СТАТИСТИКА БОЛЬШИХ ДАННЫХ

Лекция 25. Методы и алгоритмы кластеризации больших данных

Постановка задачи кластеризации

Рассматривается множество объектов (ситуаций) X. Задано **подмножество прецедентов** $X^l = \{x_1, \dots, x_l\} \subset X$, по каждому из которых собраны (измерены) некоторые данные. Задана функция расстояния между объектами $\rho(x, x')$, где $x, x' \in X$.

Задача. Разбить выборку X на непересекающиеся подмножества, называемые **кластерами**, так, чтобы каждый кластер состоял из объектов, близких по метрике $\rho(x, x')$, а объекты разных кластеров существенно отличались.

Этапы решения задачи кластеризации

Применение кластерного анализа сводится к следующим этапам:

- Отбор выборки объектов для кластеризации.
- Определение множества переменных, по которым будут оцениваться объекты в выборке. При необходимости – нормализация значений переменных.
- Задание меры сходства (расстояния) между объектами.
- Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
- Представление результатов анализа.

Меры расстояния

Евклидово расстояние

$$\rho_2(x, x') = \sqrt{\sum_i^n (x_i - x_i')^2}$$

Квадрат евклидова расстояния

$$\rho_2^2(x, x') = \sum_i^n (x_i - x_i')^2$$

Манхэттенское расстояние

$$\rho_1(x, x') = \sum_i^n |x_i - x_i'|$$

Расстояние Чебышева

$$\rho_\infty(x, x') = max_i(|x_i - x_i'|)$$

Частота несовпадений

$$\rho_I(x, x') = \frac{1}{n}\sum_i^n I(x_i \neq x_i')$$

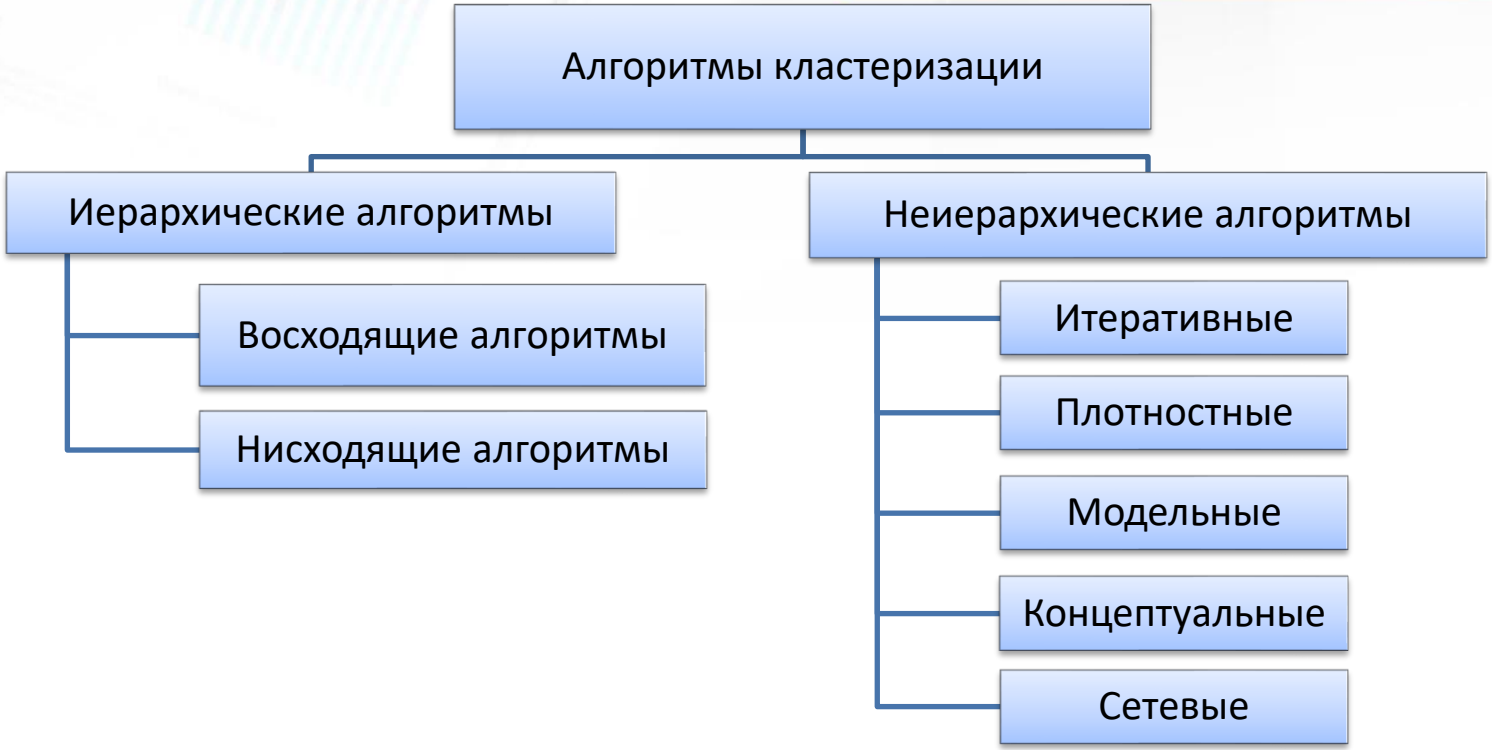
Оптимальное разбиение

Задачу кластеризации можно рассматривать как построение оптимального разбиения множества объектов *X* на непересекающиеся классы $X = \bigcup_{j=1}^{k} X_j$. При этом оптимальность может быть определена как требование минимизации среднеквадратической ошибки разбиения:

$$R(X_1, \dots, X_k) = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \rho_2^2 \left(x_i^{(j)}, m_j \right)$$

где k – число кластеров, n_j — число элементов в j-ом кластере, m_j — центр кластера X_j, $j = 1, \dots, k$

Классификация алгоритмов



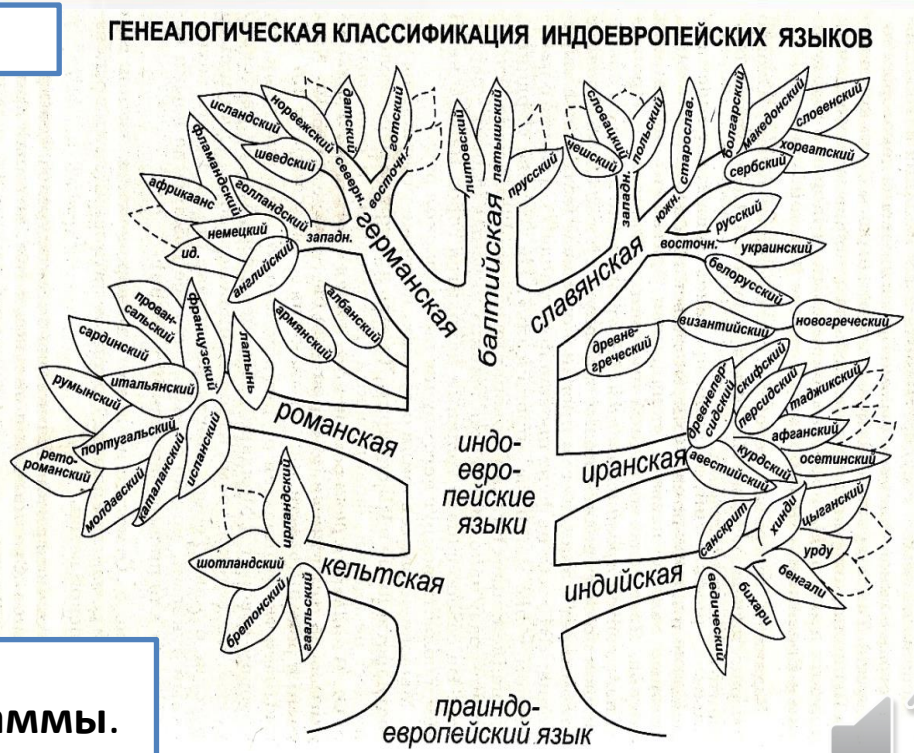
Иерархические алгоритмы

Восходящие и нисходящие алгоритмы

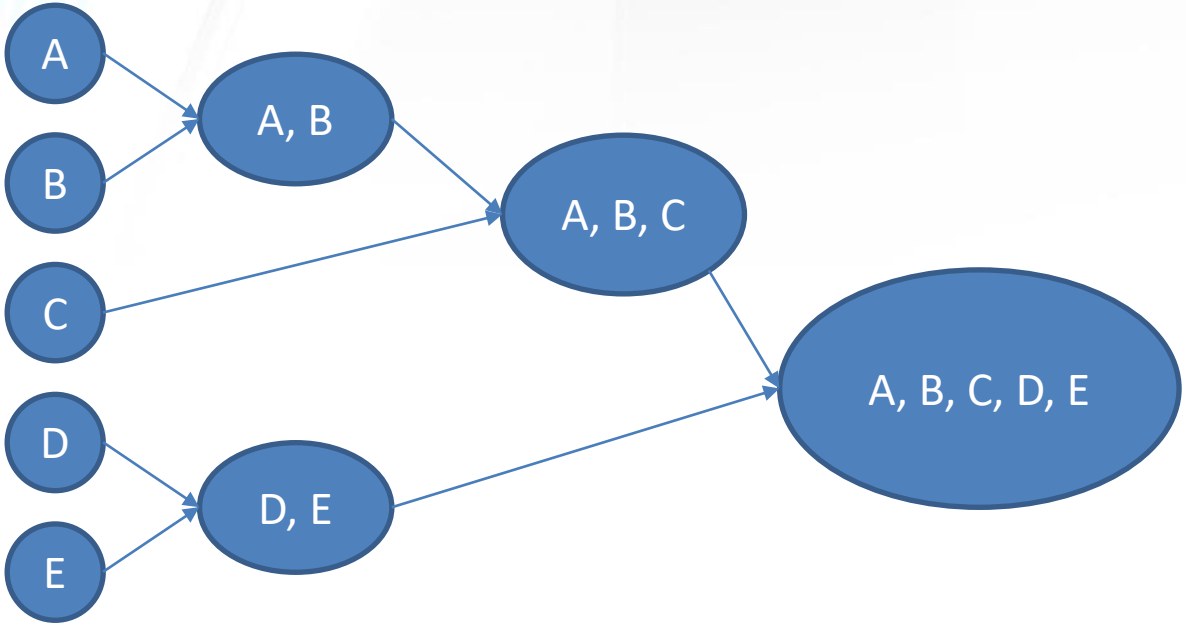
Нисходящие алгоритмы (сверху-вниз): в начале все объекты помещаются в один кластер, который затем разбивается на все более мелкие кластеры

Восходящие алгоритмы в начале работы помещают каждый объект в отдельный кластер, а затем объединяют кластеры во все более крупные.

Результаты таких алгоритмов обычно представляют в виде дерева – **дендрограммы**.



Пример иерархической кластеризации



Алгоритм ближайшего соседа

Составление матрицы попарных расстояний между объектами. Каждому объекту назначается свой кластер.

1) Нахождение в матрице наименьшего элемента (то есть наименьшего расстояния между соседями).

2) Объединение кластеров, в которые входят объекты, имеющие наименьшее расстояние.

3) Проверка: сколько осталось кластеров. Если один, то завершить алгоритм. Если два и более, то перейти к шагу 1.

Алгоритм k средних

1. Назначаем число *k* кластеров
2. Назначаем *k* начальных центров $m_j, j = 1, \dots, k$ этих кластеров.
3. Все объекты относим к ближайшим центрам, формируя тем самым начальные кластеры $C_1, C_2, \dots, C_k$.
4. Вычисляем центры m_j (например, средние значения) каждого кластера C_j.
5. Для каждого объекта x_i вычисляем расстояния $\rho(x_i, m_j)$. Ищем минимум этих расстояний по *j*. Если этот минимум достигается на «чужом» кластере, то объект x_i приписывается этому кластеру.
6. Если в результате работы п. 5 хотя бы один кластер изменился, то переходим на Шаг 4 иначе Завершение алгоритма.

Плотностные алгоритмы. DBSCAN

Алгоритм DBSCAN (Density-based spatial clustering of applications with noise)

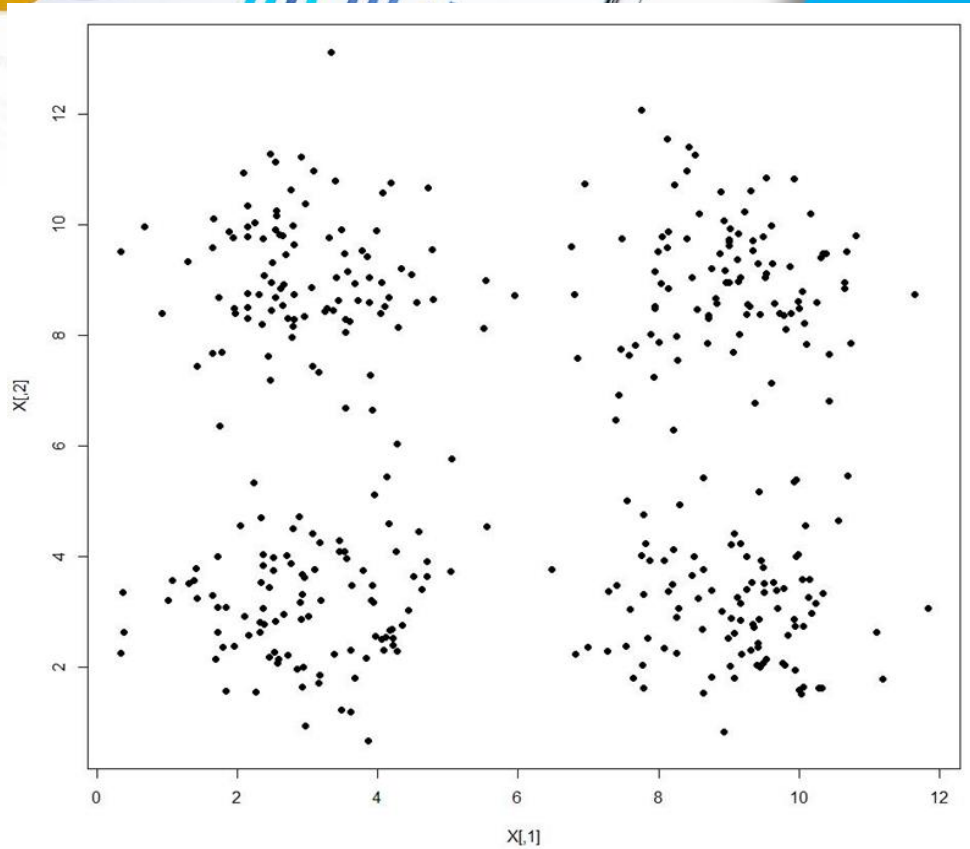
0. Выбираем число M и радиус окрестности ε.
1. Для точки x проверяем, что в ее ε-окрестности содержится не менее M других точек.
2 Если это не так, то эта точка - шум. Берем следующую точку.
3. Если это так, то помечаем эту точку как корневую точку кластера. Заносим окружающие ее точки в отдельное множество.
4. Рассматриваем каждую точку из этого множества и помечаем ее как принадлежащую кластеру, а затем проверяем, что в ее ε-окрестности есть как минимум M других точек. Если это так, то заносим точки из этой окрестности в то же множество. Если нет, то исключаем рассматриваемую точку.
5. Выбираем следующую точку x и переходим к пункту 1.

Модельные алгоритмы. EM-алгоритм

EM-алгоритм (Expectation-Maximization)

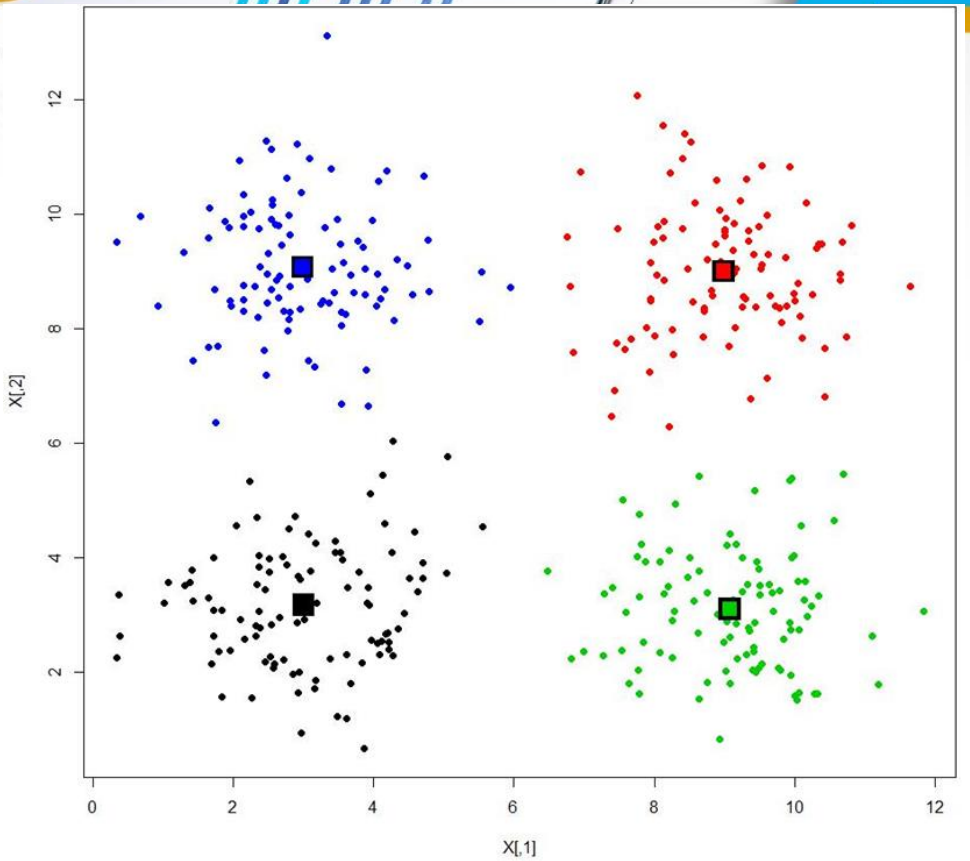
Алгоритм EM основан на предположении, что исследуемое множество данных может быть смоделировано с помощью смеси нормальных распределений.
Целью EM алгоритма является оценка параметров распределения, которые максимизируют функцию правдоподобия.
Определяются оптимальные параметры закона распределения – математическое ожидание и дисперсия, при которых функция правдоподобия максимальна.
Задача заключается в «подгонке» параметров смеси распределений к данным, а затем в определении вероятностей принадлежности объекта к каждому кластеру.
Любой объект принадлежит ко всем кластерам, но с разной вероятностью и должен быть отнесен к тому кластеру, для которого данная вероятность выше.

Пример. Метод k-средних в R




```
library(cluster)
library(MASS)
n<-100
k<-4
a1<-c(3,3)
a2<-c(9,3)
a3<-c(3,9)
a4<-c(9,9)
S<-diag(1,2,2)
X1<-mvrnorm(n,a1,S)
X2<-mvrnorm(n,a2,S)
X3<-mvrnorm(n,a3,S)
X4<-mvrnorm(n,a4,S)
X<-rbind(X1,X2,X3,X4)
plot(X, col = "black",type="p",pch=16)
cl<-kmeans(X,k)
cl
```

Пример. Метод k-средних в R



K-means clustering with 4 clusters of sizes 101, 99, 101, 99

Cluster means:

	[,1]	[,2]
1	3.006087	3.163084
2	8.989674	9.000543
3	9.078337	3.090313
4	2.990029	9.078335

Within cluster sum of squares by cluster:
230.1633 225.8615 206.3521 227.3265