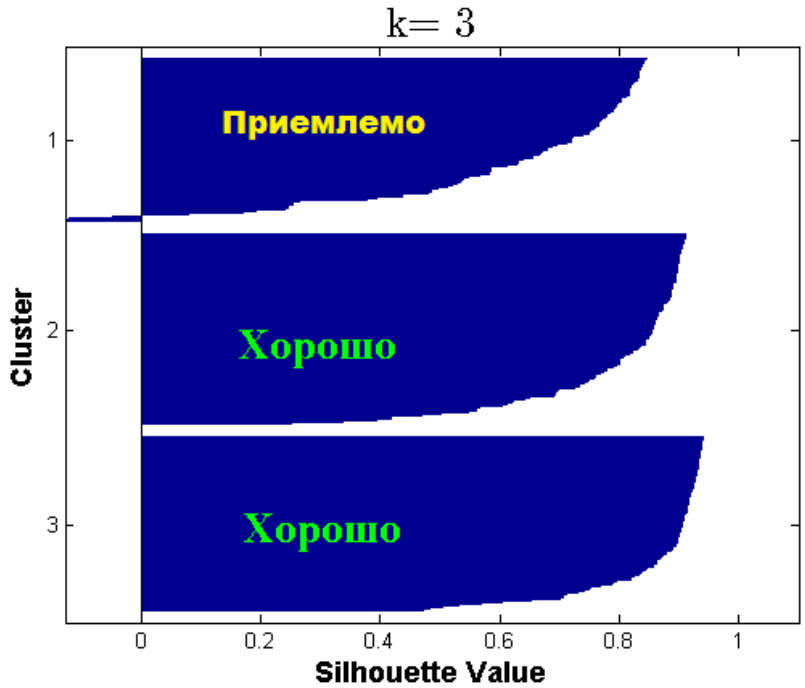
$$s_h(i) = \frac{\min\limits_m\{b_m(i)\} - a(i)}{\max\{a(i), min(b_m(i))\}} \qquad (m = 1, \ldots, k, m \neq h),$$

где $a(i)$ — среднее расстояние от i-го элемента кластера C_h до каждого из остальных элементов этого кластера, а $b_m(i)$ — среднее расстояние до элементов одного из «прочих» кластеров.

Силуэт

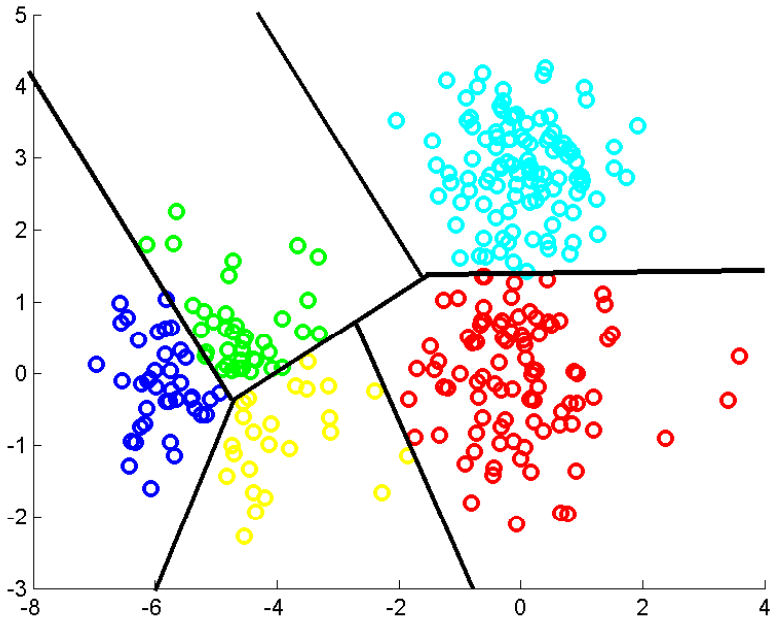
Приемлемое число кластеров

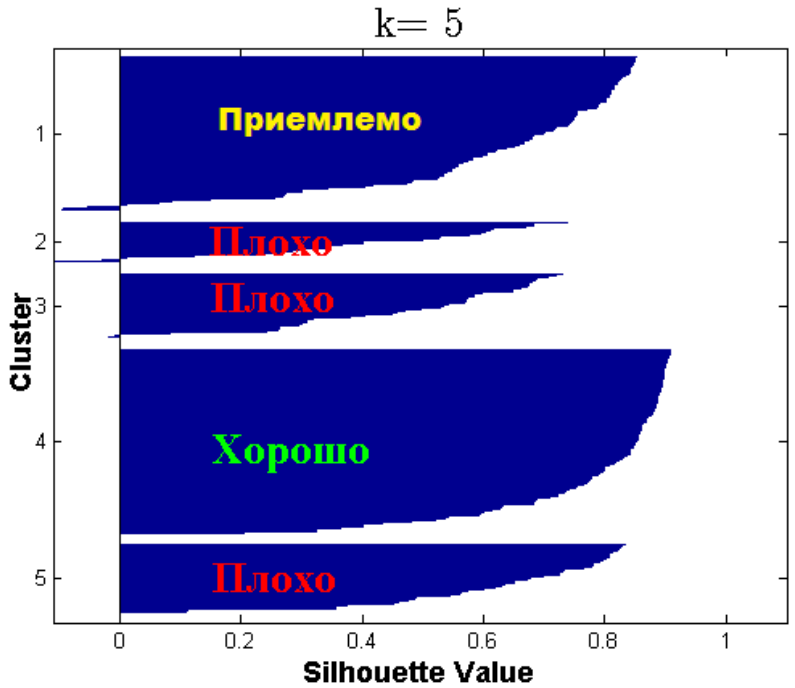




Силуэт

Неудачное число кластеров





Метод k-медоидов

Идея метода похожа на k-средних, однако, теперь центроиды (здесь медоиды) всегда являются объектами исходной выборки.

- Устойчивость к выбросам
- Медленная работа

Этапы алгоритма

1. Инициализация: Назначение k точек в качестве начальных медоидов M
2. Обновление кластеров: При заданных k медоидах, каждый объект приписывается к ближайшему медоиду. Объекты, приписанные к медоидам c_i $(i = 1 \dots k)$, образуют кластер C_i
3. Смещение медоидов: Для каждой пары (c_i, x_h), $c_i \in M$ и $x_h \in \overline{M}$, вычисляется $Cost(c_i, x_h) = J(C') - J(C)$, где C' - разбиение, в котором объект x_h является медоидом вместо c_i.
   Если $Cost < 0$, то объект x_h назначается медоидом вместо c_i

Итерационный процесс 2-3 продолжается до тех пор, пока получаемые кластеры изменяются.

Fuzzy c-means

[Bezdek, 1981]

- Пусть w_{ij} — степень принадлежности объекта x_i к кластеру C_j, $i = 1, \ldots, n$, $j = 1, \ldots, k$.
- $\sum_j w_{ij} = 1$.
- Целевая функция

$$J(C) = \sum_{j=1}^{k} \sum_{i \in C_j} w_{ij}^p d(x_i, c_j)^2$$

- p — параметр влияния весов

Fuzzy c-means

Этапы алгоритма

Вход: Данные, k — параметр

Выход: Матрица степеней принадлежности $W^{n \times k}$

* * *

1. Инициализация: Назначение k точек в качестве начальных центроидов

2. Обновление степеней принадлежности: $w_{ij} = \frac{(1/d(x_i, c_j)^2)^{\frac{1}{p-1}}}{\sum_{q=1}^{k} (1/d(x_i, c_q)^2)^{\frac{1}{p-1}}}$

3. Обновление центроидов: $c_j = \frac{\sum_{i=1}^{n} w_{ij}^p x_i}{\sum_{i=1}^{n} w_{ij}^p}$

Итеративный процесс 2-3 продолжается до тех пор, пока полученные кластеры изменяются (или изменение целевой функции несущественно).

Иерархические методы

От матрицы «объект-признак» можно перейти к матрице попарных расстояний между объектами

$$\begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^m \\ x_2^1 & x_2^2 & \cdots & x_2^m \\ \cdots & \cdots & \cdots & \cdots \\ x_n^1 & x_n^m & \cdots & x_n^m \end{bmatrix} \Rightarrow \begin{pmatrix} d(x_1,x_1) & d(x_1,x_2) & \cdots & d(x_1,x_n) \\ d(x_2,x_1) & \ddots & \ddots & d(x_2,x_n) \\ \vdots & \ddots & \ddots & \vdots \\ d(x_n,x_1) & d(x_n,x_2) & \cdots & d(x_n,x_n) \end{pmatrix}$$

Матрица расстояний симметрична, на главной диагонали нули.

Иерархические методы

От матрицы «объект-признак» можно перейти к матрице попарных расстояний между объектами

$$\begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^m \\ x_2^1 & x_2^2 & \cdots & x_2^m \\ \cdots & \cdots & \cdots & \cdots \\ x_n^1 & x_n^m & \cdots & x_n^m \end{bmatrix} \Rightarrow \begin{pmatrix} 0 & d(x_1,x_2) & d(x_1,x_3) & \cdots & d(x_1,x_n) \\ & 0 & d(x_2,x_3) & \cdots & d(x_2,x_n) \\ & & \ddots & \cdots & \cdots \\ & & & 0 & d(x_{n-1},x_n) \\ & & & & 0 \end{pmatrix}$$

Матрица расстояний симметрична, на главной диагонали нули.

Агломеративная кластеризация

Агломеративный подход — последовательное объединение близких кластеров.

0. Начиная с одноэлементных кластеров
1. Найти пару наиболее близких кластеров
2. Объединить два кластера

Продолжать шаги 1-2 пока все объекты не объединятся в один кластер.

Предположим, выбрана мера расстояния и найдена пара наиболее близких объектов. Произведено их объединение в кластер большего размера.
Вопрос: как вычислить расстояния между новым и другими кластерами?

Агломеративная кластеризация

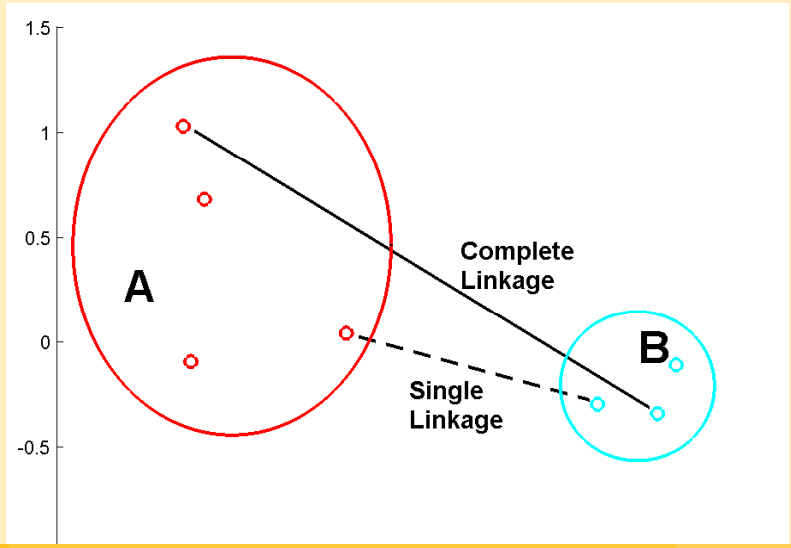
Типы связей

1. Одиночная связь (Single Linkage)

$$d(A, B) = \min_{x \in A, y \in B} d(x, y)$$

2. Полная связь (Complete Linkage)

$$d(A, B) = \max_{x \in A, y \in B} d(x, y)$$



Агломеративная кластеризация

Типы связей

3. Средняя связь (Average Linkage)

$$d(A,B) = \frac{1}{|A||B|}\sum_{i\in A}\sum_{j\in B} d(x_i, y_j)$$

4. Взвешенная средняяя связь (Weighted Average Linkage)
Пусть кластер A получился в результате объединения кластеров q и p. Тогда

$$d(A,B) = \frac{d(p,B) + d(q,B)}{2}$$

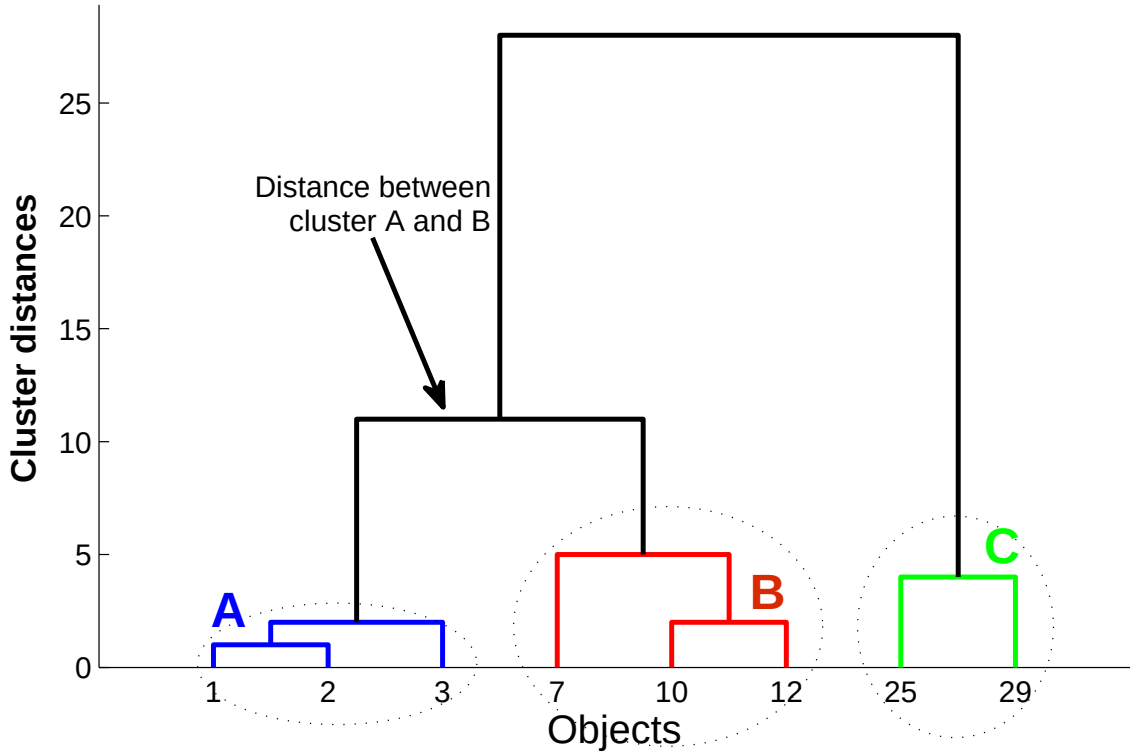
5. Центроидная связь (Centroid Linkage)

$$d(A,B) = \|c_A - c_B\|_2$$

Агломеративная кластеризация

Процесс объединения можно изобразить в виде древовидной структуры – **дендрограммы**.

Пусть даны одномерные наблюдения $\{ 1, \quad 2, \quad 3, \quad 7, \quad 10, \quad 12, \quad 25, \quad 29 \}$

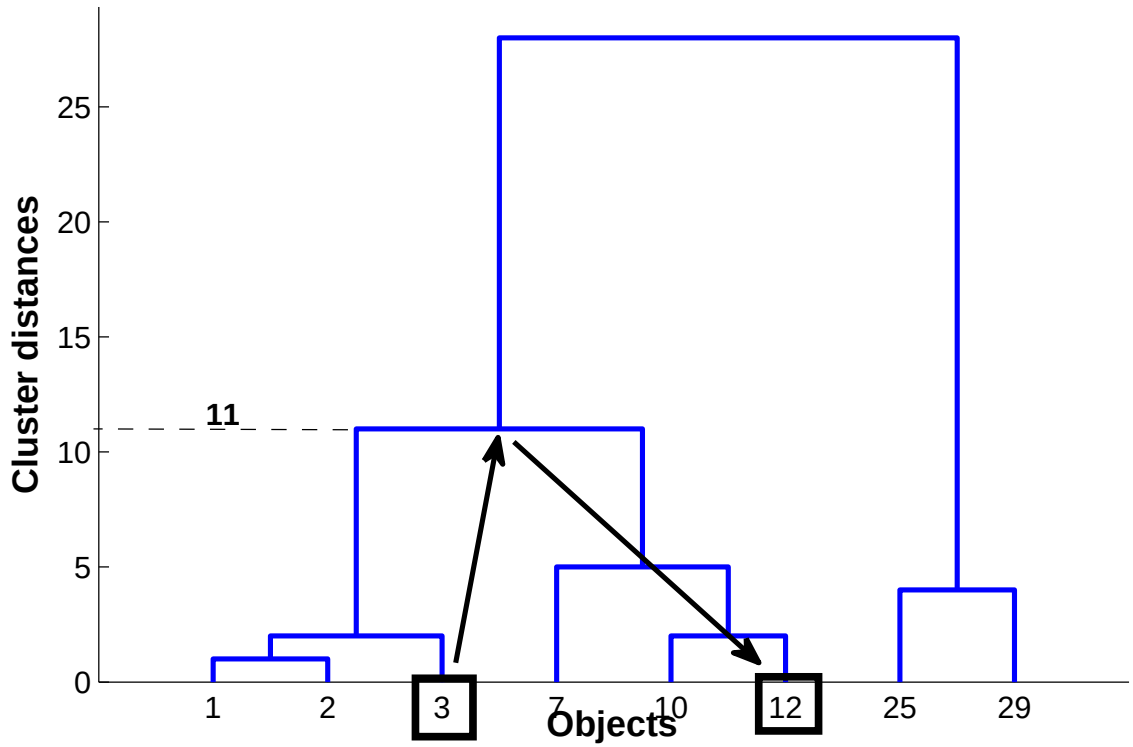


Оценка качества

Кофенетическая корреляция

Кофенетическое расстояние

Кофенетическим расстоянием между объектами x_i и x_j называется высота дерева, при котором эти объекты стали содержаться в одном кластере.



Оценка качества

Кофенетическая корреляция

Кофенетическая корреляция

Кофенетическая корреляция — коэффициент корреляции между рядами попарных расстояний и попарных кофенетических расстояний.
При «удачно» построенном дереве эти ряды должны хорошо коррелировать.

$$copCorr = \frac{\sum_{i<j}(d(x_i, x_j) - \overline{d})(coph(x_i, x_j) - \overline{coph})}{\sqrt{\sum_{i<j}(d(x_i, x_j) - \overline{d})^2 \cdot \sum_{i<j}(coph(x_i, x_j) - \overline{coph})^2}}$$

Дивизивная кластеризация

Дивизивная кластеризация идет в обратном направлении, разбивая большие кластеры на меньшие.

1. Найти объект x_{i*} с наибольшим средним расстоянием от остальных. Добавить его ко множеству S — отсоединённых объектов
2. Для каждого объекта $x_i \notin S$ вычислить разницу между средними расстояниями до объектов из S и не из S:

$$D_i = \frac{1}{|S|}\sum_{j \in S} d(x_i, x_j) - \frac{1}{|\overline{S}|}\sum_{j \notin S} d(x_i, x_j)$$

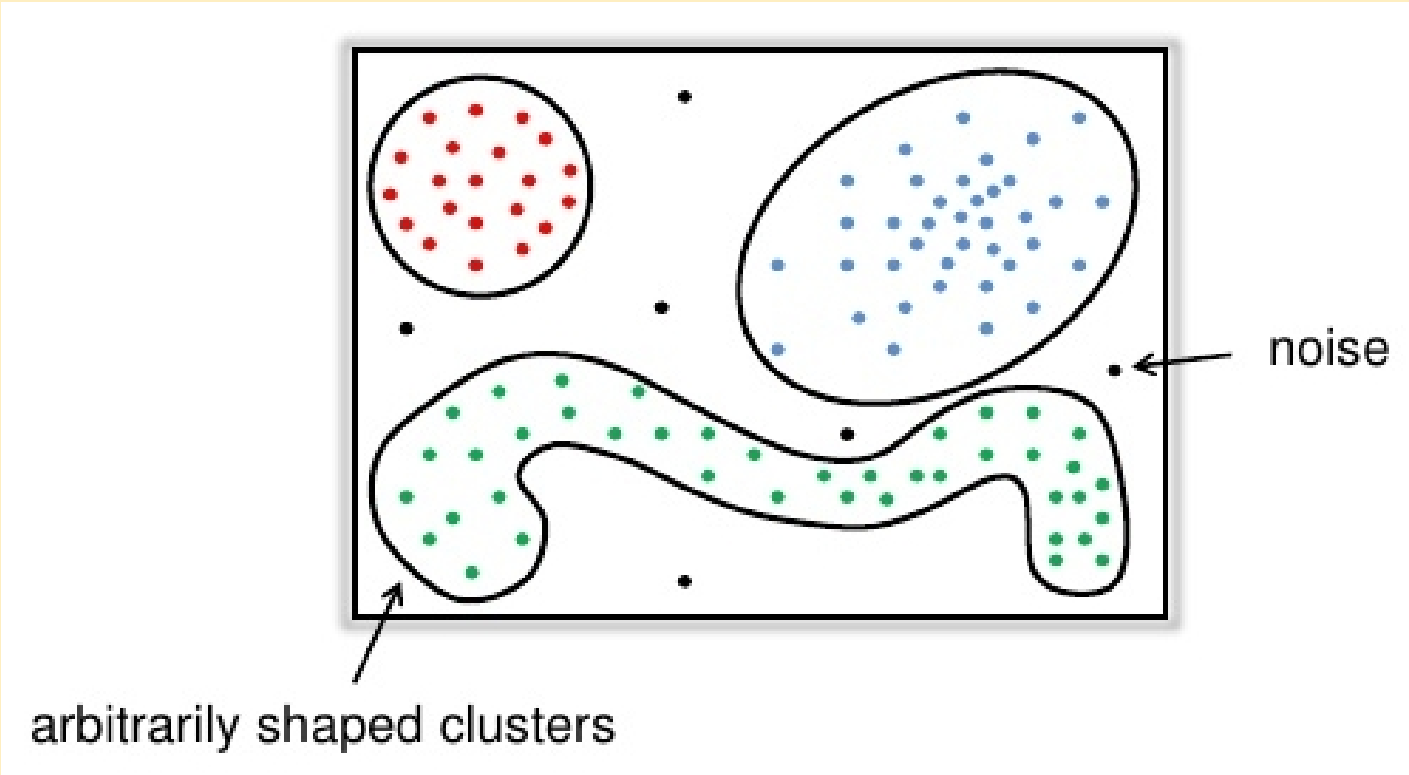
3. Добавить объект x_h с наименьшим D_h к S
4. Повторить шаги 2-3 пока все D_i не окажутся положительными
5. Повторить шаги 1-4 для кластера с наибольшим диаметром (наибольшим расстоянием между парой объектов)

Закончить процесс необходимо, когда останутся только одноэлементные кластеры.

Плотностные методы

Алгоритм DBSCAN

Алгоритм DBSCAN (Density Based Spatial Clustering of Applications with Noise) — плотностный алгоритм для кластеризации пространственных данных с присутствием шума. Способен распознать кластеры различной формы.



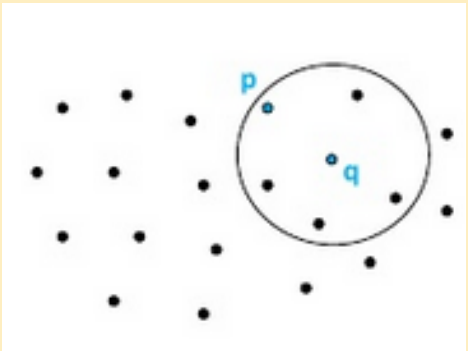
Алгоритм DBSCAN

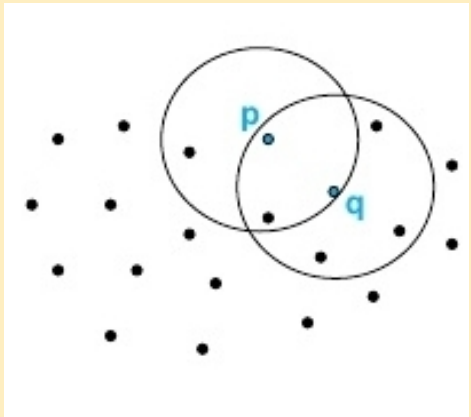
Основная идея

Для каждой точки кластера её окрестность заданного радиуса должно содержать не менее некоторого числа точек M, то есть $N_{eps}(p) \geq M$, где $N_{eps}(p)$ — множество точек, расположенных не далее, чем на расстоянии Eps от p.
Но возникает проблема с граничными точками.

Определение

Точка p непосредственно плотно-достижима из точки q (при заданных Eps и M), если $p \in N_{eps}(q)$ и $|N_{eps}(q)| \geq M$.

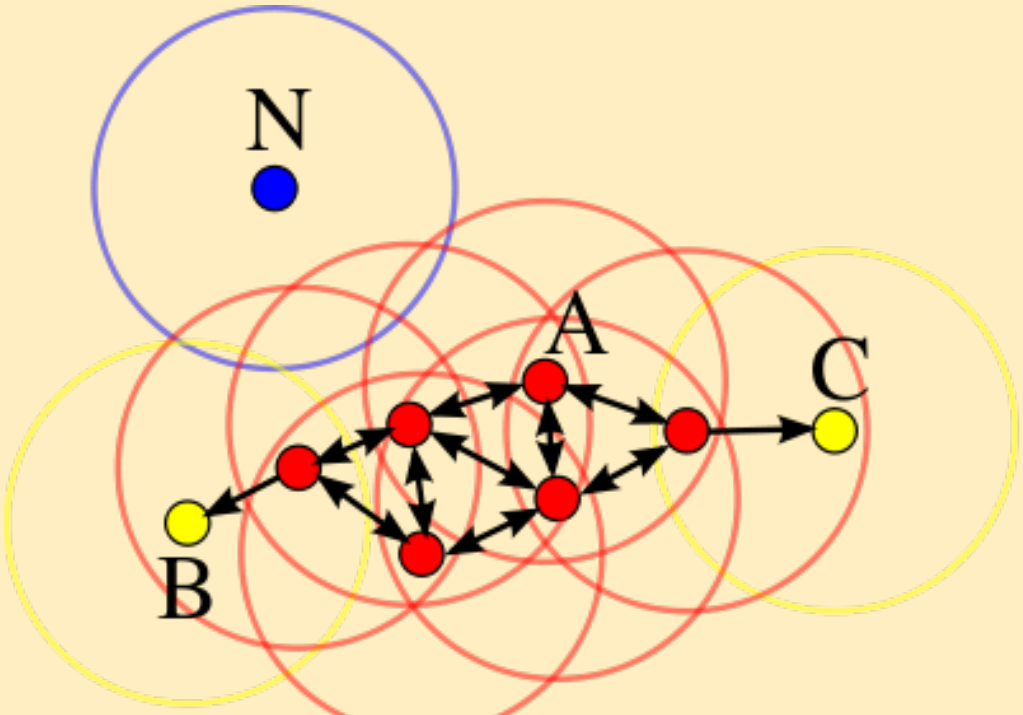




Алгоритм DBSCAN

Определение

Точка p плотно-достижима из точки q (при заданных Eps и M), если между ними существует последовательность точек, таких что каждая непосредственно плотно-достижима из предыдущей.
Точка B плотно-связана (при заданных Eps и M) с точкой C, если существует точка A, такая что B и C плотно-достижимы из A (при заданных Eps и M).



Алгоритм DBSCAN

Определение кластера

Кластер C_j (при заданных Eps и M) — это непустое множество объектов:
1) $\forall p, q : p \in C_j, q$ плотно-достижима (при заданных Eps и M) из точки $p \Rightarrow q \in C_j$
2) $\forall p, q \in C_j : p$ плотно-связана (при заданных Eps и M) с q.

Псевдокод алгоритма

Require: Данные $\mathcal{D}$, Eps, M - параметры.
Ensure: Кластеры C_j.

Устанавливаем всем элементам множества $\mathcal{D}$ флаг «не посещён», $j = 0$, $Noise = \emptyset$
for all $d_i \in \mathcal{D}$: флаг(d_i) == «не посещен» **do**
    флаг(d_i) = «посещен», $N_i = N_{eps}(d_i)$
    if $|N_i| < M$ **then**
        $Noise = Noise + \{d_i\}$ (отсев шумов)
    else
        Номер следующего кластера $j = j + 1$
        Расширяем кластер – Expand(d_i, N_i, C_j, Eps, M)
return $\mathcal{C} = \{C_j\}$

Алгоритм DBSCAN

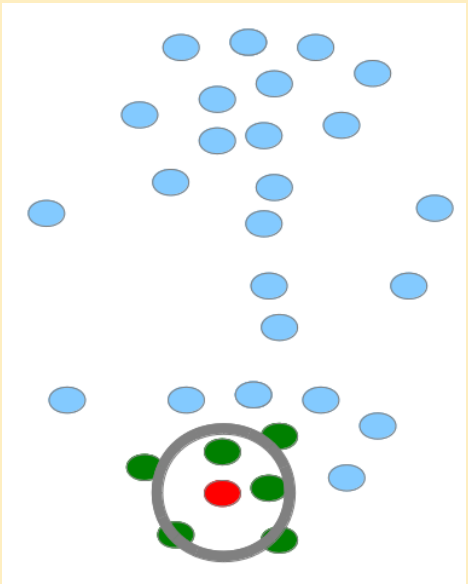
Псевдокод алгоритма. Расширение кластера

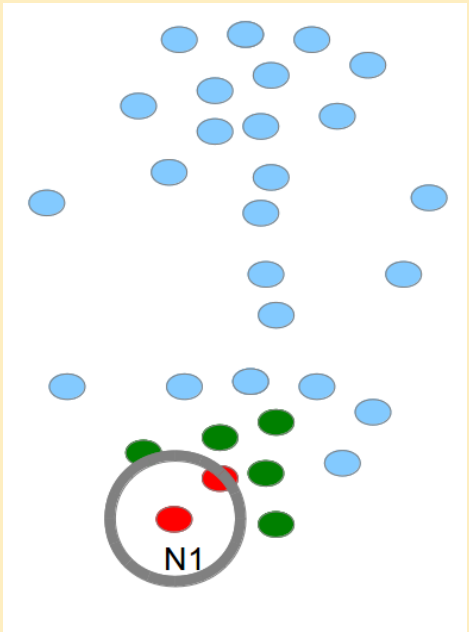
Require: Текущий объект d_i, его окрестность N_i, текущий кластер C_j, Eps, M — параметры
Ensure: Кластер C_j.

$C_j = C_j + \{d_i\}$
for all $d_k \in N_i$: флаг(d_i) == «не посещен» **do**
  if флаг(d_k) == «не посещен» **then**
    флаг(d_k) = «посещен»
    $N_{ik} = N_{eps}(d_k)$
    if $|N_{ik}| \geq M$ **then**
      $N_i = N_i + N_{ik}$
  if $\nexists p : d_k \in C_p$(d_k еще нет ни в одном кластере) **then**
    $C_j = C_j + \{d_k\}$
return $\mathcal{C} = \{C_j\}$

DBSCAN. Пример

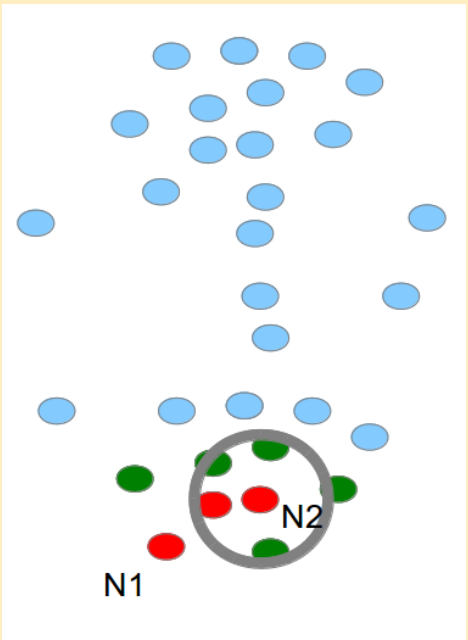
Начальные параметры: $M = 4, Eps > 0$.
Берем наугад первую точку. У нее 6 соседей из N_{eps} (рис. слева) $\Rightarrow$ создаем первый кластер (красный) и начинаем расширение. Первый из соседей $N1$ оказался граничным — добавляем его в кластер (рис. справа).

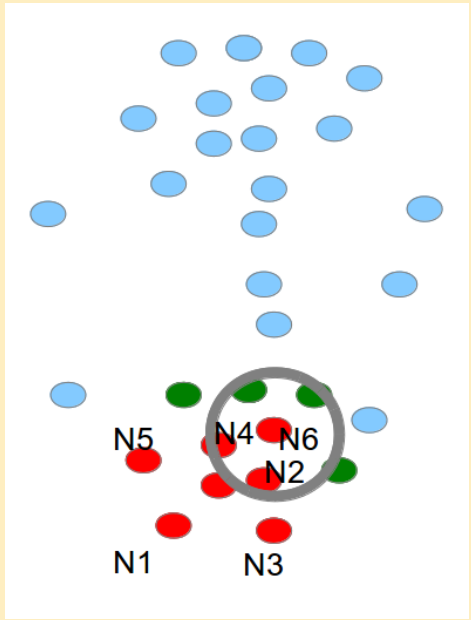




DBSCAN. Пример

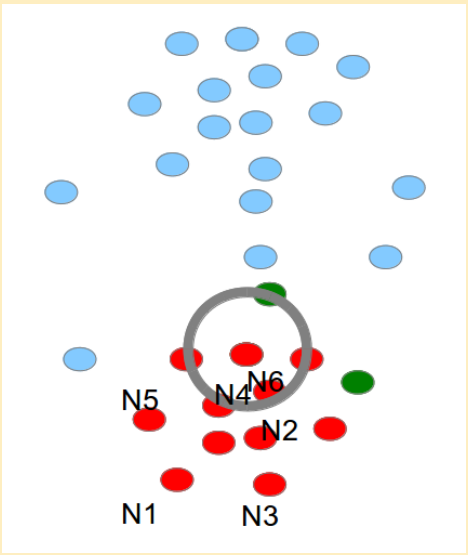
Переходим к следующему соседу $N2$. У него 5 своих соседей из N_{ik}. (рис. слева) $\Rightarrow$ Добавляем новых соседей к старым (появился еще один зеленый сосед). И так далее. Когда обошли всех исходных соседей $N1$—$N6$ (рис. справа), продолжаем с новыми, «зелеными».

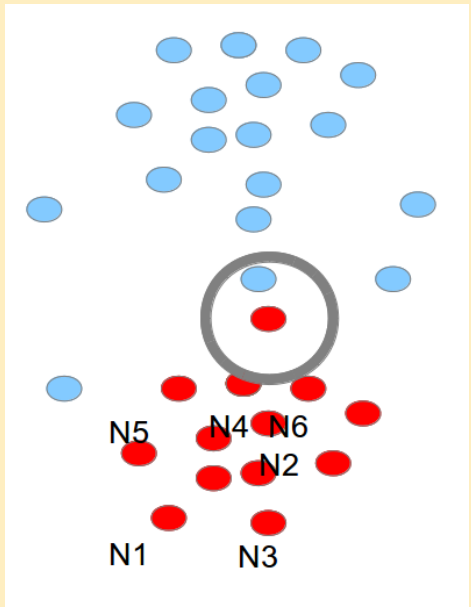




DBSCAN. Пример

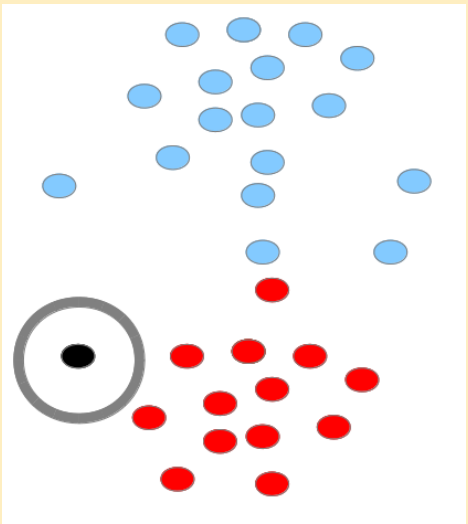
После обхода соседей точек $N1$—$N6$ остаются всего две «зеленые» точки (рис. слева), после обработки которых формируется первый кластер (рис. справа) и далее снова наугад берется точка из изходного массива.

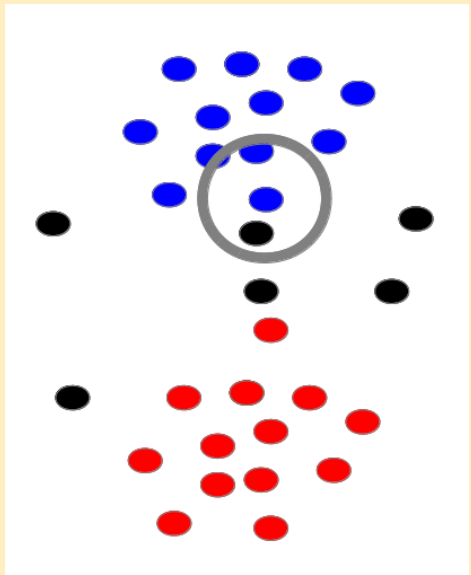




DBSCAN. Пример

Когда выбор пал на «одинокую точку», у которой число соседей меньше $M = 4$ (рис. слева), она добавляется в массив шумов *Noise*, и далее опять наугад выбирается следующая непосещенная точка. В итоге в данном примере формируются 2 кластера, а 6 точек классифицируются как шумы (рис. справа). Заметим, что в число шумов попали и две точки между кластерами («перешеек»).





Плюсы и минусы алгоритма DBSCAN

Плюсы

+ Находит кластеры произвольной формы
+ Алгоритм легко реализовать
+ Различает шумы во входных данных
+ Хорошее быстродействие — $O(n\log(n))$ при правильном выборе структуры данных (в противном случае — $O(n^2)$)

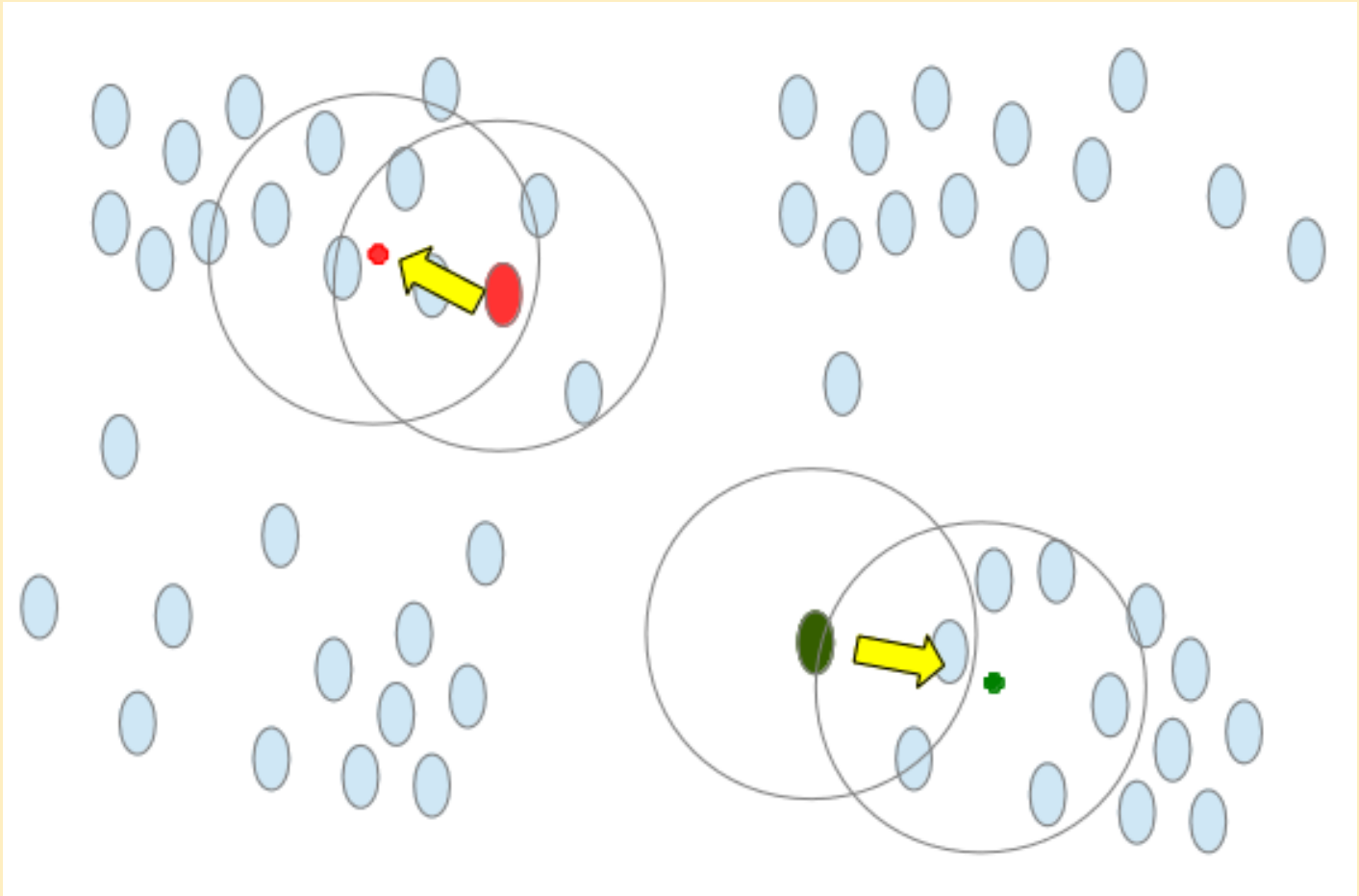
Минусы

- Параметрический. Плохо работает при больших разностях плотности из-за параметра M (минимальное число соседей). Есть модификации алгоритма, учитывающие эту проблему.
- Зависит от выбора метрики расстояния

Метод Mean-Shift

Идея

Найти центр масс объектов, где плотность точек максимальна, и использовать его как центроид.

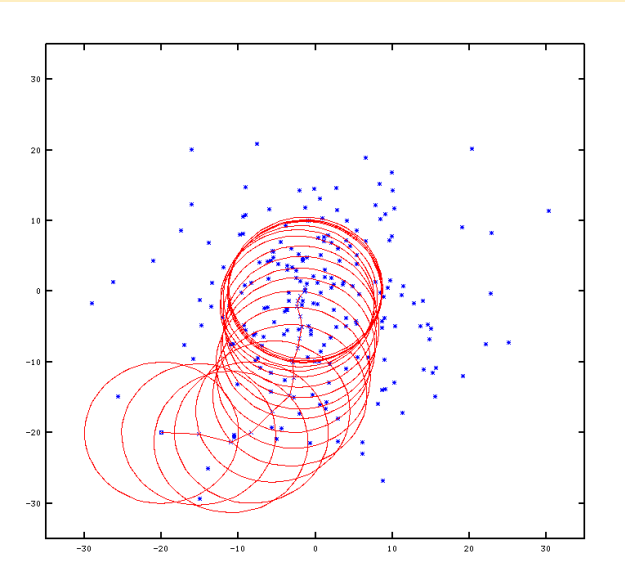


Метод Mean-Shift

Идея

- Задать область вокруг каждой точки выборки
- Вычислить в каждой области центроид
- Переместить центр области в центроид

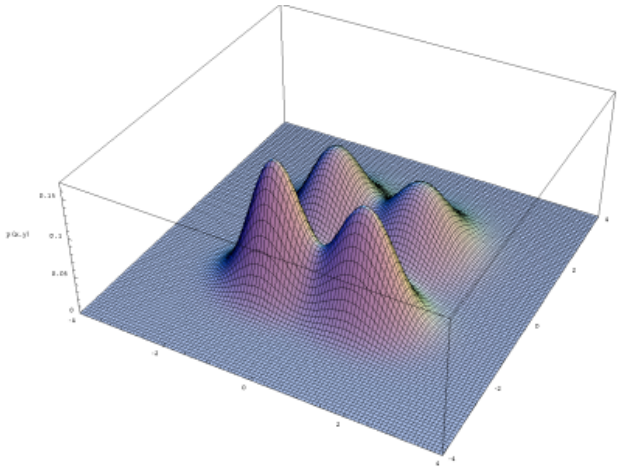
После каждой итерации центроиды перемещаются в «более плотные» области до сходимости к пикам плотности (density modes).



Метод Mean-Shift

Пики плотности (density modes) точек задаются с помощью ядерной оценки плотности (kernel density estimation):

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{\mathbf{x} - \mathbf{x_i}}{h})$$



d – размерность данных, h – ширина окна (bandwidth).
$K(\frac{\mathbf{x}-\mathbf{x_i}}{h})$ – ядро как функция от расстояния задает вклад соседей x_i при вычислении среднего в пределах окна. Часто используется ядро Гаусса:

$$K(\frac{\mathbf{x} - \mathbf{x_i}}{h}) = \frac{1}{2\pi^{d/2}} e^{-\frac{||\mathbf{x}-\mathbf{x_i}||^2}{2h^2}}$$

Метод Mean-Shift

Mean-Shift использует градиентный подъем (gradient ascent).

$$\nabla \hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} \frac{\partial}{\partial \mathbf{x}} K(\frac{\mathbf{x} - \mathbf{x_i}}{h})$$

$$\nabla \hat{f}(\mathbf{x}) = 0$$

Для Гауссового ядра:

$$\frac{\partial}{\partial \mathbf{x}} K(\frac{\mathbf{x} - \mathbf{x_i}}{h}) = K(\frac{\mathbf{x} - \mathbf{x_i}}{h}) \frac{\mathbf{x} - \mathbf{x_i}}{h} \frac{1}{h}$$

$$\Longrightarrow \sum_{i=1}^{n} K(\frac{\mathbf{x} - \mathbf{x_i}}{h})\mathbf{x} = \sum_{i=1}^{n} K(\frac{\mathbf{x} - \mathbf{x_i}}{h})\mathbf{x_i}$$

Метод Mean-Shift

Тогда направление наибольшего роста ядерной функции плотности задается вектором:

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^{n} K(\frac{\mathbf{x}-\mathbf{x_i}}{h})\mathbf{x_i}}{\sum_{i=1}^{n} K(\frac{\mathbf{x}-\mathbf{x_i}}{h})}$$

Собственно, смещение среднего (mean shift):

$$\mathbf{m}(\mathbf{x}) - \mathbf{x} = \frac{\sum_{i=1}^{n} K(\frac{\mathbf{x}-\mathbf{x_i}}{h})\mathbf{x_i}}{\sum_{i=1}^{n} K(\frac{\mathbf{x}-\mathbf{x_i}}{h})} - \mathbf{x}$$

Метод Mean-Shift

Шаги

Вход: Данные D

Выход: Кластеры C_j

1. Вычисление mean shift: Для каждой точки начальной выборки $x_i \in D$ вычисляется вектор смещения среднего $\mathbf{m}(\mathbf{x_i})$

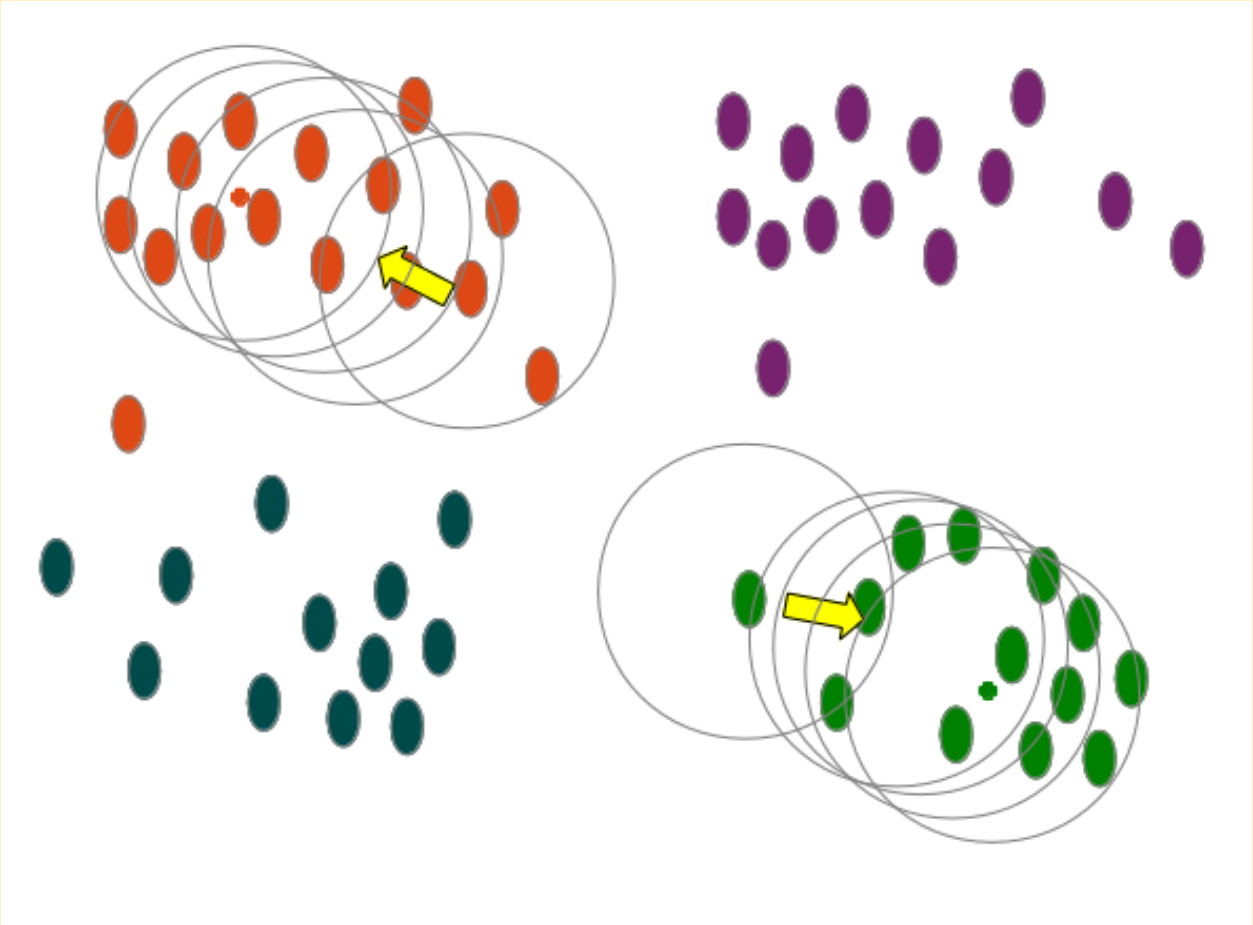
2. Расширение кластера: Аргумент фунцкии ядерной оценки плотности смещается на $\mathbf{m}(\mathbf{x}) : \hat{f}(\mathbf{x}) \to \hat{f}(\mathbf{m}(\mathbf{x}) - \mathbf{x})$.

Шаги 1, 2 повторяются до сходимости к пикам фунцкии ядерной оценки плотности.

Метод Mean-Shift

Сходимость к локальному максимуму гарантирована

Yizong Cheng, Mean Shift, Mode Seeking, and Clustering. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,(8) 1995



Вопросы и контакты

www.hse.ru/staff/dima

Спасибо!

dmitrii.ignatov[at]gmail.com