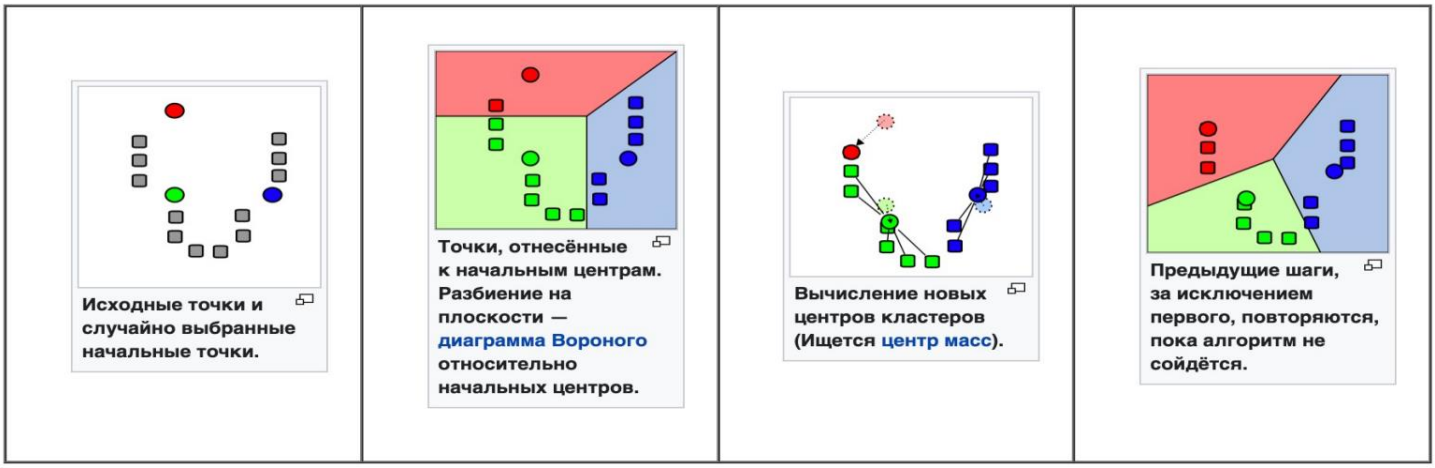
Агломеративные методы (англ. agglomerative): новые кластеры создаются путем объединения более мелких кластеров и, таким образом, дерево создается от листьев к стволу

Дивизивные или дивизионные методы (англ. divisive): новые кластеры создаются путем деления более крупных кластеров на более мелкие и, таким образом, дерево создается от ствола к листьям

Итеративные методы (англ. Iterative): дробления исходной совокупности. В процессе деления новые кластеры формируются до тех пор, пока не будет выполнено правило остановки

Метод k-средних (англ. k-means) — наиболее популярный метод кластеризации. Был изобретён в 1950-х годах математиком Гуго Штейнгаузом и почти одновременно Стюартом Ллойдом. Особую популярность приобрёл после работы Маккуина.



Алгоритм кластеризации K-средних вычисляет центроиды и выполняет итерации, пока мы не найдем оптимальный центроид. Предполагается, что количество кластеров уже известно. Это также называется алгоритм плоской кластеризации . Количество кластеров, идентифицированных по данным алгоритмом, обозначается буквой «K» в K-средних.

В этом алгоритме точки данных назначаются кластеру таким образом, чтобы сумма квадратов расстояния между точками данных и центроидом была бы минимальной. Следует понимать, что меньшее отклонение в кластерах приведет к большему количеству сходных точек данных в одном кластере.

Проблемы K-средних:

- Не гарантируется достижение глобального минимума суммарного квадратичного отклонения V, а только одного из локальных минимумов
- Результат зависит от выбора исходных центров кластеров, их оптимальный выбор неизвестен
- Число кластеров надо знать заранее.

Алгоритм k-средних принимает в качестве входных данных набор данных X, содержащий N точек, а также параметр K, задающий требуемое количество кластеров. На выходе получаем набор из K центроидов кластеров, кроме того, всем точкам множества X присваиваются метки, относящие их к определенному кластеру. Все точки в пределах данного кластера расположены ближе к своему центроиду, чем к любому другому центроиду

Математическое выражение для K кластеров C_k и K центроидов μ_k имеет вид:

$$\text{Минимизировать} \sum_{k=1}^{K} \sum_{x_n \in C_k} ||x_n - \mu_k||^2 \text{ относительно Ck, μk.}$$

Существует итерационный метод, известный как алгоритм Ллойда, который сходится (хотя и к локальному минимуму) в пределах небольшого количества итераций. В рамках данного алгоритма поочередно выполняются две операции.

(1) Как только набор центроидов μk становится доступен, каждый кластер обновляется таким образом, чтобы содержать точки ближайшие к данному центроиду.
(2) Как только набор кластеров становится доступен, каждый центроид пересчитывается, как среднее значение всех точек, принадлежащих данному кластеру.

Двухэтапная процедура повторяется, пока кластеры и центроиды не перестанут изменяться. Как уже было сказано, сходимость гарантируется, но решение может представлять собой локальный минимум. На практике, алгоритм выполняется несколько раз, и результаты усредняются. Для получения набора начальных центроидов можно использовать несколько методов, например центроиды можно задать случайным образом.

Минимизация суммы квадратов внутрикластерных расстояний:

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

Алгоритм Ллойда

вход: X^ℓ, $K = |Y|$; **выход:** центры кластеров μ_a, $a \in Y$;
$\mu_a :=$ начальное приближение центров, для всех $a \in Y$;
повторять
  отнести каждый x_i к ближайшему центру:
  $a_i := \arg\min_{a \in Y} \|x_i - \mu_a\|, \quad i = 1, \ldots, \ell$;
  вычислить новые положения центров:
  $\mu_a := \dfrac{\sum_{i=1}^{\ell} [a_i = a] x_i}{\sum_{i=1}^{\ell} [a_i = a]}, \quad a \in Y$;
пока a_i не перестанут изменяться;

Модификация алгоритма Ллойда

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

вход: X^ℓ, $K = |Y|$;
выход: центры кластеров μ_a, $a \in Y$;

$\mu_a :=$ начальное приближение центров, для всех $a \in Y$;
повторять
  отнести каждый $x_i \in U$ к ближайшему центру:
  $a_i := \arg\min_{a \in Y} \|x_i - \mu_a\|, \quad i = k+1, \ldots, \ell$;
  вычислить новые положения центров:
  $\mu_a := \dfrac{\sum_{i=1}^{\ell} [a_i = a] x_i}{\sum_{i=1}^{\ell} [a_i = a]}, \quad a \in Y$;
пока a_i не перестанут изменяться;

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

начальное приближение w_a, μ_a, Σ_a для всех $a \in Y$;
повторять
  E-шаг: отнести каждый x_i к ближайшим центрам:
    $g_{ia} := P(a|x_i) \equiv \dfrac{w_a p_a(x_i)}{\sum_y w_y p_y(x_i)}, \quad a \in Y, \quad i = 1, \ldots, \ell;$
    $a_i := \arg\max_{a \in Y} g_{ia}, \quad i = 1, \ldots, \ell;$
  M-шаг: вычислить новые положения центров:
    $\mu_{ad} := \frac{1}{\ell w_a} \sum_{i=1}^{\ell} g_{ia} f_d(x_i), \quad a \in Y, \quad d = 1, \ldots, n;$
    $\sigma^2_{ad} := \frac{1}{\ell w_a} \sum_{i=1}^{\ell} g_{ia} (f_d(x_i) - \mu_{ad})^2, \quad a \in Y, \; d = 1, \ldots, n;$
    $w_a := \frac{1}{\ell} \sum_{i=1}^{\ell} g_{ia}, \quad a \in Y;$
пока a_i не перестанут изменяться;

Основные отличия GMM-EM и *k*-means:

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

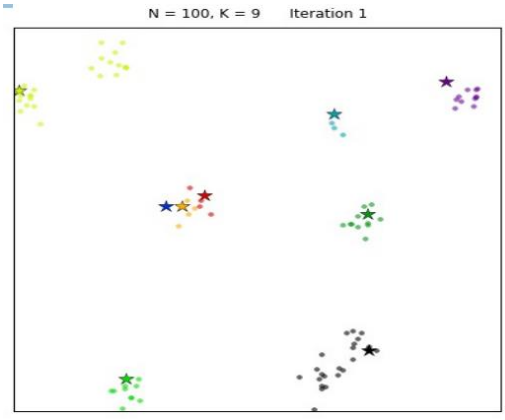
Гибриды (упрощение GMM-EM — усложнение *k*-means):

- 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++)

Если целевое распределение имеет разрозненную структуру, и используется только один экземпляр алгоритма Ллойда, возникает опасность того, что полученный локальный минимум не является оптимальным решением. Это показано в примере ниже, где начальные данные имеют островершинное гауссовское распределение:



Желтый и черный центроиды представляют по два различных кластера каждый, в то время как оранжевый, красный и синий центроиды теснятся в пределах одного кластера вследствие неудачной случайной инициализации. В подобных случаях помогает более рациональный выбор начальных кластеров.

Работа алгоритма k-средних:

Шаг 1 — Указать количество кластеров, K, которые должны быть сгенерированы этим алгоритмом.

Шаг 2 — Случайным образом выбрать K точек данных и назначить каждую точку данных кластеру. Т.е., классифицировать данные на основе количества точек данных.

Шаг 3 — Алгоритм готов вычислять кластерные центроиды.

Шаг 4 — Далее, выполняется следующее до тех пор, пока мы не найдем оптимальный центроид, который является назначением точек данных кластерам, которые больше не меняются

- 4.1 — Сначала будет вычислена сумма квадратов расстояния между точками данных и центроидами.
- 4.2 — Назначение каждую точку данных кластеру, который находится ближе, чем другой кластер (центроид).
- 4.3 — Вычисление центроиды для кластеров, взяв среднее значение всех точек данных этого кластера.

4.1 — Сначала будет вычислена сумма квадратов расстояния между точками данных и центроидами.

4.2 — Назначение каждую точку данных кластеру, который находится ближе, чем другой кластер (центроид).

4.3 — Вычисление центроиды для кластеров, взяв среднее значение всех точек данных этого кластера.

При работе с алгоритмом K-means необходимо:

- При работе с алгоритмами кластеризации, включая K-Means, рекомендуется стандартизировать данные, поскольку такие алгоритмы используют измерения на основе расстояний для определения сходства между точками данных.
- Из-за итеративной природы K-средних и случайной инициализации центроидов K-средние могут придерживаться локального оптимума и могут не сходиться к глобальному оптимуму. Вот почему рекомендуется использовать разные инициализации центроидов.

Преимущества:

- При большом количество переменных, K-means быстрее, чем иерархическая кластеризация и метод главных компонент.
- При повторном вычислении центроидов экземпляр может изменить кластер.
- Более плотные кластеры формируются с помощью K-средних по сравнению с иерархической кластеризацией

Недостатки:

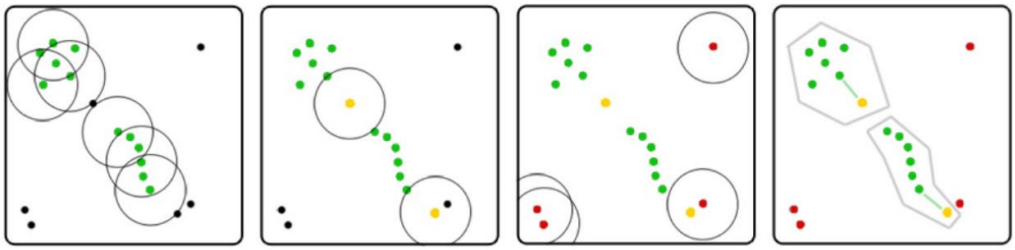
- Немного сложно предсказать количество кластеров, то есть значение k.
- На выход сильно влияют исходные данные, такие как количество кластеров (значение k)
- Порядок данных будет иметь сильное влияние на конечный результат.
- Это очень чувствительно к масштабированию. Если мы будем масштабировать наши данные с помощью нормализации или стандартизации, то вывод полностью изменится.
- В кластеризации плохо работать, если кластеры имеют сложную геометрическую форму.

Алгоритм DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

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

Каждый объект может быть одного из трёх типов:

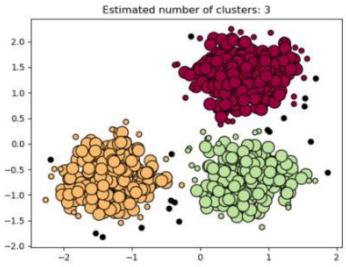
- корневой: имеющий плотную окрестность, $|U_\varepsilon(x)| \geqslant m$
- граничный: не корневой, но в окрестности корневого
- шумовой (выброс): не корневой и не граничный



вход: выборка $X^\ell = \{x_1, \ldots, x_\ell\}$; параметры ε и m;
выход: разбиение выборки на кластеры и шумовые выбросы;
$U := X^\ell$ — непомеченные; $a := 0$;
пока в выборке есть непомеченные точки, $U \neq \varnothing$:
  взять случайную точку $x \in U$;
  если $|U_\varepsilon(x)| < m$ **то**
    пометить x как, возможно, шумовой;
  иначе
    создать новый кластер: $K := U_\varepsilon(x)$; $a := a + 1$;
    для всех $x' \in K$, не помеченных или шумовых
      если $|U_\varepsilon(x')| \geqslant m$ **то** $K := K \cup U_\varepsilon(x')$;
      иначе пометить x' как граничный кластера K;
    $a_i := a$ для всех $x_i \in K$;
    $U := U \setminus K$;

Преимущества алгоритма:

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.

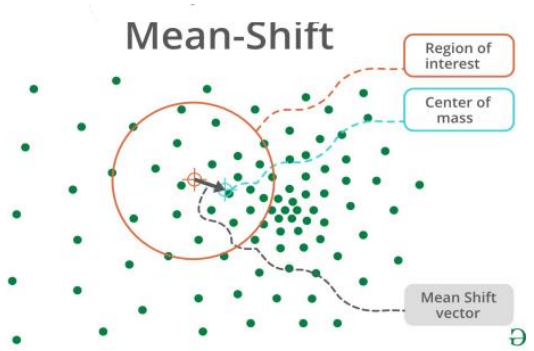


В алгоритма два входных параметра min_samples и eps. Если в заданной окрестности eps есть минимальное необходимое количество объектов min_samples, то данная окрестность будет считаться кластером.

Если в заданной области нет необходимого количества объектов, то инициирующая эту область точка считается выбросом

В отличие от k-средних, DBSCAN определит количество кластеров. DBSCAN работает, определяя, достаточно ли минимальное количество точек достаточно близко друг к другу, чтобы считаться частью одного кластера. DBSCAN очень чувствителен к масштабу, поскольку epsilon - это фиксированное значение для максимального расстояния между двумя точками.

Mean shift – алгоритм сдвига среднего значения



Сдвиг среднего значения — это непараметрическая техника анализа пространства признаков для определения местоположения максимума плотности вероятности, так называемый алгоритм поиска моды. Область применения техники — кластерный анализ в компьютерном зрении и обработке изображений.

Преимущества:

- Сдвиг среднего значения является независимым от приложения средством, пригодным для анализа реальных данных.
- Метод не предполагает предварительного задания формы кластеров.
- Алгоритм способен обрабатывать произвольные пространства признаков.
- Процедура опирается на выбор единственного параметра — ширина полосы.
- Размер полосы пропускания/окна h имеет физический смысл, не совпадающий с k-средним.

Недостатки:

- Выбор размера окна нетривиален
- Неподходящий размер окна может привести к слиянию мод или образованию дополнительных «теневых» мод.
- Часто требуется использования самонастраиваемого размера окна.

Шаги работы алгоритма среднего сдвига:

- Шаг 1. Начинает с точек данных, назначенных собственному кластеру.
- Шаг 2 - Затем вычисляет центроиды.
- Шаг 3 - Обновление расположение новых центроидов.
- Шаг 4 - Процесс будет повторятся с перемещением в область с более высокой плотностью.
- Шаг 5 - Процесс будет остановлен, как только центроиды достигнут с наивысше й плотностью.

Агломеративная иерархическая кластеризация

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

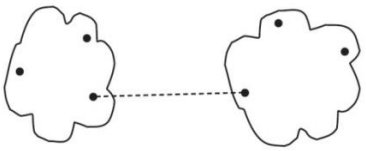
$C_1 := \{\{x_1\}, \ldots, \{x_\ell\}\}$ — все кластеры 1-элементные;
$R_{\{x_i\}\{x_j\}} := \rho(x_i, x_j)$ — расстояния между ними;
для всех $t = 2, \ldots, \ell$ (t — номер итерации):
  найти в C_{t-1} пару кластеров (U, V) с минимальным R_{UV};
  слить их в один кластер:
  $W := U \cup V$;
  $C_t := C_{t-1} \cup \{W\} \setminus \{U, V\}$;
  для всех $S \in C_t$
    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:

$R^{\text{б}}_{WS} = \min\limits_{w \in W, s \in S} \rho(w, s)$;

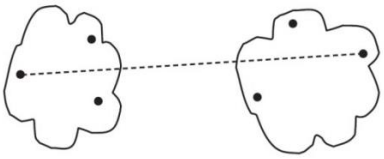
$\alpha_U = \alpha_V = \frac{1}{2}, \ \beta = 0, \ \gamma = -\frac{1}{2}$.



2. Расстояние дальнего соседа:

$R^{\text{д}}_{WS} = \max\limits_{w \in W, s \in S} \rho(w, s)$;

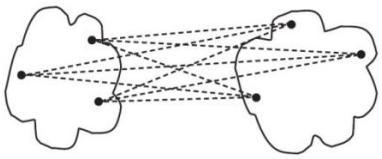
$\alpha_U = \alpha_V = \frac{1}{2}, \ \beta = 0, \ \gamma = \frac{1}{2}$.



3. Групповое среднее расстояние:

$R^{\text{г}}_{WS} = \frac{1}{|W||S|} \sum\limits_{w \in W} \sum\limits_{s \in S} \rho(w, s)$;

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

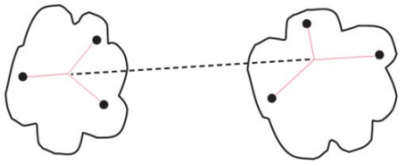


4. Расстояние между центрами:

$R^{\text{ц}}_{WS} = \rho^2\Big(\sum\limits_{w \in W} \frac{w}{|W|}, \sum\limits_{s \in S} \frac{s}{|S|}\Big)$;

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

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



5. Расстояние Уорда:

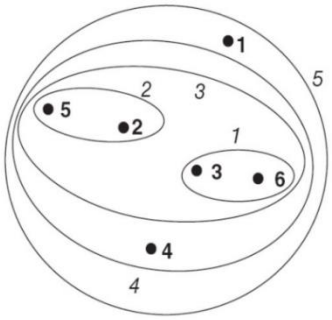
$R^{\text{у}}_{WS} = \frac{|S||W|}{|S|+|W|} \rho^2\Big(\sum\limits_{w \in W} \frac{w}{|W|}, \sum\limits_{s \in S} \frac{s}{|S|}\Big)$;

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

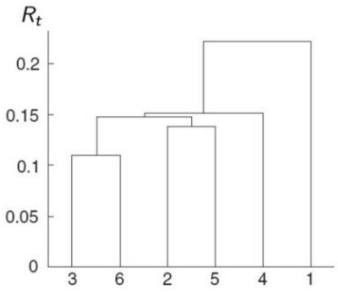
Визуализация кластерной структуры

1. Расстояние ближнего соседа:

Диаграмма вложения



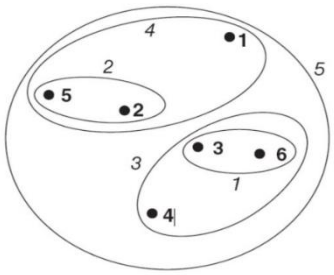
Дендрограмма



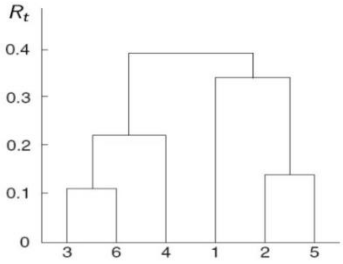
метод одиночной связи (метод «ближайшего соседа»). Алгоритм начинается с поиска двух наиболее близких объектов, пара которых образует первичный кластер. Каждый последующий объект присоединяется к тому кластеру, к одному из объектов которого он ближе

2. Расстояние дальнего соседа:

Диаграмма вложения



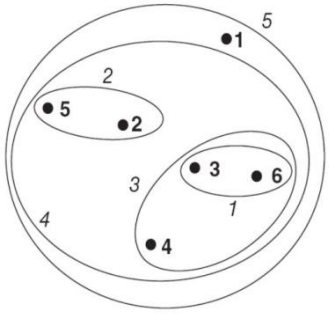
Дендрограмма



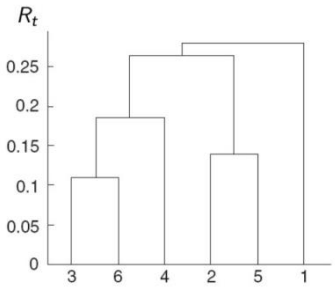
метод полной связи (метод «дальнего соседа»). Правило объединения этого метода подразумевает, что новый объект присоединяется к тому кластеру, самый далекий элемент которого находится ближе к новому объекту, чем самые далекие элементы других кластеров

3. Групповое среднее расстояние:

Диаграмма вложения



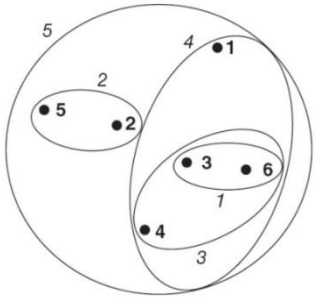
Дендрограмма



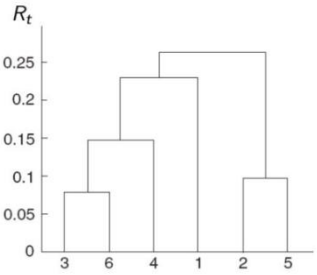
метод средней (межгрупповой) связи. На каждом шаге вычисляется среднее арифметическое расстояние между каждым объектом из одного кластера и каждым объектом другого кластера либо вычисляется расстояние между центрами тяжести кластеров. Объединяются те кластеры, расстояние между которыми является наименьшим

5. Расстояние Уорда:

Диаграмма вложения



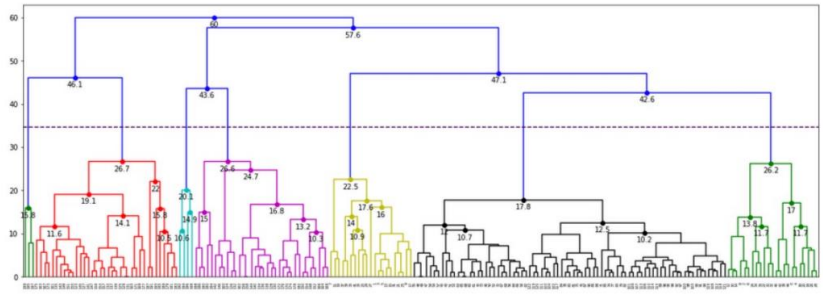
Дендрограмма



метод Уорда. На первом шаге каждый кластер состоит из одного объекта, в силу чего внутрикластерная дисперсия расстояний равна нулю. Объединяются те объекты, которые дают минимальное приращение дисперсии.

Дендрограмма – визуализация иерархической системы

- Кластеры группируются вдоль горизонтальной оси
- По вертикальной оси откладываются расстояния R_t
- Расстояния возрастают, линии нигде не пересекаются
- Верхние уровни различимы лучше, чем нижние
- Уровень отсечения определяет число кластеров



Основные свойства иерархической системы

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

Теорема (Миллиган, 1979)

Кластеризация монотонна, если выполняются условия

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

$R^{ч}$ не монотонно; $R^{б}$, $R^{д}$, $R^{г}$, $R^{у}$ — монотонны.

$R^{б}$ — сжимающее; $R^{д}$, $R^{у}$ — растягивающие;

Выводы

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