

# Кластеризация и частичное обучение

К. В. Воронцов  
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса  
<http://www.MachineLearning.ru/wiki>  
«Машинное обучение (курс лекций, К.В.Воронцов)»

2 апреля 2018

## 1 Оптимизационные методы кластеризации

- Задачи кластеризации и частичного обучения
- Метод k-средних и EM-алгоритм
- Сеть Кохонена

## 2 Эвристические методы кластеризации

- Графовые методы и алгоритм FOREL
- Алгоритм DBSCAN
- Иерархические методы

## 3 Частичное обучение на основе классификации

- Обёртки над методами классификации
- Трансдуктивный SVM
- Регуляризация правдоподобия

## Постановка задачи кластеризации

**Дано:**

$X$  — пространство объектов;

$X^\ell = \{x_1, \dots, x_\ell\}$  — обучающая выборка;

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

**Найти:**

$Y$  — множество кластеров,

$a: X \rightarrow Y$  — алгоритм кластеризации,

такие, что:

- каждый кластер состоит из близких объектов;
- объекты разных кластеров существенно различны.

Это задача *обучения без учителя* (unsupervised learning).

## Некорректность задачи кластеризации

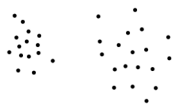
Решение задачи кластеризации принципиально неоднозначно:

- точной постановки задачи кластеризации нет;
- существует много критериев качества кластеризации;
- существует много эвристических методов кластеризации;
- число кластеров  $|Y|$ , как правило, неизвестно заранее;
- результат кластеризации сильно зависит от метрики  $\rho$ , выбор которой также является эвристикой.

## Цели кластеризации

- Упростить дальнейшую обработку данных, разбить множество  $X^\ell$  на группы схожих объектов чтобы работать с каждой группой в отдельности (задачи классификации, регрессии, прогнозирования).
- Сократить объём хранимых данных, оставив по одному представителю от каждого кластера (задачи сжатия данных).
- Выделить нетипичные объекты, которые не подходят ни к одному из кластеров (задачи одноклассовой классификации).
- Построить иерархию множества объектов, пример — классификация животных и растений К.Линнея (задачи таксономии).

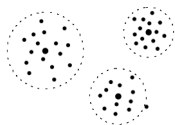
## Типы кластерных структур



внутрикластерные расстояния, как правило,  
меньше межкластерных



ленточные кластеры



кластеры с центром

## Типы кластерных структур



кластеры могут соединяться перемычками



кластеры могут накладываться на разреженный фон из редко расположенных объектов



кластеры могут перекрываться

## Типы кластерных структур



кластеры могут образовываться не по сходству, а по иным типам регулярностей



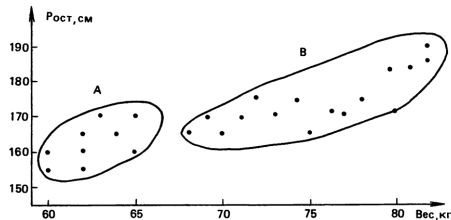
кластеры могут вообще отсутствовать

- Каждый метод кластеризации имеет свои ограничения и выделяет кластеры лишь некоторых типов.
- Понятие «тип кластерной структуры» зависит от метода и также не имеет формального определения.

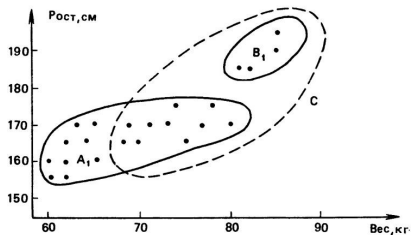


## Проблема чувствительности к выбору метрики

Результат зависит от нормировки признаков:



A — студентки,  
B — студенты



после перенормировки  
(сжали ось «вес» вдвое)

## Постановка задачи частичного обучения

**Дано:**

множество объектов  $X$ , множество классов  $Y$ ;

$X^k = \left\{ \begin{array}{l} x_1, \dots, x_k \\ y_1, \dots, y_k \end{array} \right\}$  — размеченные объекты (labeled data);

$U = \{x_{k+1}, \dots, x_\ell\}$  — неразмеченные объекты (unlabeled data).

**Два варианта постановки задачи:**

- *Частичное обучение* (semi-supervised learning):  
построить алгоритм классификации  $a: X \rightarrow Y$ .
- *Трансдуктивное обучение* (transductive learning):  
зная **все**  $\{x_{k+1}, \dots, x_\ell\}$ , получить метки  $\{a_{k+1}, \dots, a_\ell\}$ .

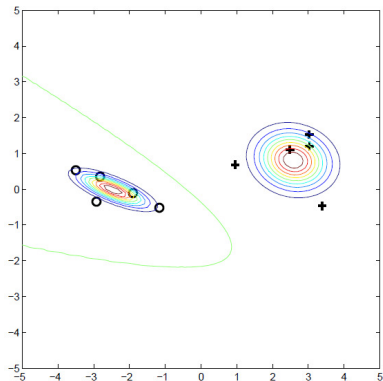
**Типичные приложения:**

классификация и каталогизация текстов, изображений, и т. п.

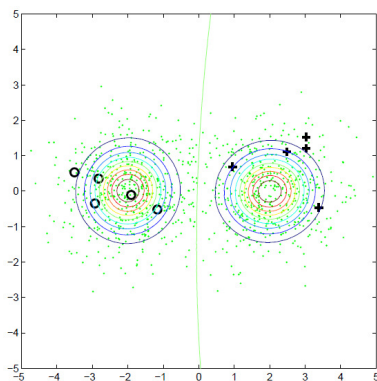
## SSL не сводится к классификации

**Пример 1.** плотности классов, восстановленные:

по размеченным данным  $X^k$

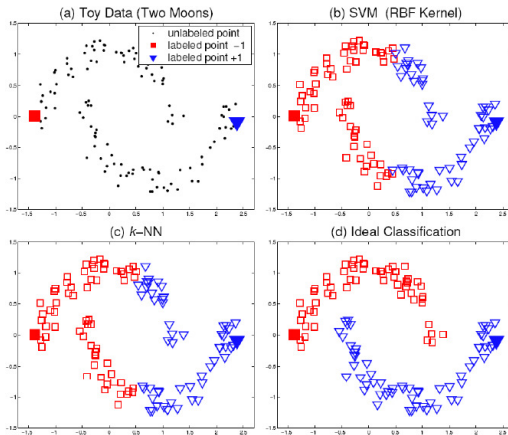


по полным данным  $X^l$



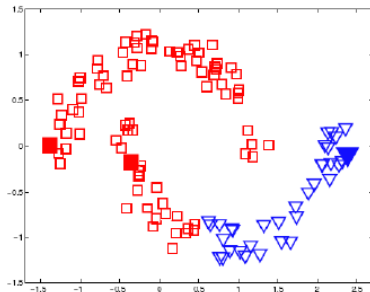
## SSL не сводится к классификации

Пример 2. Методы классификации не учитывают кластерную структуру неразмеченных данных



## Однако и к кластеризации SSL также не сводится

**Пример 3.** Методы кластеризации не учитывают приоритетность разметки над кластерной структурой.



## Качество кластеризации в метрическом пространстве

Пусть известны только попарные расстояния между объектами.

- Среднее внутрикластерное расстояние:

$$F_0 = \frac{\sum_{i < j} [a_i = a_j] \rho(x_i, x_j)}{\sum_{i < j} [a_i = a_j]} \rightarrow \min .$$

- Среднее межкластерное расстояние:

$$F_1 = \frac{\sum_{i < j} [a_i \neq a_j] \rho(x_i, x_j)}{\sum_{i < j} [a_i \neq a_j]} \rightarrow \max .$$

- Отношение пары функционалов:  $F_0/F_1 \rightarrow \min .$

## Качество кластеризации в линейном векторном пространстве

Пусть объекты  $x_i$  задаются векторами  $(f_1(x_i), \dots, f_n(x_i))$ .

- Сумма средних внутрикластерных расстояний:

$$\Phi_0 = \sum_{a \in Y} \frac{1}{|X_a|} \sum_{i: a_i = a} \rho(x_i, \mu_a) \rightarrow \min,$$

$X_a = \{x_i \in X^\ell \mid a_i = a\}$  — кластер  $a$ ,  
 $\mu_a$  — центр масс кластера  $a$ .

- Сумма межкластерных расстояний:

$$\Phi_1 = \sum_{a, b \in Y} \rho(\mu_a, \mu_b) \rightarrow \max.$$

- Отношение пары функционалов:  $\Phi_0/\Phi_1 \rightarrow \min$ .

## Кластеризация как задача дискретной оптимизации

Веса на парах объектов (близости):  $w_{ij} = \exp(-\beta\rho(x_i, x_j))$ ,  
где  $\rho(x, x')$  — расстояние между объектами,  $\beta$  — параметр.

**Задача кластеризации:** найти метки кластеров  $a_i$

$$\sum_{i=1}^{\ell} \sum_{j=i+1}^{\ell} w_{ij} [a_i \neq a_j] \rightarrow \min_{\{a_i \in Y\}} .$$

**Задача частичного обучения:**

$$\lambda \sum_{i=1}^k [a_i \neq y_i] + \sum_{i=k+1}^{\ell-1} \sum_{j=i+1}^{\ell} w_{ij} [a_i \neq a_j] \rightarrow \min_{\{a_i \in Y\}} .$$

где  $\lambda$  — ещё один параметр.



## Метод K-средних (K-means) для кластеризации

Минимизация суммы квадратов внутрикластерных расстояний:

$$\sum_{i=1}^{\ell} \|x_i - \mu_{a_i}\|^2 \rightarrow \min_{\{a_i\}, \{\mu_a\}}, \quad \|x_i - \mu_a\|^2 = \sum_{j=1}^n (f_j(x_i) - \mu_{aj})^2$$

### Алгоритм Ллойда

**Вход:**  $X^\ell$ ,  $K = |Y|$ . **Выход:** центры кластеров  $\mu_a$ ,  $a \in Y$

1:  $\mu_a :=$  начальное приближение центров, для всех  $a \in Y$ ;

2: **повторять**

3: отнести каждый  $x_i$  к ближайшему центру:

$$a_i := \arg \min_{a \in Y} \|x_i - \mu_a\|, \quad i = 1, \dots, \ell;$$

4: вычислить новые положения центров:

$$\mu_a := \frac{\sum_{i=1}^{\ell} [a_i = a] x_i}{\sum_{i=1}^{\ell} [a_i = a]}, \quad a \in Y;$$

5: **пока**  $a_i$  не перестанут изменяться;

## Метод K-средних (K-means) для частичного обучения

### Модификация алгоритма Ллойда

при наличии размеченных объектов  $\{x_1, \dots, x_k\}$

**Вход:**  $X^\ell$ ,  $K = |Y|$ . **Выход:** центры кластеров  $\mu_a$ ,  $a \in Y$

1:  $\mu_a :=$  начальное приближение центров, для всех  $a \in Y$ ;

2: **повторять**

3: отнести каждый  $x_i \in U$  к ближайшему центру:

$$a_i := \arg \min_{a \in Y} \|x_i - \mu_a\|, \quad i = k + 1, \dots, \ell;$$

4: вычислить новые положения центров:

$$\mu_a := \frac{\sum_{i=1}^{\ell} [a_i = a] x_i}{\sum_{i=1}^{\ell} [a_i = a]}, \quad a \in Y;$$

5: **пока**  $a_i$  не перестанут изменяться;

## Метод K-средних — упрощение EM-алгоритма для GMM

**EM-алгоритм:** максимизация правдоподобия для разделения смеси гауссиан (GMM, Gaussian Mixture Model)

1: начальное приближение  $w_a$ ,  $\mu_a$ ,  $\Sigma_a$  для всех  $a \in Y$ ;

2: **повторять**

3: E-шаг: отнести каждый  $x_i$  к ближайшим центрам:

$$g_{ia} := P(a|x_i) \equiv \frac{w_a p_a(x_i)}{\sum_{y \in Y} w_y p_y(x_i)}, \quad a \in Y, \quad i = 1, \dots, \ell;$$

$$a_i := \arg \max_{a \in Y} g_{ia}, \quad i = 1, \dots, \ell;$$

4: M-шаг: вычислить новые положения центров:

$$\mu_{ad} := \frac{1}{\ell w_a} \sum_{i=1}^{\ell} g_{ia} f_d(x_i), \quad a \in Y, \quad d = 1, \dots, n;$$

$$\sigma_{ad}^2 := \frac{1}{\ell w_a} \sum_{i=1}^{\ell} g_{ia} (f_d(x_i) - \mu_{ad})^2, \quad a \in Y, \quad d = 1, \dots, n;$$

$$w_a := \frac{1}{\ell} \sum_{i=1}^{\ell} g_{ia}, \quad a \in Y;$$

5: **пока**  $a_i$  не перестанут изменяться;

## Сравнение EM-алгоритма для GMM и метода $k$ -средних

### Основные отличия GMM-EM и $k$ -means:

- GMM-EM: мягкая кластеризация:  $g_{ia} = P(a_i = a)$   
 $k$ -means: жёсткая кластеризация:  $g_{ia} = [a_i = a]$
- GMM-EM: кластеры эллиптические, настраиваемые  
 $k$ -means: кластеры сферические, не настраиваемые

### Гибриды (упрощение GMM-EM — усложнение $k$ -means):

- GMM-EM с жёсткой кластеризацией на E-шаге
- GMM-EM без настройки дисперсий (сферические гауссианы)

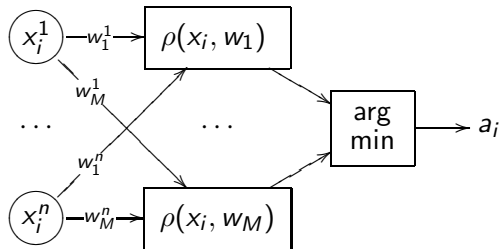
### Недостатки $k$ -means:

- чувствительность к выбору начального приближения
- медленная сходимость (пользуйтесь  $k$ -means++)

## Сеть Кохонена (сеть с конкурентным обучением)

Объект  $x_i \in \mathbb{R}^n$  относится к ближайшему центру  $w_a \in \mathbb{R}^n$   
 (правило жёсткой конкуренции WTA — Winner Takes All):

$$a_i = \arg \min_{a \in Y} \rho(x_i, w_a), \quad Y = \{1, \dots, M\}$$



Для поиска центров  $w_a$  будем минимизировать сумму  
 внутрикластерных расстояний, аналогично методу  $k$ -средних.

## Оптимизационная задача и метод стохастического градиента

Минимизация суммы внутрикластерных расстояний:

$$Q(w; X^\ell) = \frac{1}{2} \sum_{i=1}^{\ell} \rho^2(x_i, w_{a_i}) \rightarrow \min_{\{w_a\}, \{a_i\}} .$$

Пусть метрика евклидова,  $\rho^2(x_i, w_a) = \|w_a - x_i\|^2$ .

$$\frac{\partial Q(w; X^\ell)}{\partial w_a} = \sum_{i=1}^{\ell} (w_a - x_i) [a_i = a] .$$

Градиентный шаг в методе SG: для случайного  $x_i \in X^\ell$

$$w_a := w_a + \eta(x_i - w_a) [a_i = a]$$

(если  $x_i$  относится к кластеру  $a$ , то  $w_a$  сдвигается в сторону  $x_i$ ).

## Алгоритм SG (Stochastic Gradient)

**Вход:** выборка  $X^\ell$ ; темп обучения  $\eta$ ; параметр  $\lambda$ ;

**Выход:** центры  $w_a \in \mathbb{R}^n$ ,  $a \in Y$ ; кластеризация  $\{a_i\}_{i=1}^\ell$ ;

- 
- 1: инициализировать центры  $w_a$ ,  $a \in Y$ ;
  - 2: инициализировать текущую оценку функционала:  

$$Q := \frac{1}{|U|} \sum_{x_i \in U} \rho^2(x_i, w_{a_i})$$
 по случайной подвыборке  $U \subseteq X^\ell$ ;
  - 3: **повторять**
  - 4: выбрать объект  $x_i$  из  $X^\ell$  (например, случайно);
  - 5: вычислить кластеризацию:  $a_i := \arg \min_{a \in Y} \rho(x_i, w_a)$ ;
  - 6: **градиентный шаг:**  $w_{a_i} := w_{a_i} + \eta(x_i - w_{a_i})$ ;
  - 7: оценить значение функционала:  

$$Q := (1 - \lambda)Q + \lambda \rho^2(x_i, w_{a_i})$$
;
  - 8: **пока** значение  $Q$  и/или веса  $w$  не стабилизируются;

## Жёсткая и мягкая конкуренция

Правило жёсткой конкуренции WTA (winner takes all):

$$w_a := w_a + \eta(x_i - w_a) [a_i = a], \quad a \in Y,$$

Недостатки правила WTM:

- медленная скорость сходимости;
- некоторые  $w_a$  могут никогда не выбираться.

Правило мягкой конкуренции WTM (winner takes most):

$$w_a := w_a + \eta(x_i - w_a) K(\rho(x_i, w_a)), \quad a \in Y,$$

где ядро  $K(\rho)$  — неотрицательная невозрастающая функция.

Теперь центры всех кластеров смещаются в сторону  $x_i$ , но чем дальше от  $x_i$ , тем меньше величина смещения.



## Карта Кохонена (Self Organizing Map, SOM)

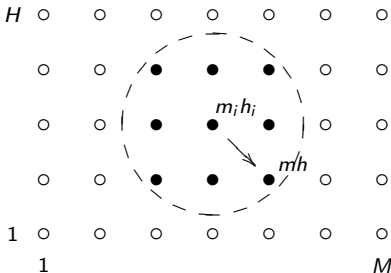
$Y = \{1, \dots, M\} \times \{1, \dots, H\}$  — прямоугольная сетка кластеров.

Каждому узлу  $(m, h)$  приписан нейрон Кохонена  $w_{mh} \in \mathbb{R}^n$ .

Наряду с метрикой  $\rho(x_i, x)$  на  $X$  вводится метрика на сетке  $Y$ :

$$r((m_i, h_i), (m, h)) = \sqrt{(m - m_i)^2 + (h - h_i)^2}.$$

Окрестность  $(m_i, h_i)$ :



## Обучение карты Кохонена

**Вход:**  $X^\ell$  — обучающая выборка;  $\eta$  — темп обучения;

**Выход:**  $w_{mh} \in \mathbb{R}^n$  — векторы весов,  $m = 1..M$ ,  $h = 1..H$ ;

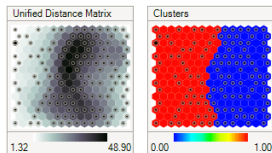
---

- 1:  $w_{mh} := \text{random} \left( -\frac{1}{2MH}, \frac{1}{2MH} \right)$  — инициализация весов;
- 2: **повторять**
- 3: выбрать объект  $x_i$  из  $X^\ell$  случайным образом;
- 4: WTA: вычислить координаты кластера:  
$$(m_i, h_i) := \arg \min_{(m,h) \in Y} \rho(x_i, w_{mh});$$
- 5: **для всех**  $(m, h) \in \text{Окрестность}(m_i, h_i)$
- 6: WTM: сделать шаг градиентного спуска:  
$$w_{mh} := w_{mh} + \eta(x_i - w_{mh}) K(r((m_i, h_i), (m, h)));$$
- 7: **пока** кластеризация не стабилизируется;

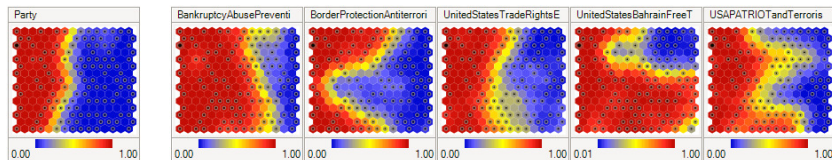
## Интерпретация карт Кохонена: два типа $M \times H$ -графиков

**Пример:** задача UCI house-votes (US Congress voting patterns),  
 объекты — конгрессмены, признаки — результаты голосования

1. Цвет  $(m, h)$  — среднее расстояние до  $k$  ближайших точек



2. Цвет  $(m, h)$  — значение  $j$ -й компоненты вектора  $w_{mh}$ .



## Достоинства и недостатки карт Кохонена

### Достоинства:

- Возможность визуального анализа многомерных данных.

### Недостатки:

- **Субъективность.** Карта зависит не только от кластерной структуры данных, но и от...
  - свойств сглаживающего ядра;
  - (случайной) инициализации;
  - (случайного) выбора  $x_i$  на каждой итерации.
- **Искажения.** Близкие объекты исходного пространства могут переходить в далёкие точки на карте, и наоборот.

Рекомендуется только для разведочного анализа данных.

## Алгоритм КНП для кластеризации

### Графовый алгоритм КНП (кратчайший незамкнутый путь)

- 1: Найти пару вершин  $(x_i, x_j) \in X^\ell$  с наименьшим  $\rho(x_i, x_j)$  и соединить их ребром;
- 2: **пока** в выборке остаются изолированные точки
- 3: найти изолированную точку, ближайшую к некоторой неизолированной;
- 4: соединить эти две точки ребром;
- 5: удалить  $K - 1$  самых длинных рёбер;

### Ограничения алгоритма:

- необходимость задавать число кластеров  $K$
- высокая чувствительность к шуму

## Алгоритм КНП для частичного обучения

### Графовый алгоритм КНП (кратчайший незамкнутый путь)

- 1: Найти пару вершин  $(x_i, x_j) \in X^\ell$  с наименьшим  $\rho(x_i, x_j)$  и соединить их ребром;
- 2: **пока** в выборке остаются изолированные точки
- 3: найти изолированную точку, ближайшую к некоторой неизолированной;
- 4: соединить эти две точки ребром;
- 5: ~~удалить  $K-1$  самых длинных рёбер;~~
- 6: **пока** есть путь между двумя вершинами разных классов
- 7: удалить самое длинное ребро на этом пути.

Задача частичного обучения: заменяется только шаг 5...

## Алгоритм кластеризации FOREL (ФОРмальные Элементы)

- 1:  $U := X^\ell$  — множество некластеризованных точек;
- 2: **пока** в выборке есть некластеризованные точки,  $U \neq \emptyset$ ;
- 3: взять случайную точку  $x_0 \in U$ ;
- 4: **повторять**
- 5: образовать кластер с центром в  $x_0$  и радиусом  $R$ :  
$$K_0 := \{x_i \in U \mid \rho(x_i, x_0) \leq R\};$$
- 6: переместить центр  $x_0$  в центр масс кластера:  
$$x_0 := \frac{1}{|K_0|} \sum_{x_i \in K_0} x_i;$$
- 7: **пока** состав кластера  $K_0$  не стабилизируется;
- 8:  $U := U \setminus K_0$ ;
- 9: применить алгоритм КНП к множеству центров кластеров;
- 10: каждый  $x_i \in X^\ell$  приписать кластеру с ближайшим центром;

---

Ёлкина В.Н., Ёлкин Е.А. Загоруйко Н.Г. О применении методики распознавания образов к решению задач палеонтологии. 1967.

## Алгоритм кластеризации FOREL (ФОРмальные Элементы)

### Замечание к шагу 6:

если  $X$  не является линейным векторным пространством, то

$$x_0 := \frac{1}{|K_0|} \sum_{x_i \in K_0} x_i \quad \longrightarrow \quad x_0 := \arg \min_{x \in K_0} \sum_{x' \in K_0} \rho(x, x');$$

### Преимущества FOREL:

- получаем двухуровневую структуру кластеров;
- кластеры могут быть произвольной формы;
- варьируя  $R$ , можно управлять детальностью кластеризации.

### Недостаток FOREL:

- чувствительность к  $R$  и начальному выбору точки  $x_0$ .

**Устранение:** сгенерировать несколько кластеризаций и выбрать лучшую по *критерию качества кластеризации*.



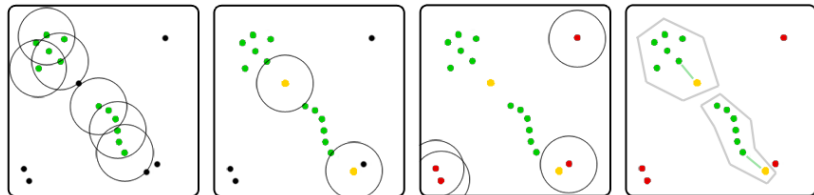
# Алгоритм кластеризации DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

Объект  $x \in U$ , его  $\varepsilon$ -окрестность  $U_\varepsilon(x) = \{u \in U: \rho(x, u) \leq \varepsilon\}$

Каждый объект может быть одного из трёх типов:

- **корневой**: имеющий плотную окрестность,  $|U_\varepsilon(x)| \geq m$
- **граничный**: не корневой, но в окрестности корневого
- **шумовой (выброс)**: не корневой и не граничный



*Ester, Kriegel, Sander, Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. KDD-1996.*

## Алгоритм кластеризации DBSCAN

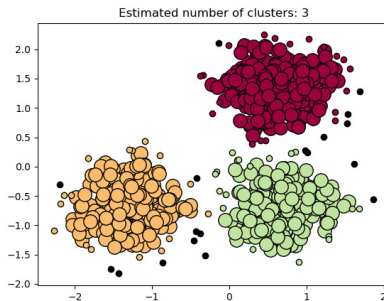
**Вход:** выборка  $X^\ell = \{x_1, \dots, x_\ell\}$ , параметры  $\varepsilon$  и  $m$ ;

**Выход:** разбиение выборки на кластеры и шумовые выбросы  $N$ ;

- 
- 1:  $U := X^\ell$  — некластеризованные;  $N := \emptyset$ ;  $a := 0$ ;
  - 2: **пока** в выборке есть некластеризованные точки,  $U \neq \emptyset$ ;
  - 3:   взять случайную точку  $x \in U$ ;
  - 4:   **если**  $|U_\varepsilon(x)| < m$  **то**
  - 5:     позначить  $x$  как, возможно, шумовой;
  - 6:   **иначе**
  - 7:     создать новый кластер:  $K := U_\varepsilon(x)$ ;  $a := a + 1$ ;
  - 8:     **для всех**  $x' \in K$
  - 9:       **если**  $|U_\varepsilon(x')| \geq m$  **то**  $K := K \cup U_\varepsilon(x')$ ;
  - 10:      **иначе** позначить  $x'$  как граничный кластера  $K$ ;
  - 11:      $a_i := a$  для всех  $x_i \in K$ ;
  - 12:      $U := U \setminus K$ ;

## Преимущества алгоритма DBSCAN

- быстрая кластеризация больших данных:  
 $O(\ell^2)$  в худшем случае,  
 $O(\ell \ln \ell)$  при эффективной реализации  $U_\varepsilon(x)$ ;
- кластеры произвольной формы (долой центры!);
- деление объектов на корневые, граничные, шумовые.



## Агломеративная иерархическая кластеризация

Алгоритм иерархической кластеризации (Ланс, Уильямс, 1967):  
итеративный пересчёт расстояний  $R_{UV}$  между кластерами  $U, V$ .

- 1:  $C_1 := \{\{x_1\}, \dots, \{x_\ell\}\}$  — все кластеры 1-элементные;  
 $R_{\{x_i\}\{x_j\}} := \rho(x_i, x_j)$  — расстояния между ними;
- 2: **для всех**  $t = 2, \dots, \ell$  ( $t$  — номер итерации):
- 3: найти в  $C_{t-1}$  пару кластеров  $(U, V)$  с минимальным  $R_{UV}$ ;
- 4: слить их в один кластер:  
 $W := U \cup V$ ;  
 $C_t := C_{t-1} \cup \{W\} \setminus \{U, V\}$ ;
- 5: **для всех**  $S \in C_t$
- 6: вычислить  $R_{WS}$  по формуле Ланса-Уильямса:  
 $R_{WS} := \alpha_U R_{US} + \alpha_V R_{VS} + \beta R_{UV} + \gamma |R_{US} - R_{VS}|$ ;

## Алгоритм Ланса-Уильямса для частичного обучения

Алгоритм иерархической кластеризации (Ланс, Уильямс, 1967):  
итеративный пересчёт расстояний  $R_{UV}$  между кластерами  $U, V$ .

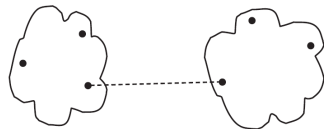
- 1:  $C_1 := \{\{x_1\}, \dots, \{x_\ell\}\}$  — все кластеры 1-элементные;  
 $R_{\{x_i\}\{x_j\}} := \rho(x_i, x_j)$  — расстояния между ними;
- 2: **для всех**  $t = 2, \dots, \ell$  ( $t$  — номер итерации):
- 3: найти в  $C_{t-1}$  пару кластеров  $(U, V)$  с минимальным  $R_{UV}$ ,  
**при условии, что в  $U \cup V$  нет объектов с разными метками;**
- 4: слить их в один кластер:  
 $W := U \cup V$ ;  
 $C_t := C_{t-1} \cup \{W\} \setminus \{U, V\}$ ;
- 5: **для всех**  $S \in C_t$
- 6: вычислить  $R_{WS}$  по формуле Ланса-Уильямса:  
 $R_{WS} := \alpha_U R_{US} + \alpha_V R_{VS} + \beta R_{UV} + \gamma |R_{US} - R_{VS}|$ ;

## Частные случаи формулы Ланса-Уильямса

1. Расстояние ближнего соседа:

$$R_{WS}^b = \min_{w \in W, s \in S} \rho(w, s);$$

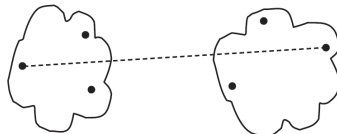
$$\alpha_U = \alpha_V = \frac{1}{2}, \quad \beta = 0, \quad \gamma = -\frac{1}{2}.$$



2. Расстояние дальнего соседа:

$$R_{WS}^d = \max_{w \in W, s \in S} \rho(w, s);$$

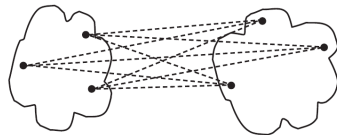
$$\alpha_U = \alpha_V = \frac{1}{2}, \quad \beta = 0, \quad \gamma = \frac{1}{2}.$$



3. Групповое среднее расстояние:

$$R_{WS}^g = \frac{1}{|W||S|} \sum_{w \in W} \sum_{s \in S} \rho(w, s);$$

$$\alpha_U = \frac{|U|}{|W|}, \quad \alpha_V = \frac{|V|}{|S|}, \quad \beta = \gamma = 0.$$



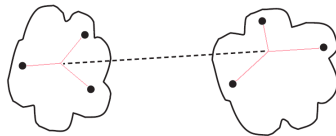
## Частные случаи формулы Ланса-Уильямса

### 4. Расстояние между центрами:

$$R_{WS}^U = \rho^2 \left( \sum_{w \in W} \frac{w}{|W|}, \sum_{s \in S} \frac{s}{|S|} \right);$$

$$\alpha_U = \frac{|U|}{|W|}, \quad \alpha_V = \frac{|V|}{|W|},$$

$$\beta = -\alpha_U \alpha_V, \quad \gamma = 0.$$



### 5. Расстояние Уорда:

$$R_{WS}^Y = \frac{|S||W|}{|S|+|W|} \rho^2 \left( \sum_{w \in W} \frac{w}{|W|}, \sum_{s \in S} \frac{s}{|S|} \right);$$

$$\alpha_U = \frac{|S|+|U|}{|S|+|W|}, \quad \alpha_V = \frac{|S|+|V|}{|S|+|W|}, \quad \beta = \frac{-|S|}{|S|+|W|}, \quad \gamma = 0.$$

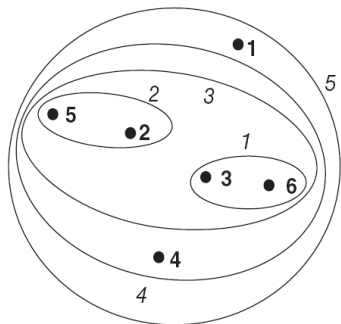
## Проблема выбора

Какая функция расстояния лучше?

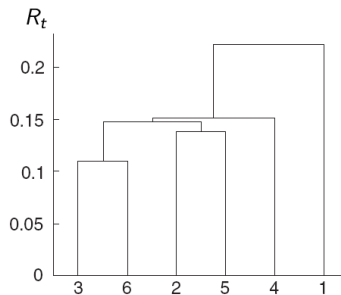
## Визуализация кластерной структуры

### 1. Расстояние ближнего соседа:

Диаграмма вложения



Дендрограмма

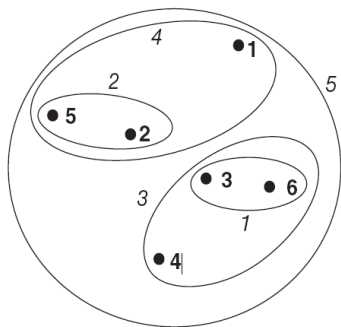




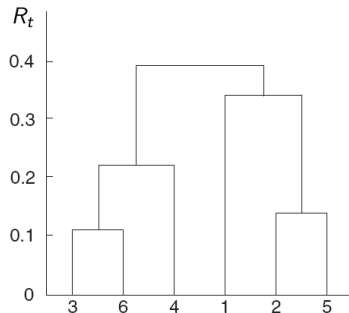
## Визуализация кластерной структуры

### 2. Расстояние дальнего соседа:

Диаграмма вложения



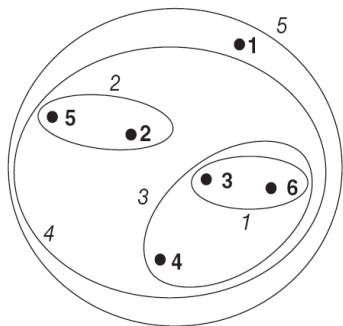
Дендрограмма



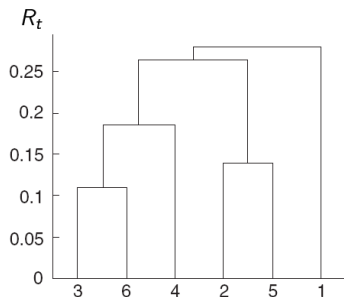
## Визуализация кластерной структуры

### 3. Групповое среднее расстояние:

Диаграмма вложения



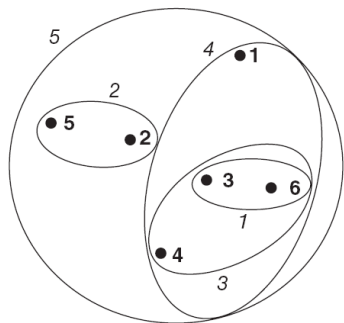
Дендрограмма



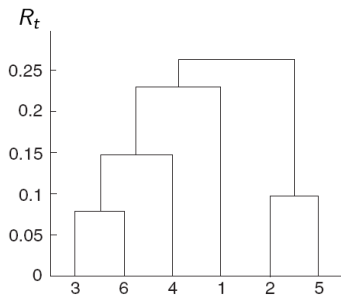
## Визуализация кластерной структуры

### 5. Расстояние Уорда:

Диаграмма вложения



Дендрограмма



## Основные свойства иерархической кластеризации

- *Монотонность*: дендрограмма не имеет самопересечений, при каждом слиянии расстояние между объединяемыми кластерами только увеличивается:  $R_2 \leq R_3 \leq \dots \leq R_\ell$ .
- *Сжимающее расстояние*:  $R_t \leq \rho(\mu_U, \mu_V), \forall t$ .
- *Растягивающее расстояние*:  $R_t \geq \rho(\mu_U, \mu_V), \forall t$

### Теорема (Миллиган, 1979)

Кластеризация монотонна, если выполняются условия

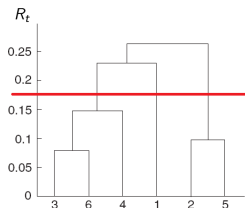
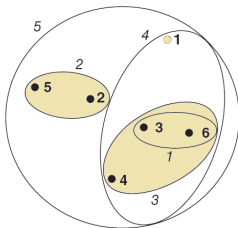
$$\alpha_U \geq 0, \alpha_V \geq 0, \alpha_U + \alpha_V + \beta \geq 1, \min\{\alpha_U, \alpha_V\} + \gamma \geq 0.$$

$R^C$  не монотонно;  $R^b$ ,  $R^d$ ,  $R^g$ ,  $R^y$  — монотонны.

$R^b$  — сжимающее;  $R^d$ ,  $R^y$  — растягивающие;

## Рекомендации и выводы

- рекомендуется пользоваться расстоянием Уорда  $R^y$ ;
- обычно строят несколько вариантов и выбирают лучший визуально по дендрограмме;
- определение числа кластеров — по максимуму  $|R_{t+1} - R_t|$ , тогда результирующее множество кластеров  $:= C_t$ .



## Метод частичного обучения self-training (1965-1970)

Пусть  $\mu: X^k \rightarrow a$  — метод обучения классификации;  
классификаторы имеют вид  $a(x) = \arg \max_{y \in Y} \Gamma_y(x)$ ;

*Псевдоотступ* — степень уверенности классификации  $a_i = a(x_i)$ :

$$M_i(a) = \Gamma_{a_i}(x_i) - \max_{y \in Y \setminus a_i} \Gamma_y(x_i).$$

**Алгоритм self-training** — обёртка (wrapper) над методом  $\mu$ :

- 1:  $Z := X^k$ ;
- 2: **пока**  $|Z| < \ell$
- 3:  $a := \mu(Z)$ ;
- 4:  $\Delta := \{x_i \in U \setminus Z \mid M_i(a) \geq M_0\}$ ;
- 5:  $a_i := a(x_i)$  для всех  $x_i \in \Delta$ ;
- 6:  $Z := Z \cup \Delta$ ;

$M_0$  можно определять, например, из условия  $|\Delta| = 0.05 |U|$

## Метод частичного обучения co-training (Blum, Mitchell, 1998)

Пусть  $\mu_1: X^k \rightarrow a_1$ ,  $\mu_2: X^k \rightarrow a_2$  — два существенно различных метода обучения, использующих

- либо разные наборы признаков;
- либо разные парадигмы обучения (inductive bias);
- либо разные источники данных  $X_1^{k_1}$ ,  $X_2^{k_2}$ .

1:  $Z_1 := X_1^{k_1}$ ;  $Z_2 := X_2^{k_2}$ ;

2: **пока**  $|Z_1 \cup Z_2| < \ell$

3:  $a_1 := \mu_1(Z_1)$ ;  $\Delta_1 := \{x_i \in U \setminus Z_1 \setminus Z_2 \mid M_i(a_1) \geq M_{01}\}$ ;

4:  $a_i := a_1(x_i)$  для всех  $x_i \in \Delta_1$ ;

5:  $Z_2 := Z_2 \cup \Delta_1$ ;

6:  $a_2 := \mu_2(Z_2)$ ;  $\Delta_2 := \{x_i \in U \setminus Z_1 \setminus Z_2 \mid M_i(a_2) \geq M_{02}\}$ ;

7:  $a_i := a_2(x_i)$  для всех  $x_i \in \Delta_2$ ;

8:  $Z_1 := Z_1 \cup \Delta_2$ ;

## Метод частичного обучения co-learning (deSa, 1993)

Пусть  $\mu_t: X^k \rightarrow a_t$  — разные методы обучения,  $t = 1, \dots, T$ .

**Алгоритм co-learning** — это self-training для композиции — простого голосования базовых алгоритмов  $a_1, \dots, a_T$ :

$$a(x) = \arg \max_{y \in Y} \Gamma_y(x), \quad \Gamma_y(x_i) = \sum_{t=1}^T [a_t(x_i) = y].$$

тогда  $M_i(a)$  — степень уверенности классификации  $a(x_i)$ .

- 1:  $Z := X^k$ ;
- 2: **пока**  $|Z| < \ell$
- 3:  $a := \mu(Z)$ ;
- 4:  $\Delta := \{x_i \in U \setminus Z \mid M_i(a) \geq M_0\}$ ;
- 5:  $a_i := a(x_i)$  для всех  $x_i \in \Delta$ ;
- 6:  $Z := Z \cup \Delta$ ;



## Напоминание: SVM для двухклассовой классификации

Линейный классификатор на два класса  $Y = \{-1, 1\}$ :

$$a(x) = \text{sign}(\langle w, x \rangle - w_0), \quad w, x \in \mathbb{R}^n, \quad w_0 \in \mathbb{R}.$$

Отступ объекта  $x_i$ :

$$M_i(w, w_0) = (\langle w, x_i \rangle - w_0) y_i.$$

Задача обучения весов  $w, w_0$  по размеченной выборке:

$$Q(w, w_0) = \sum_{i=1}^k (1 - M_i(w, w_0))_+ + \frac{1}{2C} \|w\|^2 \rightarrow \min_{w, w_0}.$$

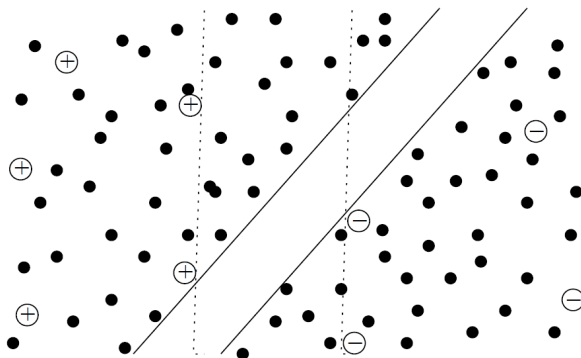
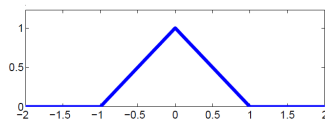
Функция  $\mathcal{L}(M) = (1 - M)_+$  штрафует за уменьшение отступа.

**Идея!**

Функция  $\mathcal{L}(M) = (1 - |M|)_+$  штрафует за попадание объекта внутрь разделяющей полосы.

## Функция потерь для трансдуктивного SVM

Функция потерь  $\mathcal{L}(M) = (1 - |M|)_+$  штрафует за попадание объекта внутрь разделяющей полосы.



## Метод частичного обучения Transductive SVM

Обучение весов  $w, w_0$  по частично размеченной выборке:

$$Q(w, w_0) = \sum_{i=1}^k (1 - M_i(w, w_0))_+ + \frac{1}{2C} \|w\|^2 + \\ + \gamma \sum_{i=k+1}^{\ell} (1 - |M_i(w, w_0)|)_+ \rightarrow \min_{w, w_0} .$$

**Достоинства и недостатки TSVM:**

- ⊕ как и в обычном SVM, можно использовать ядра;
- ⊕ имеются эффективные реализации для больших данных;
- ⊖ задача невыпуклая, методы оптимизации сложнее;
- ⊖ решение неустойчиво, если нет области разреженности;
- ⊖ требуется настройка двух параметров  $C, \gamma$ ;

---

*Sindhwani, Keerthi. Large scale semisupervised linear SVMs. SIGIR 2006.*

## Напоминание: многоклассовая логистическая регрессия

Линейный классификатор по конечному множеству классов  $|Y|$ :

$$a(x) = \arg \max_{y \in Y} \langle w_y, x \rangle, \quad x, w_y \in \mathbb{R}^n.$$

Вероятность того, что объект  $x_i$  относится к классу  $y$ :

$$P(y|x_i, w) = \frac{\exp\langle w_y, x_i \rangle}{\sum_{c \in Y} \exp\langle w_c, x_i \rangle}.$$

Задача максимизации регуляризованного правдоподобия:

$$Q(w) = \sum_{i=1}^k \log P(y_i|x_i, w) - \frac{1}{2C} \sum_{y \in Y} \|w_y\|^2 \rightarrow \max_w,$$

Оптимизация  $Q(w)$  — методом стохастического градиента, по  $n|Y|$ -мерному вектору параметров  $w = (w_y : y \in Y)$ .

## Согласование модели на размеченных и неразмеченных данных

Теперь учтём неразмеченные данные  $U = \{x_{k+1}, \dots, x_\ell\}$ .  
Пусть  $b_j(x)$  — бинарные признаки,  $j = 1, \dots, m$ .

Оценим вероятности  $P(y|b_j(x) = 1)$  двумя способами:

1) эмпирическая оценка по размеченным данным  $X^k$ :

$$\hat{p}_j(y) = \frac{\sum_{i=1}^k b_j(x_i) [y_i = y]}{\sum_{i=1}^k b_j(x_i)};$$

2) оценка по неразмеченным данным  $U$  и вероятностной модели:

$$p_j(y|w) = \frac{\sum_{i=k+1}^{\ell} b_j(x_i) P(y|x_i, w)}{\sum_{i=k+1}^{\ell} b_j(x_i)}.$$

Максимизируем правдоподобие вероятностной модели  $p_j(y|w)$ , приближающей эмпирическое распределение  $\hat{p}_j(y)$ .

## Построение регуляризатора (XR, eXpectation Regularization)

Логарифм правдоподобия модели классов по  $j$ -му признаку:

$$L_j(w) = \sum_{y \in Y} \hat{p}_j(y) \log p_j(y|w) \rightarrow \max_w.$$

Регуляризация критерия  $Q(w)$  суммой log-правдоподобий  $L_j(w)$  с коэффициентом регуляризации  $\gamma$ :

$$\begin{aligned} Q(w) + \gamma \sum_{j=1}^m L_j(w) &= \sum_{i=1}^k \log P(y_i|x_i, w) - \frac{1}{2C} \sum_{y \in Y} \|w_y\|^2 + \\ &+ \gamma \sum_{j=1}^m \sum_{y \in Y} \hat{p}_j(y) \log \left( \sum_{i=k+1}^{\ell} b_j(x_i) P(y|x_i, w) \right) \rightarrow \max_w. \end{aligned}$$

---

*Mann, McCallum*. Simple, robust, scalable semi-supervised learning via expectation regularization. ICML 2007.

## Особенности метода XR (eXpectation Regularization)

- 1 XR — это SSL, но это вообще не кластеризация!
- 2 Оптимизация методом стохастического градиента.
- 3 Возможные варианты задания переменных  $b_j$ :
  - $b_j(x) \equiv 1$ , тогда  $P(y|b_j(x) = 1)$  — априорная вероятность класса  $y$  (label regularization)  
— подходит для задач с несбалансированными классами;
  - $b_j(x) = [\text{термин } j \text{ содержится в тексте } x]$   
— подходит для задач классификации текстов.
- 4 метод слабо чувствителен к выбору  $C$  и  $\gamma$ ,
- 5 устойчив к погрешностям оценивания  $\hat{p}_j(y)$ ,
- 6 не требует большого числа размеченных объектов  $k$ ,
- 7 хорошо подходит для категоризации текстов.

---

*Mann, McCallum*. Simple, robust, scalable semi-supervised learning via expectation regularization. ICML 2007.

## Резюме в конце лекции

- Кластеризация — это обучение без учителя, некорректно поставленная задача, существует много оптимизационных и эвристических алгоритмов кластеризации
- DBSCAN — популярный быстрый алгоритм кластеризации
- Карты Кохонена — кластеризация + визуализация
- Задача SSL занимает промежуточное положение между классификацией и кластеризацией, но не сводится к ним.
- Методы кластеризации легко адаптируются к SSL путём введения ограничений (constrained clustering).
- Адаптация методов классификации реализуется сложнее, но приводит к более эффективным методам.
- Регуляризация позволяет учитывать дополнительные данные в оптимизационной задаче обучения.