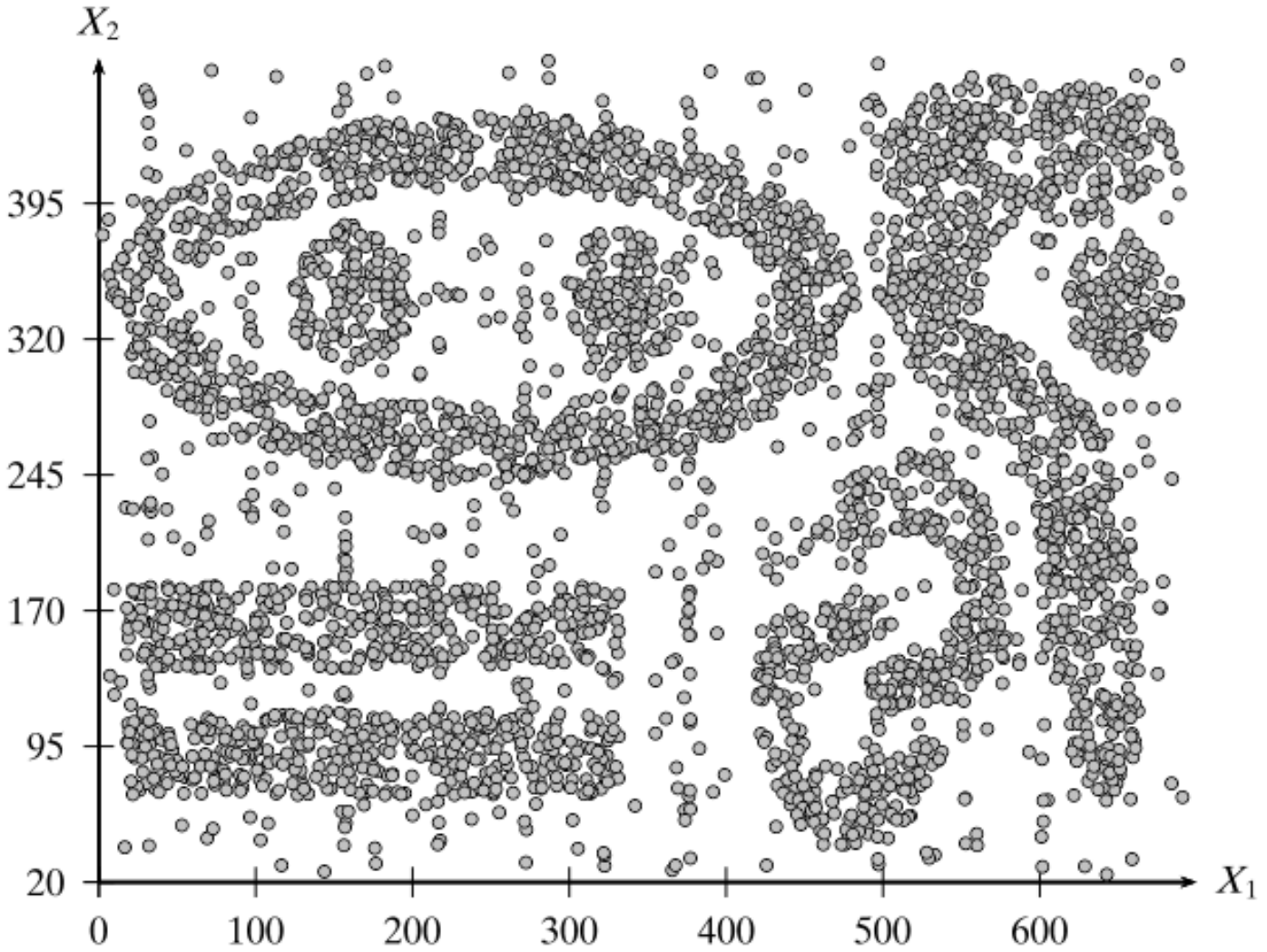
Лекция 7
Глава 15

Основанная на плотности кластеризация

Пример датасета для кластеризации, основанной на плотности



Алгоритм DBSCAN

Алгоритм DBSCAN

Кластеризация, основанная на плотности, использует не расстояние между точками, а локальную плотность точек для определения кластеров. Будем называть *ϵ-соседями* точки $\mathbf{x} \in \mathbb{R}^d$ сферу радиуса ϵ вокруг этой точки, заданную следующим образом:

$$N_\epsilon(\mathbf{x}) = B_d(\mathbf{x}, \epsilon) = \{\mathbf{y} \mid \delta(\mathbf{x}, \mathbf{y}) \leq \epsilon\},$$

где $\delta(\mathbf{x}, \mathbf{y})$ представляет собой расстояние между точками $\mathbf{x}$ и $\mathbf{y}$. Как правило, речь идёт о евклидовом расстоянии, то есть, $\delta(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$, но могут применяться и другие метрики.

Алгоритм DBSCAN

Мы будем называть точку $\mathbf{x} \in \mathbf{D}$ *базовой (ядерной)*, если хотя бы $minpts$ точек являются её ϵ-соседями. Другими словами, точка $\mathbf{x}$ является базовой, если $|N_\epsilon(\mathbf{x})| \geq minpts$, где $minpts$ – задаваемая пользователем локальная плотность или порог частоты.

Граничной точкой называется точка, которая не удовлетворяет порогу $minpts$, то есть, для неё $|N_\epsilon(\mathbf{x})| \leq minpts$, но при этом она является ϵ-соседней для некоторой базовой точки $\mathbf{z}$, то есть, $\mathbf{x} \in N_\epsilon(\mathbf{z})$. Наконец, если точка не является ни базовой, ни граничной, она считается *точкой шума* или выбросом.

Будем считать, что точка $\mathbf{x}$ *напрямую достижима по плотности* из точки $\mathbf{y}$, если $\mathbf{x} \in N_\epsilon(\mathbf{y})$ и $\mathbf{y}$ является базовой точкой. Точка $\mathbf{x}$ *достижима по плотности* из точки $\mathbf{y}$, если существует последовательность точек $\mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_l$ такая, что $\mathbf{x} = \mathbf{x}_0$, $\mathbf{y} = \mathbf{x}_l$ и $\mathbf{x}_i$ напрямую достижима из $\mathbf{x}_{i-1}$ для всех $i = 1, \ldots, l$. Другими словами, существует множество базовых точек, ведущих от $\mathbf{y}$ к $\mathbf{x}$. Заметим, что отношение достижимости по плотности асимметричное или направленное. Определим любые две точки $\mathbf{x}$ и $\mathbf{y}$ как *связанные по плотности*, если существует базовая точка $\mathbf{z}$ такая, что и $\mathbf{x}$, и $\mathbf{y}$ достижимы из $\mathbf{z}$. Кластером, основанным на плотности, называется максимальное множество точек, связанных по плотности.

Сначала DBSCAN вычисляет ϵ-соседей $N_\epsilon(\mathbf{x}_i)$ для всех точек $\mathbf{x}_i$ датасета $\mathbf{D}$, а затем проверяет, являются ли они базовыми (строки 2-5). Кроме того, алгоритм присваивает всем точкам значение идентификатора кластера $id(\mathbf{x}_i) = \emptyset$, отмечая, что они не принадлежат ни одному кластеру. Далее, начиная с каждой базовой точки, не присвоенной ни одному кластеру, метод рекурсивно ищет все точки, связанные по плотности с исходной, и присваивает их одному кластеру (строка 10). Некоторые граничные точки могут быть достижимы из базовых из более чем одного кластера, они могут быть присвоены любому кластеру или всем (если допускается пересечение кластеров). Те точки, которые не принадлежат ни одному кластеру, помечаются как выбросы или шум.

DBSCAN $(\mathbf{D}, \epsilon, minpts)$:

1 $Core \leftarrow \emptyset$
2 **foreach** $\mathbf{x}_i \in \mathbf{D}$ **do** // Find the core points
3 Compute $N_\epsilon(\mathbf{x}_i)$
4 $id(\mathbf{x}_i) \leftarrow \emptyset$ // cluster id for $\mathbf{x}_i$
5 **if** $N_\epsilon(\mathbf{x}_i) \geq minpts$ **then** $Core \leftarrow Core \cup \{\mathbf{x}_i\}$
6 $k \leftarrow 0$ // cluster id
7 **foreach** $\mathbf{x}_i \in Core$, *such that* $id(\mathbf{x}_i) = \emptyset$ **do**
8 $k \leftarrow k + 1$
9 $id(\mathbf{x}_i) \leftarrow k$ // assign $\mathbf{x}_i$ to cluster id k
10 DENSITYCONNECTED $(\mathbf{x}_i, k)$
11 $\mathcal{C} \leftarrow \{C_i\}_{i=1}^k$, where $C_i \leftarrow \{\mathbf{x} \in \mathbf{D} \mid id(\mathbf{x}) = i\}$
12 $Noise \leftarrow \{\mathbf{x} \in \mathbf{D} \mid id(\mathbf{x}) = \emptyset\}$
13 $Border \leftarrow \mathbf{D} \setminus \{Core \cup Noise\}$
14 **return** $\mathcal{C}, Core, Border, Noise$

DENSITYCONNECTED $(\mathbf{x}, k)$:

15 **foreach** $\mathbf{y} \in N_\epsilon(\mathbf{x})$ **do**
16 $id(\mathbf{y}) \leftarrow k$ // assign $\mathbf{y}$ to cluster id k
17 **if** $\mathbf{y} \in Core$ **then** DENSITYCONNECTED $(\mathbf{y}, k)$

DBSCAN можно рассматривать как поиск связных компонент в графе, где вершины соответствуют базовым точкам в датасете, и существует (ненаправленное) ребро между двумя вершинами (базовыми точками), если расстояние между ними меньше ϵ, то есть, каждая из них является ϵ-соседом для другой.

Ограничение DBSCAN – чувствительность к выбору ϵ, в частности, в тех случаях, когда кластеры имеют разную плотность. Если выбрать ϵ слишком маленьким, более разреженные кластеры будут классифицированы как шум. Если значение ϵ слишком велико, более плотные кластеры могут быть слиты в один.

Вычислительная сложность

Основные вычислительные затраты DBSCAN – это вычисление всех ϵ-соседей каждой точки. Если размерность датасета невелика, это можно эффективно сделать, используя структуру пространственного индекса за время $O(n \log(n))$. Если размерность велика, вычисление соседей каждой точки требует времени $O(n^2)$. После вычисления $N_\epsilon(\mathbf{x})$ алгоритму нужен только один проход по всем точкам для поиска связанных по плотности кластеров. Таким образом, вычислительная сложность алгоритма DBSCAN в худшем случае – $O(n^2)$.

ОЦЕНКА ПЛОТНОСТИ ЯДРА

Целью оценки плотности является определение неизвестной функции плотности вероятности путем нахождения плотных областей точек, которые, в свою очередь, могут использоваться для кластеризации.

Оценка плотности ядра — это непараметрический метод, который пытается напрямую вывести базовую плотность вероятности в каждой точке набора данных.

Одномерная оценка плотности

Предположим, что X является непрерывной случайной величиной, и пусть x_1, $x_2, \ldots, x_n$ будет случайной выборкой, взятой из лежащей в основе функции плотности вероятности $f(x)$, которая предполагается неизвестной. Мы можем напрямую оценить кумулятивную функцию распределения по данным, посчитав, сколько точек меньше или равно x:

$$\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} I(x_i \leq x)$$

где I – индикаторная функция, которая имеет значение 1 только тогда, когда ее аргумент истинен, и 0 в противном случае.

Одномерная оценка плотности

Мы можем оценить функцию плотности, взяв производную от $\hat{F}(x)$, рассматривая окно небольшой ширины h с центром в точке x, то есть:

$$\hat{f}(x) = \frac{\hat{F}\left(x + \frac{h}{2}\right) - \hat{F}\left(x - \frac{h}{2}\right)}{h} = \frac{\frac{k}{n}}{h} = \frac{k}{nh} \quad (15.1)$$

где k – количество точек, которые лежат в окне шириной h с центром в x, то есть в закрытом интервале $[x - \frac{h}{2},\ x + \frac{h}{2}]$. Таким образом, оценка плотности – это отношение доли точек в окне (k/n) к объему окна (h). Здесь h играет роль «влияния».

Оценщик ядра

Оценка плотности ядра основывается на функции ядра K, которая является неотрицательной, симметричной и интегрируется с 1, то есть $K(x) \geq 0, K(-x) = K$для всех значений x и $\int K(x)dx = 1$. Таким образом, K – это, по сути, функция плотности вероятности.

Дискретное ядро оценку плотности $\hat{f}(x)$ из уравнения (15.1) также можно переписать в терминах функции ядра следующим образом:

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

Оценщик ядра

дискретная функция ядра K вычисляет количество точек в окне шириной h и определяется как:

$$K(z) = \begin{cases} 1 \text{ If } |z| \leq \frac{1}{2} \\ 0 \text{ В противном случае} \end{cases} \qquad (15.2)$$

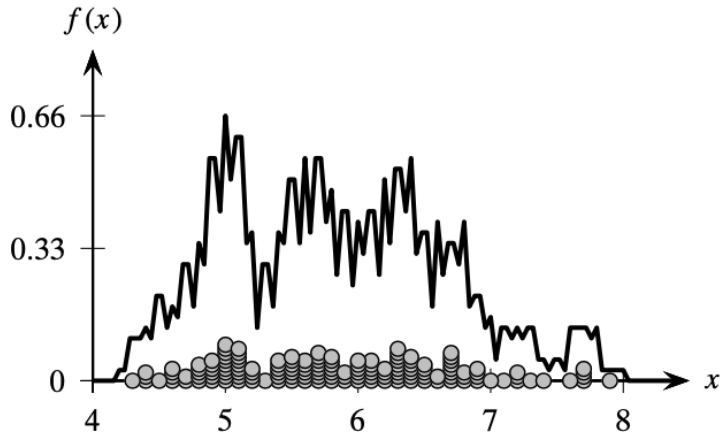
Мы видим, что если $|z| = \left|\frac{x-x_i}{h}\right| \leq \frac{1}{2}$, то точка x_i находится в пределах окна шириной h с центром в x, так как

$$\left|\frac{x-x_i}{h}\right| \leq \frac{1}{2} \text{ подразумевает что } -\frac{1}{2} \leq \frac{x-x_i}{h} \leq \frac{1}{2}, \text{ или}$$

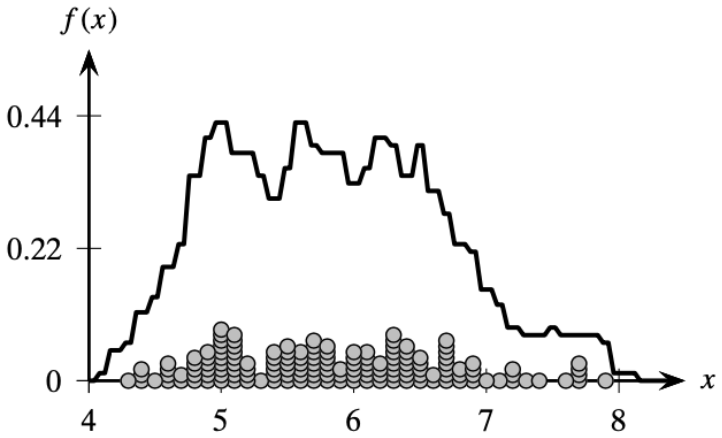
$$-\frac{h}{2} \leq x - x_i \leq \frac{h}{2}, \text{ и наконец}$$

$$x - \frac{h}{2} \leq x_i \leq x + \frac{h}{2}$$

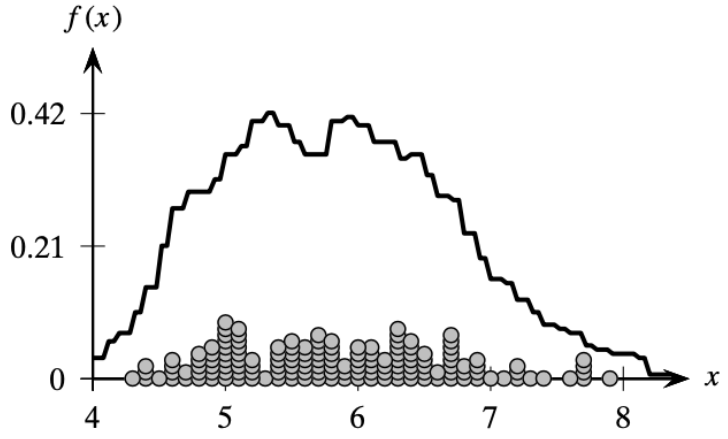
Пример 15.4.



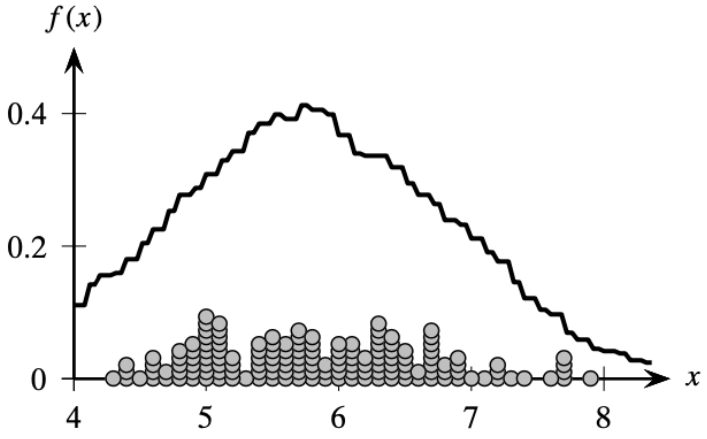
(a) $h = 0.25$



(b) $h = 0.5$



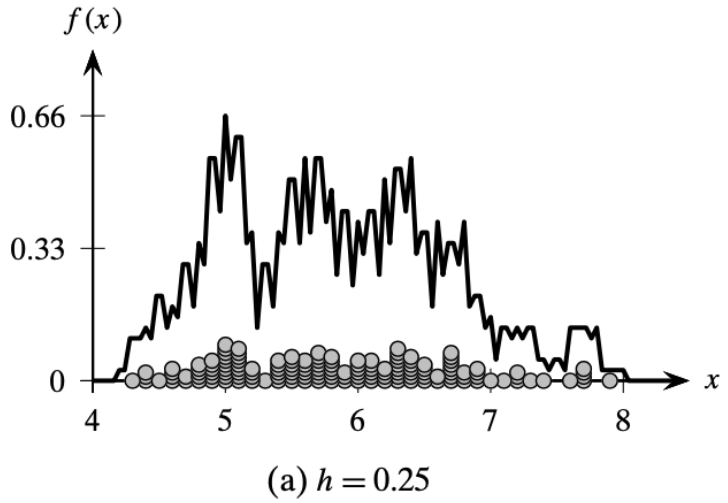
(c) $h = 1.0$



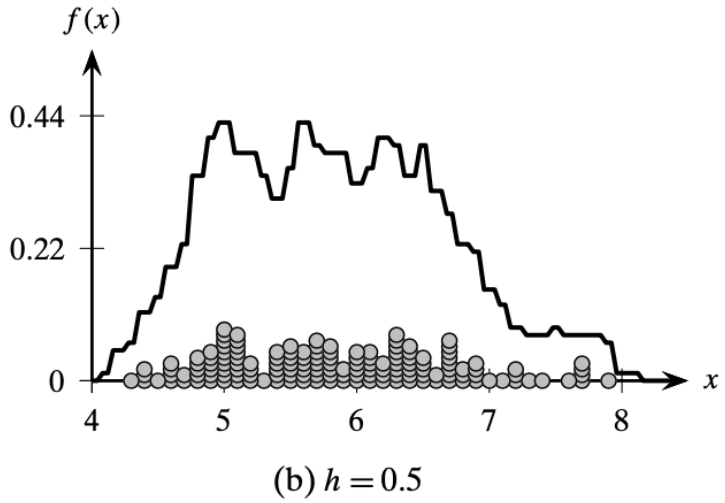
(d) $h = 2.0$

На рисунке 15.5 показаны оценки плотности ядра с использованием дискретного ядра для различных значений параметра влияния h для одномерного набора данных Iris, содержащего атрибут длины чашелистника. По оси абсцисс отложены n = 150 точек данных. Поскольку несколько точек имеют одно и то же значение, они отображаются в стопке, где высота стопки соответствует частоте этого значения.

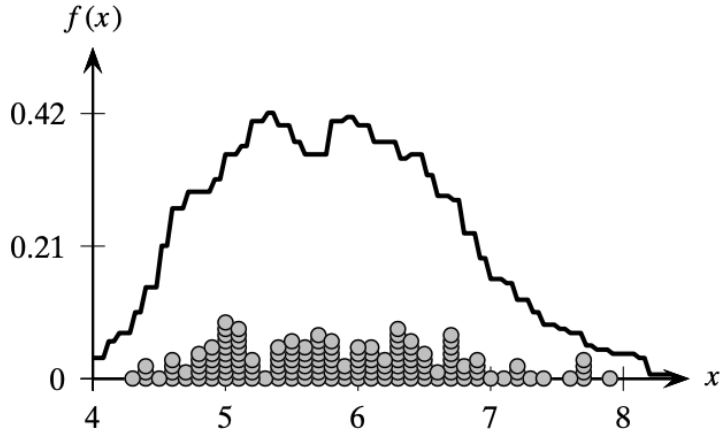
Пример 15.4.



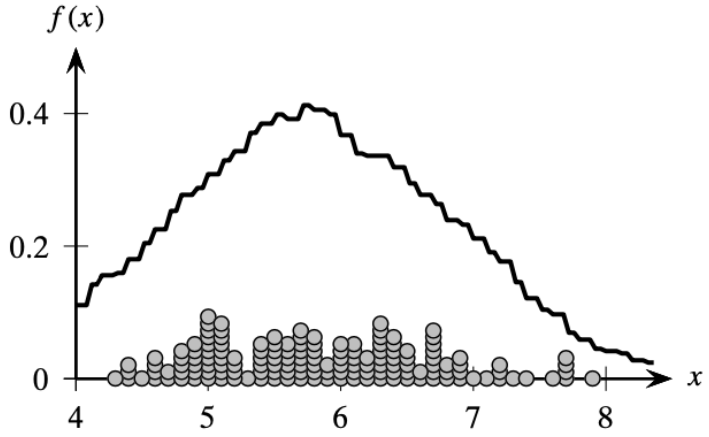
(a) $h = 0.25$



(b) $h = 0.5$



(c) $h = 1.0$



(d) $h = 2.0$

Когда h мало, как показано на рисунке 15.5a, функция плотности имеет много локальных максимумов или мод. Однако, когда мы увеличиваем h с 0,25 до 2, количество мод уменьшается, пока h не станет достаточно большим, чтобы получить унимодальное распределение, как показано на рисунке 15.5d. Мы можем заметить, что дискретное ядро дает негладкую (или зубчатую) функцию плотности.

Гауссово ядро

Гауссово ядро ширина h – параметр, обозначающий разброс или гладкость оценки плотности. Если разброс слишком большой, мы получаем более усредненное значение. Если он слишком маленький, нам не хватает точек в окне. Кроме того, функция ядра в уравнении (15.2) оказывает большое влияние. Для точек внутри окна ($|z| \leq \frac{1}{2}$) существует чистый вклад в оценку вероятности $\hat{f}(x)$ составляет $\frac{1}{nh}$. С другой стороны, точки вне окна ($|z| > \frac{1}{2}$) дают 0.

Гауссово ядро

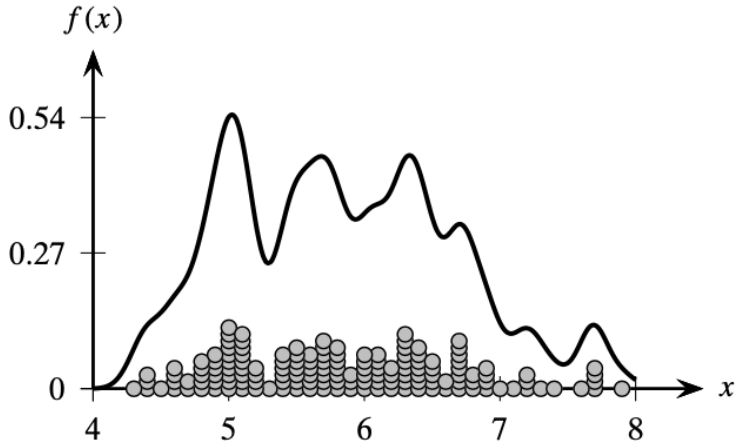
Вместо дискретного ядра мы можем определить более плавный переход влияния через гауссово ядро:

$$K(z) = \frac{1}{\sqrt{2\pi}} exp\left\{-\frac{z^2}{2}\right\}$$

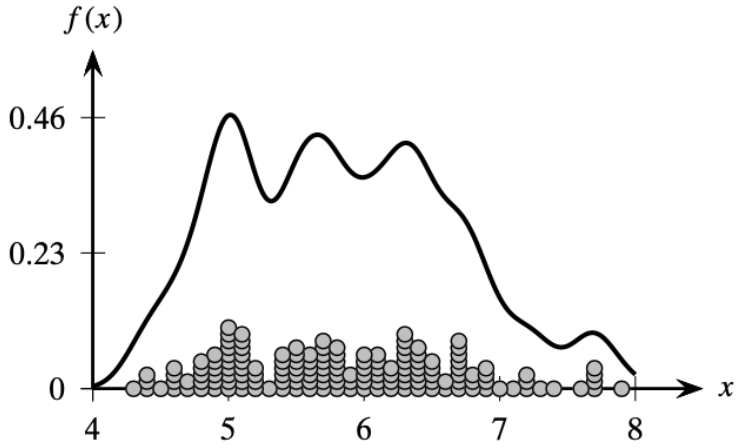
Таким образом, мы имеем$K\left(\frac{x-x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} exp\left\{-\frac{(x-x_i)^2}{2h^2}\right\}$

Здесь x, который находится в центре окна, играет роль среднего, а h действует как стандартное отклонение.

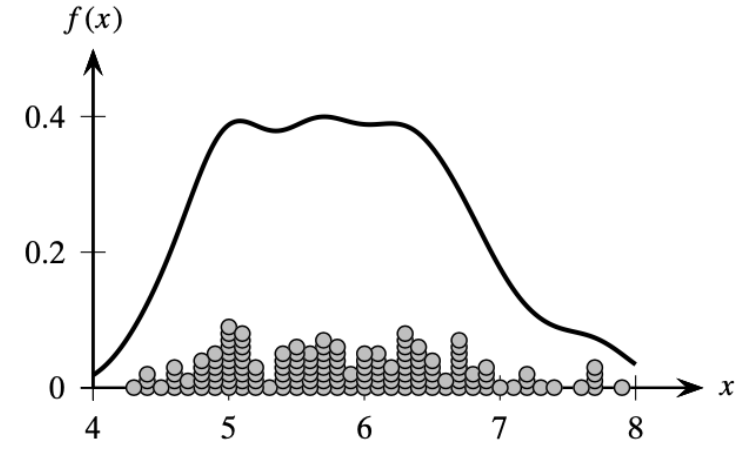
Пример 15.5.



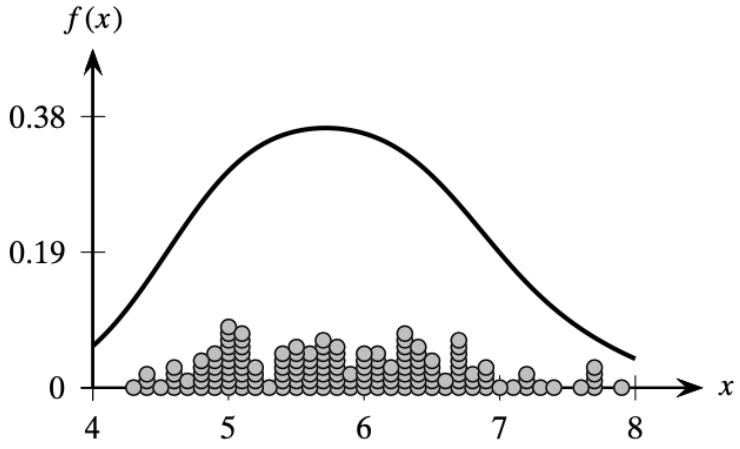
(a) $h = 0.1$



(b) $h = 0.15$



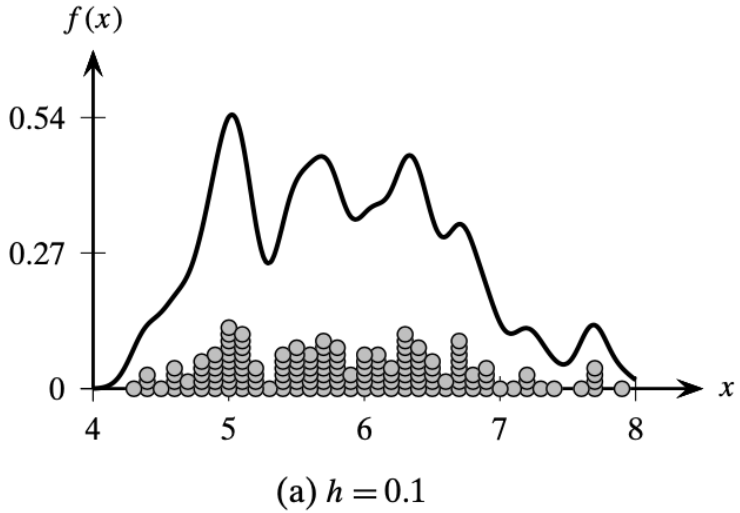
(c) $h = 0.25$



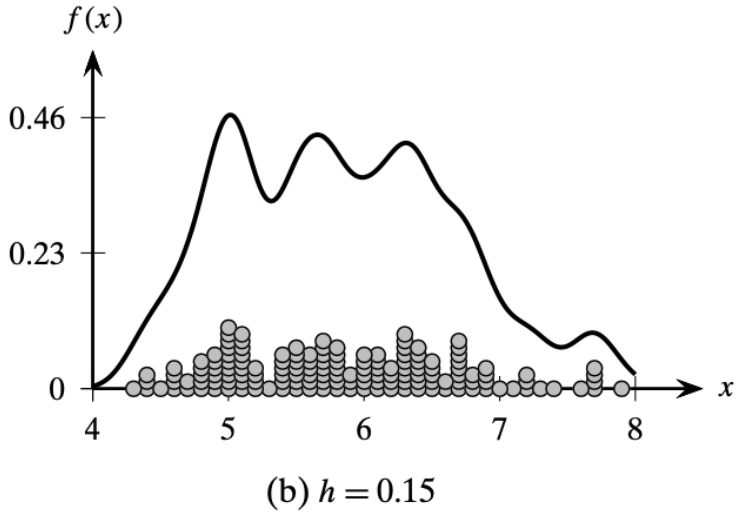
(d) $h = 0.5$

На рис. 15.6 показана одномерная функция плотности для одномерного набора данных Iris (по длине чашелистика) с использованием ядра Гаусса. Показаны графики для возрастающих значений параметра разброса h. Точки данных показаны сгруппированными по оси x с высотой, соответствующей частотам значений.

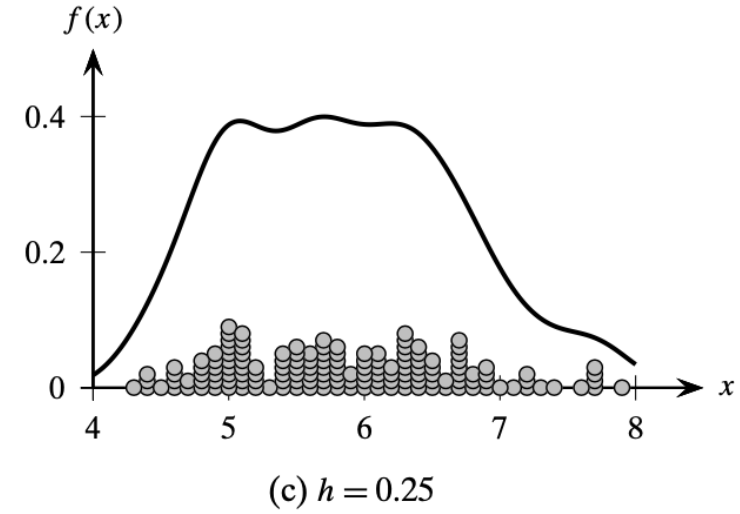
Пример 15.5.



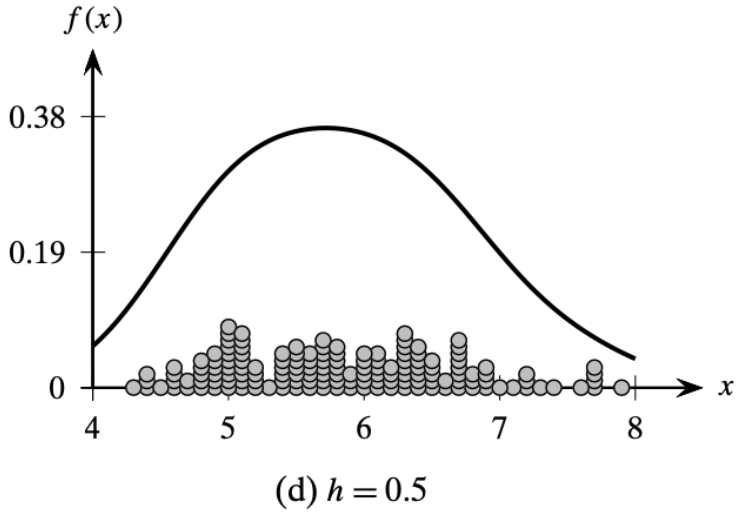
(a) $h = 0.1$



(b) $h = 0.15$



(c) $h = 0.25$



(d) $h = 0.5$

Поскольку h изменяется от 0,1 до 0,5, мы можем видеть сглаживающий эффект увеличения h на функцию плотности. Например, при h = 0,1 имеется много локальных максимумов, а при h = 0,5 - только один пик плотности. По сравнению со случаем дискретного ядра, показанным на рисунке 15.5, мы можем ясно видеть, что ядро Гаусса дает гораздо более гладкие оценки без разрывов.

Многомерная оценка плотности

Чтобы оценить плотность вероятности в d-мерной точке $x = (x_1, x_2, \dots, x_n)^T$, мы определяем d-мерное «окно» как гиперкуб в d измерениях, то есть гиперкуб с центром в x с длиной ребра h. Объем такого d-мерного гиперкуба задается как

$$\mathrm{vol}\big(H_d(h)\big) = h^d$$

Затем плотность оценивается как доля веса точки, лежащая в d-мерном окне с центром в x, деленная на объем гиперкуба:

$$\hat{f}(\boldsymbol{x}) = \frac{1}{nh^2} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x_i}}{h}\right) \quad (15.3)$$

где функция многомерного ядра K удовлетворяет условию $\int K(\boldsymbol{z}) d\boldsymbol{z} = 1$

Дискретное ядро

для любого d-мерного вектора $x = (x_1, x_2, \dots, x_n)^T$ дискретная ядерная функция в d-измерениях задается как

$$K(z) = \begin{cases} 1 \text{ If } |z_j| \leq \frac{1}{2}, \text{для всех размеров } j = 1, \dots, d \\ 0 \text{ В противном случае} \end{cases}$$

Для $z = \frac{x - x_i}{h}$ мы видим, что ядро вычисляет количество точек внутри гиперкуба с центром в x, потому что $K\left(\frac{x - x_i}{h}\right) = 1$, тогда и только тогда $\left|\frac{x_j - x_{ij}}{h}\right| \leq \frac{1}{2}$ для всех размеров j. Таким образом, каждая точка в гиперкубе дает вес $\frac{1}{n}$ для оценки плотности.

Гауссово ядро

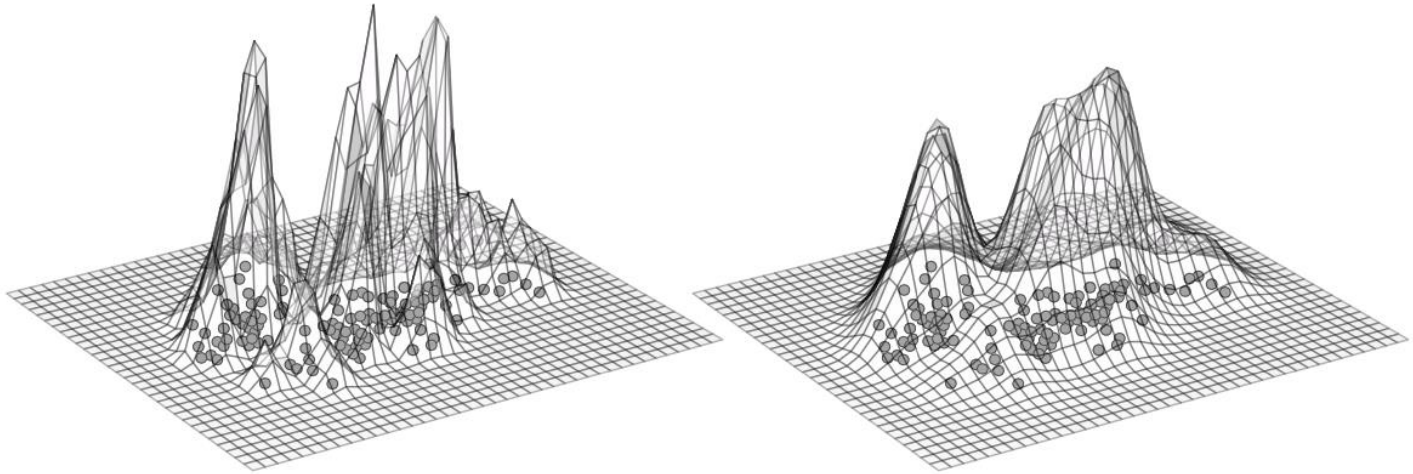
d-мерное гауссово ядро задается как

$$K(\boldsymbol{z}) = \frac{1}{(2\pi)^{\frac{d}{2}}} exp\left\{-\frac{\boldsymbol{z}^T\boldsymbol{z}}{2}\right\} \quad (15.4)$$

где мы предполагаем, что ковариационная матрица является единичной матрицей $d \times d$, то есть $\boldsymbol{\Sigma} = \mathbf{I}_d$. Подставляя $\boldsymbol{z} = \frac{x - x_i}{h}$ в уравнение (15.4), получаем

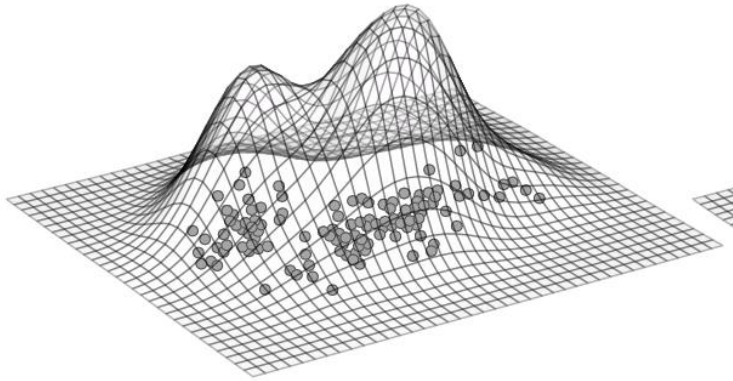
$$K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right) = \frac{1}{(2\pi)^{d/2}} exp\left\{-\frac{(\boldsymbol{x} - \boldsymbol{x}_i)^T(\boldsymbol{x} - \boldsymbol{x}_i)}{2h^2}\right\}$$

Оценка плотности: набор данных 2D Iris (различается по h)

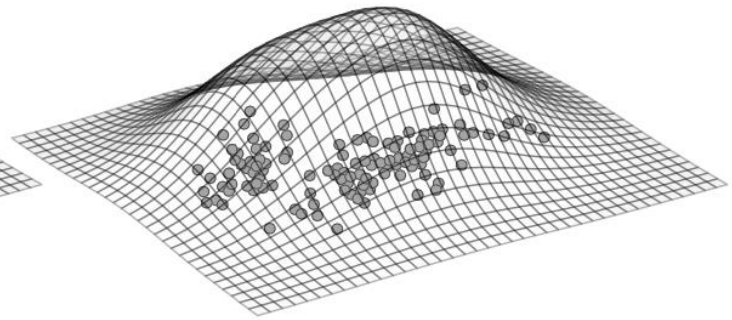


(a) $h = 0.1$

(b) $h = 0.2$

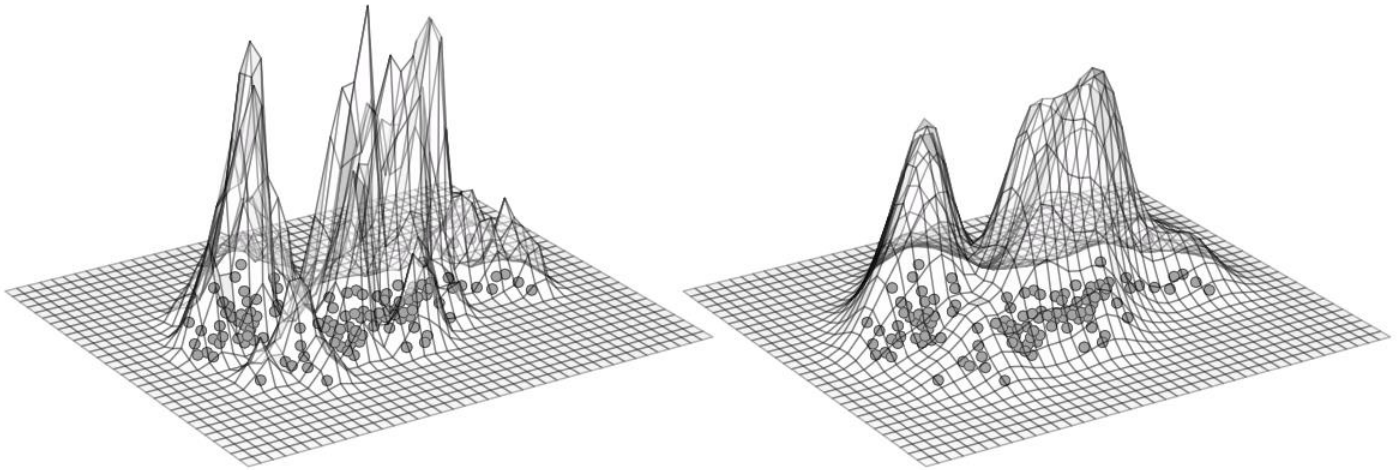


(c) $h = 0.35$



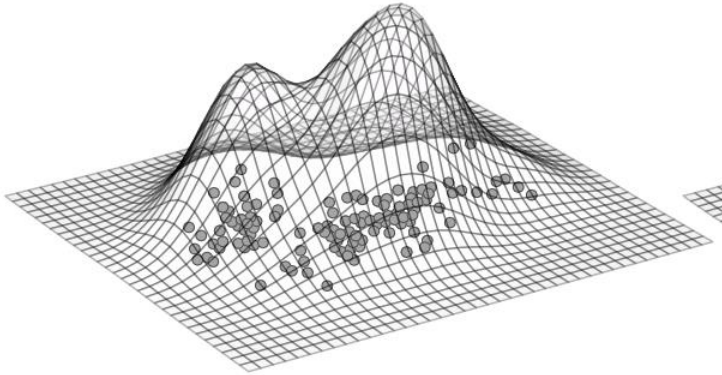
(d) $h = 0.6$

Пример 15.6.

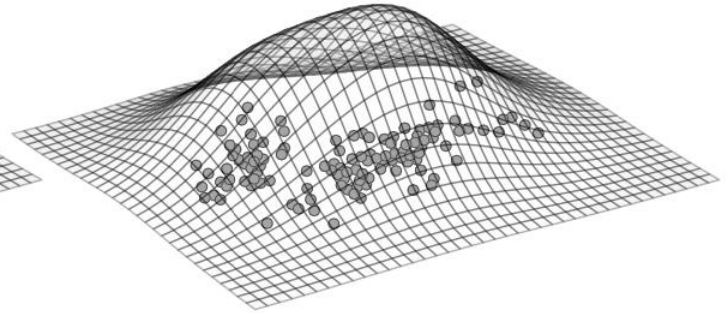


(a) $h = 0.1$

(b) $h = 0.2$



(c) $h = 0.35$

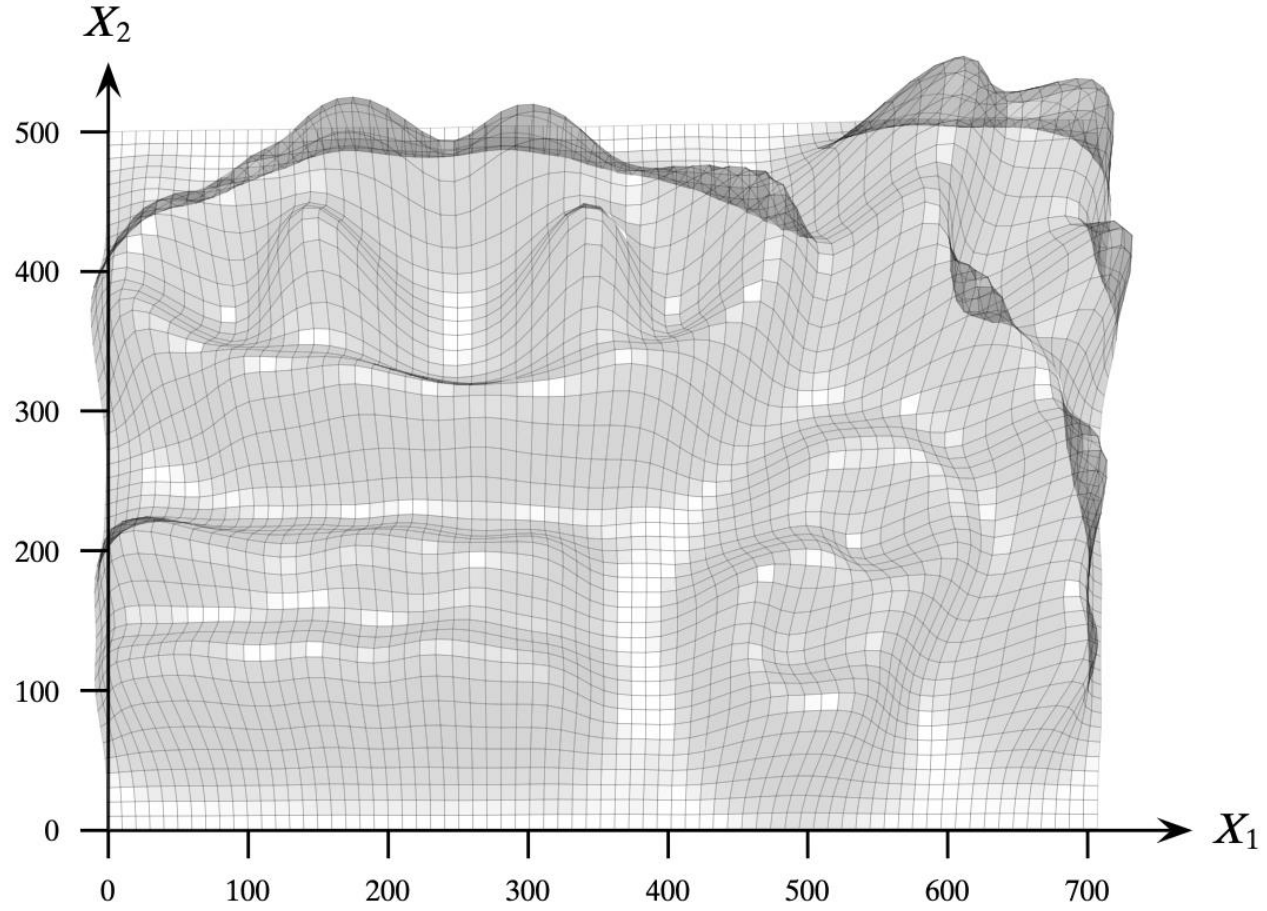


(d) $h = 0.6$

На рисунке 15.7 показана функция плотности вероятности для набора данных 2D радужной оболочки глаза, содержащего атрибуты длины и ширины чашелистика, с использованием ядра Гаусса. Как и ожидалось, для малых значений h функция плотности имеет несколько локальных максимумов, тогда как для больших значений количество максимумов уменьшается, и в конечном итоге для достаточно большого значения мы получаем унимодальное распределение.

Пример 15.7.

На рисунке 15.8 показана оценка плотности ядра для набора данных на основе плотности на рисунке 15.1 с использованием ядра Гаусса с $h = 20$. Можно четко различить, что пики плотности близко соответствуют областям с более высокой плотностью точек.



Оценка плотности ближайшего соседа

Альтернативный подход к оценке плотности состоит в том, чтобы зафиксировать k, количество точек, необходимых для оценки плотности, и позволить изменяться объему окружающей области для размещения этих k точек. Этот подход называется подходом k ближайших соседей (KNN) к оценке плотности. Как и оценка плотности ядра, оценка плотности KNN также является непараметрическим подходом. Учитывая k, количество соседей, мы оцениваем плотность в точке x следующим образом:

$$\hat{f}(\boldsymbol{x}) = \frac{k}{n \operatorname{vol}\left(S_d(h_{\boldsymbol{x}})\right)}$$

где $h_{\boldsymbol{x}}$ – это расстояние от $\boldsymbol{x}$ до k-го ближайшего соседа, а $\operatorname{vol}\left(S_d(h_{\boldsymbol{x}})\right)$ – объем d-мерной гиперсферы $S_d(h_{\boldsymbol{x}})$ с центром в $\boldsymbol{x}$ и радиусом $h_{\boldsymbol{x}}$

КЛАСТЕРИЗАЦИЯ НА ОСНОВЕ ПЛОТНОСТИ: DENCLUE

Аттракторы плотности и градиент

Точка $\boldsymbol{x}^*$ называется аттрактором плотности, если она является локальным максимумом функции плотности вероятности f. Аттрактор плотности можно найти с помощью подхода градиентного подъема, начиная с некоторой точки $\boldsymbol{x}$. Идея состоит в том, чтобы вычислить градиент плотности, направление наибольшего увеличения плотности, и двигаться в направлении градиента небольшими шагами, пока мы не достигнем локальных максимумов.

Градиент в точке $\boldsymbol{x}$ может быть вычислен как многомерная производная оценки плотности вероятности в уравнении (15.3) в виде

$$\nabla \hat{f}(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}} \hat{f}(\boldsymbol{x}) = \frac{1}{nh^2} \sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{x}} K\left(\frac{\boldsymbol{x} - \boldsymbol{x_i}}{h}\right) \quad (15.5)$$

Для ядра Гаусса имеем

$$\frac{\partial}{\partial \boldsymbol{x}} K(\boldsymbol{z}) = \left(\frac{1}{(2\pi)^{\frac{d}{2}}} exp\left\{ -\frac{\boldsymbol{z}^T \boldsymbol{z}}{2} \right\} \right) \cdot -z \cdot \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} = K(\boldsymbol{z}) \cdot -z \cdot \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}$$

Полагая $\boldsymbol{z} = \frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}$ выше, получаем

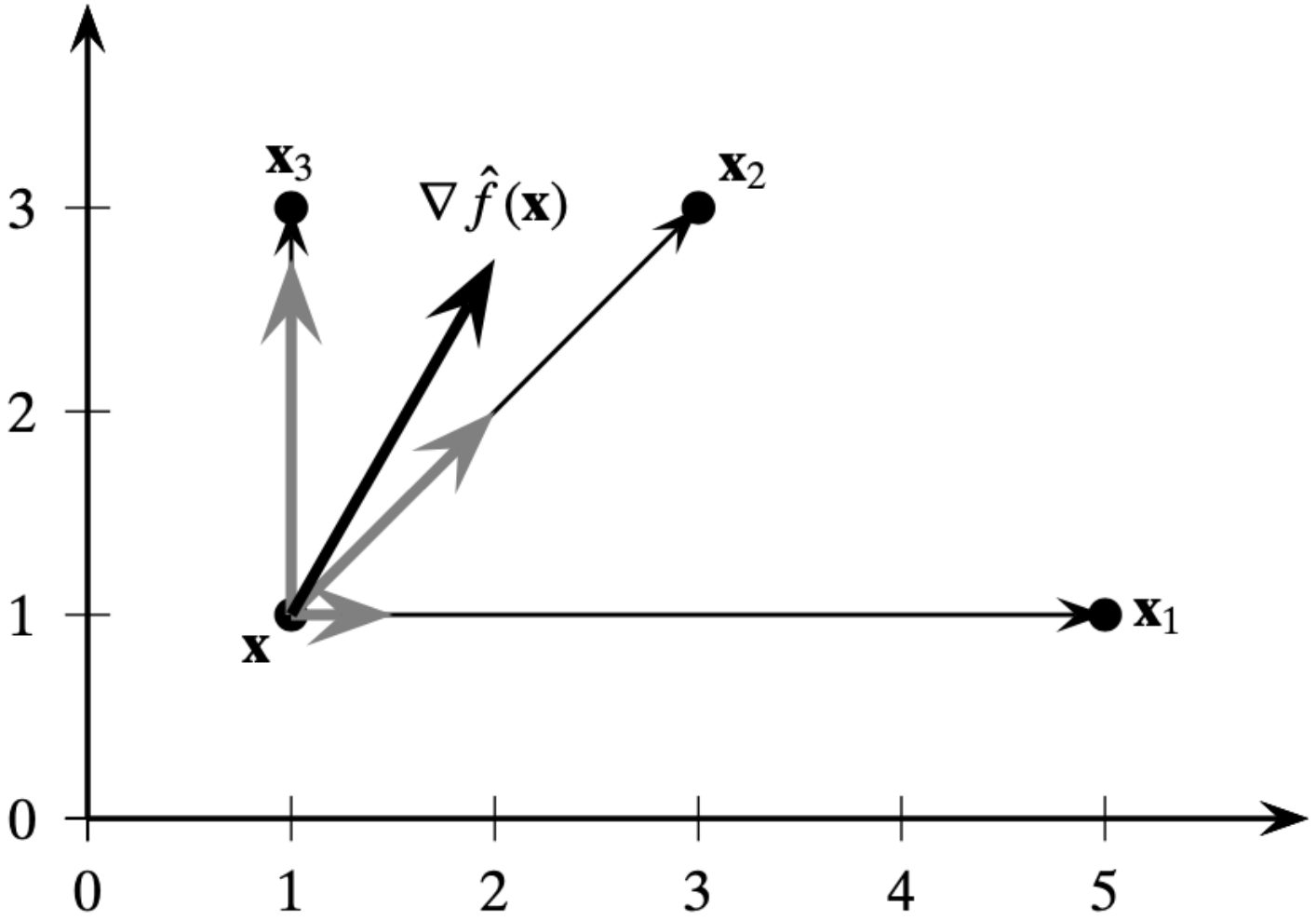
$$\frac{\partial}{\partial \boldsymbol{x}} K\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h} \right) = K\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h} \right) \cdot \left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h} \right) \cdot \left(\frac{1}{h} \right)$$

что следует из того, что $\frac{\partial}{\partial \boldsymbol{x}} \left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h} \right) = \left(\frac{1}{h} \right)$. Подставляя вышеуказанное в формулу (15.5)

градиент в точке $\boldsymbol{x}$ задается как

$$\nabla \hat{f}(\boldsymbol{x}) = \frac{1}{nh^{d+2}} \sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{x}} K\left(\frac{\boldsymbol{x}-\boldsymbol{x_i}}{h} \right) \cdot (\boldsymbol{x}_i - \boldsymbol{x})$$

Это уравнение можно представить как состоящее из двух частей: вектора $(\boldsymbol{x_i} - \boldsymbol{x})$ и скалярного значения *влияния* $K\left(\frac{\boldsymbol{x}-\boldsymbol{x_i}}{h}\right)$. Для каждой точки $\boldsymbol{x_i}$ мы сначала вычисляем направление от $\boldsymbol{x}$, то есть вектор $(\boldsymbol{x_i} - \boldsymbol{x})$. Затем мы масштабируем его, используя значение ядра Гаусса в качестве веса $K\left(\frac{\boldsymbol{x}-\boldsymbol{x_i}}{h}\right)$. Наконец, вектор $\nabla\hat{f}(\boldsymbol{x})$ является чистым влиянием в точке x, как показано на рисунке 15.9, то есть взвешенной суммой векторов разности.



Вектор градиента $\nabla\hat{f}(x)$ (показан толстым черным), полученный как сумма векторов разности $(x_i - x)$ (показан серым).

$\boldsymbol{x}^*$ является аттрактором плотности для $\boldsymbol{x}$ или, альтернативно, что $\boldsymbol{x}$ является плотностью, сходящейся к $\boldsymbol{x}^*$, если процесс восхождения на холм, начатый в $\boldsymbol{x}$, сходится к $\boldsymbol{x}^*$. То есть существует последовательность точек $\boldsymbol{x} = \boldsymbol{x}_0 \to \boldsymbol{x}_1 \to \cdots \to \boldsymbol{x}_m$, начиная с $\boldsymbol{x}$ и заканчивая $\boldsymbol{x}_m$, такое, что $\left|\left|\boldsymbol{x}_m - \boldsymbol{x}^*\right|\right| \leq \epsilon$, то есть $\boldsymbol{x}_m$ сходится к аттрактору $\boldsymbol{x}^*$.

Типичный подход заключается в использовании метода градиентного подъема для вычисления $\boldsymbol{x}^*$, то есть, начиная с $\boldsymbol{x}$, мы итеративно обновляем его на каждом шаге t с помощью правила обновления:

$$x_{t+1} = x_t + \delta \cdot \nabla \hat{f}(\boldsymbol{x})$$

где $\delta > 0$ – размер шага.

подход градиентного подъема может медленно сходиться. Вместо этого можно напрямую оптимизировать направление движения, установив градиент [Ур. (15.6)] к нулевому вектору:

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

$$\frac{1}{nh^{d+2}} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x_i}}{h}\right) \cdot (\boldsymbol{x}_i - \boldsymbol{x}) = 0$$

$$\boldsymbol{x} \cdot \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x_i}}{h}\right) = \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x_i}}{h}\right) \boldsymbol{x}_i$$

$$\boldsymbol{x} = \frac{\sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x_i}}{h}\right) \boldsymbol{x}_i}{\sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x_i}}{h}\right)}$$

Точка $\boldsymbol{x}$ задействована как слева, так и справа выше. Однако его можно использовать для получения следующего правила итеративного обновления:

$$\boldsymbol{x}_{t+1} = \frac{\sum_{i=1}^{n} K\left(\frac{\boldsymbol{x}_t - \boldsymbol{x}_i}{h}\right) \boldsymbol{x}_i}{\sum_{i=1}^{n} K\left(\frac{\boldsymbol{x}_t - \boldsymbol{x}_i}{h}\right)} \quad (15.7)$$

где t обозначает текущую итерацию, а $\boldsymbol{x}_{t+1}$ - обновленное значение для текущего вектора $\boldsymbol{x}_t$. Это правило прямого обновления является, по сути, средневзвешенным влиянием (вычисленным с помощью функции ядра K) каждой точки $\boldsymbol{x}_i \in \mathbf{D}$ на текущую точку $\boldsymbol{x}_t$. Правило прямого обновления приводит к гораздо более быстрой сходимости процесса подъема на холм.

Центральный кластер

Кластер $\boldsymbol{C} \subseteq \mathbf{D}$ называется центральным кластером, если все точки x $\in C$ являются плотностью, сходящейся к единственному аттрактору плотности $\boldsymbol{x}^*$, такому, что $\hat{f}(\boldsymbol{x}^*) \geq \xi$, где ξ – минимальная пороговая плотность, определяемый пользователем. Другими словами,

$$\hat{f}(\boldsymbol{x}^*) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x_i}}{h}\right) \geq \xi$$

Кластер на основе плотности

Кластер $\boldsymbol{C} \subseteq \mathbf{D}$ произвольной формы называется кластером на основе плотности, если существует набор плотностных аттракторов $x_1^*, x_2^*, \ldots, x_m^*$, таких что

1. Каждая точка $\boldsymbol{x} \in \boldsymbol{C}$ притягивается к некоторому аттрактору $\boldsymbol{x}^*$.
2. Каждый аттрактор плотности имеет плотность выше ξ. То есть $\hat{f}(\boldsymbol{x}^*) \geq \xi$.
3. Любые два аттрактора плотности x_i^* и x_j^* достижимы по плотности, то есть существует путь из x_i^* в x_j^* такой, что для всех точек **y** на пути $\hat{f}(\boldsymbol{x}^*) \geq \xi$.

Алгоритм DENCLUE

Псевдокод для DENCLUE показан в алгоритме 15.2. Первым шагом является вычисление аттрактора плотности $\boldsymbol{x}^*$ для каждой точки $\boldsymbol{x}$ в наборе данных (строка 4). Если плотность в $\boldsymbol{x}^*$ выше минимального порога плотности ξ, аттрактор добавляется к набору аттракторов A. Точка данных $\boldsymbol{x}$ также добавляется к набору точек $\boldsymbol{R}(\boldsymbol{x}^*)$, сходящихся к $\boldsymbol{x}^*$ (строка 9).

DENCLUE $(\mathbf{D}, h, \xi, \epsilon)$**:**

1 $\mathcal{A} \leftarrow \emptyset$
2 **foreach** $\mathbf{x} \in \mathbf{D}$ **do** // find density attractors
4 $\quad \mathbf{x}^* \leftarrow \text{FINDATTRACTOR}(\mathbf{x}, \mathbf{D}, h, \epsilon)$
5 $\quad$ **if** $\hat{f}(\mathbf{x}^*) \geq \xi$ **then**
7 $\quad\quad \mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{x}^*\}$
9 $\quad\quad R(\mathbf{x}^*) \leftarrow R(\mathbf{x}^*) \cup \{\mathbf{x}\}$
11 $\mathcal{C} \leftarrow \{\text{maximal } C \subseteq \mathcal{A} \mid \forall \mathbf{x}_i^*, \mathbf{x}_j^* \in C, \mathbf{x}_i^* \text{ and } \mathbf{x}_j^* \text{ are density reachable}\}$
12 **foreach** $C \in \mathcal{C}$ **do** // density-based clusters
13 $\quad$ **foreach** $\mathbf{x}^* \in C$ **do** $C \leftarrow C \cup R(\mathbf{x}^*)$
14 **return** $\mathcal{C}$

FINDATTRACTOR $(\mathbf{x}, \mathbf{D}, h, \epsilon)$**:**

16 $t \leftarrow 0$
17 $\mathbf{x}_t \leftarrow \mathbf{x}$
18 **repeat**
20 $\quad \mathbf{x}_{t+1} \leftarrow \frac{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right) \cdot \mathbf{x}_t}{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right)}$
21 $\quad t \leftarrow t + 1$
22 **until** $\|\mathbf{x}_t - \mathbf{x}_{t-1}\| \leq \epsilon$
24 **return** $\mathbf{x}_t$

Алгоритм DENCLUE

На втором этапе DENCLUE находит все максимальные подмножества аттракторов $\boldsymbol{C} \subseteq A$, такие, что любая пара аттракторов в $\boldsymbol{C}$ является достижимыми по плотности друг из друга (строка 11). Эти максимальные подмножества взаимно достижимых аттракторов образуют начало для каждого кластера на основе плотности. Наконец, для каждого аттрактора $\boldsymbol{x}^* \in \boldsymbol{C}$ мы добавляем к кластеру все точки $\boldsymbol{R}(\boldsymbol{x}^*)$, которые сходятся к $\boldsymbol{x}^*$, что приводит к окончательному набору кластеров $\boldsymbol{C}$.

DENCLUE $(\mathbf{D}, h, \xi, \epsilon)$:

1 $\mathcal{A} \leftarrow \emptyset$

2 **foreach** $\mathbf{x} \in \mathbf{D}$ **do** // find density attractors

4 $\quad \mathbf{x}^* \leftarrow \text{FINDATTRACTOR}(\mathbf{x}, \mathbf{D}, h, \epsilon)$

5 $\quad$ **if** $\hat{f}(\mathbf{x}^*) \geq \xi$ **then**

7 $\quad\quad \mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{x}^*\}$

9 $\quad\quad R(\mathbf{x}^*) \leftarrow R(\mathbf{x}^*) \cup \{\mathbf{x}\}$

11 $\mathcal{C} \leftarrow \{\text{maximal } C \subseteq \mathcal{A} \mid \forall \mathbf{x}_i^*, \mathbf{x}_j^* \in C, \mathbf{x}_i^* \text{ and } \mathbf{x}_j^* \text{ are density reachable}\}$

12 **foreach** $C \in \mathcal{C}$ **do** // density-based clusters

13 $\quad$ **foreach** $\mathbf{x}^* \in C$ **do** $C \leftarrow C \cup R(\mathbf{x}^*)$

14 **return** $\mathcal{C}$

FINDATTRACTOR $(\mathbf{x}, \mathbf{D}, h, \epsilon)$:

16 $t \leftarrow 0$

17 $\mathbf{x}_t \leftarrow \mathbf{x}$

18 **repeat**

20 $\quad \mathbf{x}_{t+1} \leftarrow \dfrac{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right) \cdot \mathbf{x}_t}{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right)}$

21 $\quad t \leftarrow t + 1$

22 **until** $\|\mathbf{x}_t - \mathbf{x}_{t-1}\| \leq \epsilon$

24 **return** $\mathbf{x}_t$

Алгоритм DENCLUE

Метод FINDATTRACTOR реализует процесс восхождения с использованием правила прямого обновления [Ур. (15.7)], что приводит к быстрой сходимости. Чтобы еще больше ускорить вычисление влияния, можно вычислить значения ядра только для ближайших соседей $\boldsymbol{x}_t$. То есть мы можем проиндексировать точки в наборе данных **D**, используя структуру пространственного индекса, чтобы мы могли быстро вычислить всех ближайших соседей $\boldsymbol{x}_t$ в пределах некоторого радиуса r.

DENCLUE $(\mathbf{D}, h, \xi, \epsilon)$:

$\mathcal{A} \leftarrow \emptyset$

foreach $\mathbf{x} \in \mathbf{D}$ **do** // find density attractors

  $\mathbf{x}^* \leftarrow \text{FINDATTRACTOR}(\mathbf{x}, \mathbf{D}, h, \epsilon)$

  if $\hat{f}(\mathbf{x}^*) \geq \xi$ **then**

    $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{x}^*\}$

    $R(\mathbf{x}^*) \leftarrow R(\mathbf{x}^*) \cup \{\mathbf{x}\}$

$\mathcal{C} \leftarrow \{\text{maximal } C \subseteq \mathcal{A} \mid \forall \mathbf{x}_i^*, \mathbf{x}_j^* \in C, \mathbf{x}_i^* \text{ and } \mathbf{x}_j^* \text{ are density reachable}\}$

foreach $C \in \mathcal{C}$ **do** // density-based clusters

  foreach $\mathbf{x}^* \in C$ **do** $C \leftarrow C \cup R(\mathbf{x}^*)$

return $\mathcal{C}$

FINDATTRACTOR $(\mathbf{x}, \mathbf{D}, h, \epsilon)$:

$t \leftarrow 0$

$\mathbf{x}_t \leftarrow \mathbf{x}$

repeat

  $\mathbf{x}_{t+1} \leftarrow \frac{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right) \cdot \mathbf{x}_t}{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right)}$

  $t \leftarrow t + 1$

until $\|\mathbf{x}_t - \mathbf{x}_{t-1}\| \leq \epsilon$

return $\mathbf{x}_t$

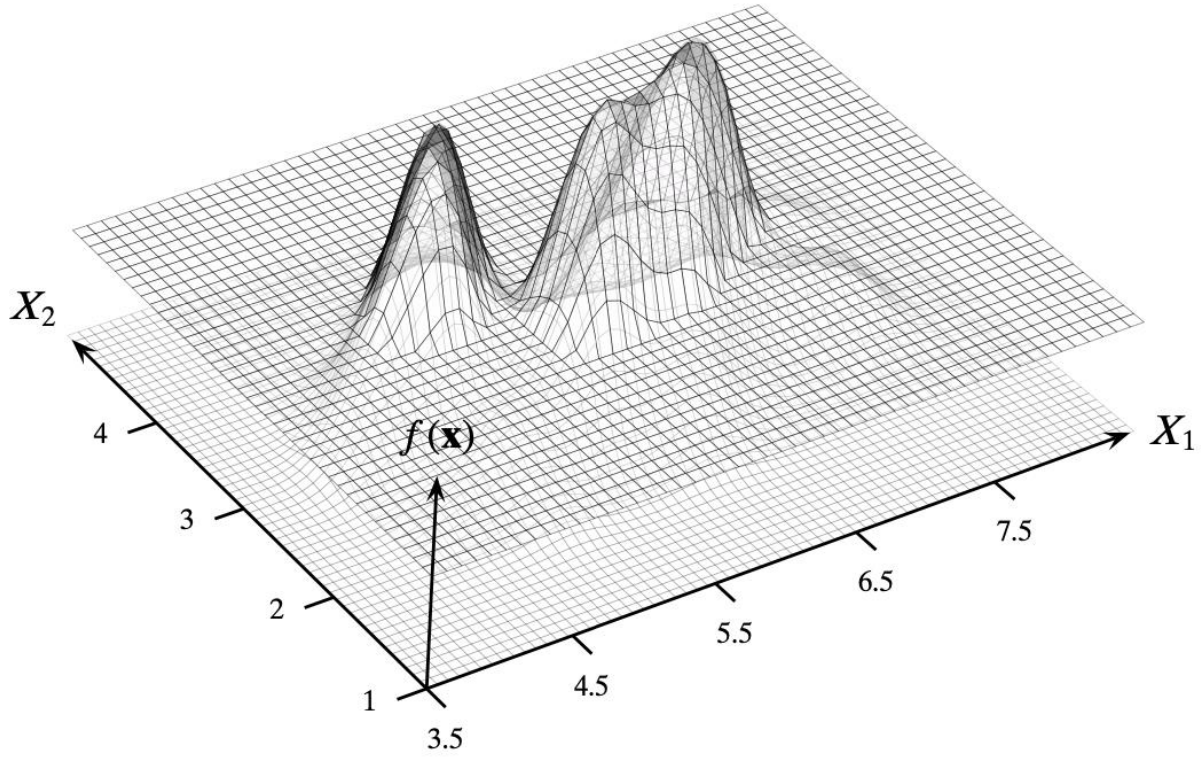
Для ядра Гаусса мы можем установить $r = h \cdot z$, где h - параметр влияния, который играет роль стандартного отклонения, а z указывает количество стандартных отклонений. Пусть $B_d(\boldsymbol{x}_t, r)$ обозначает множество всех точек в **D**, которые лежат внутри d-мерного шара радиуса r с центром в $\boldsymbol{x}_t$. Правило обновления на основе ближайшего соседа может быть выражено как

$$\boldsymbol{x}_{t+1} = \frac{\sum_{\boldsymbol{x}_i \in B_d(\boldsymbol{x}_t, r)} K\left(\frac{\boldsymbol{x}_t - \boldsymbol{x}_i}{h}\right) \boldsymbol{x}_i}{\sum_{\boldsymbol{x}_i \in B_d(\boldsymbol{x}_t, r)} K\left(\frac{\boldsymbol{x}_t - \boldsymbol{x}_i}{h}\right)}$$

который можно использовать в строке 20 алгоритма 15.2.

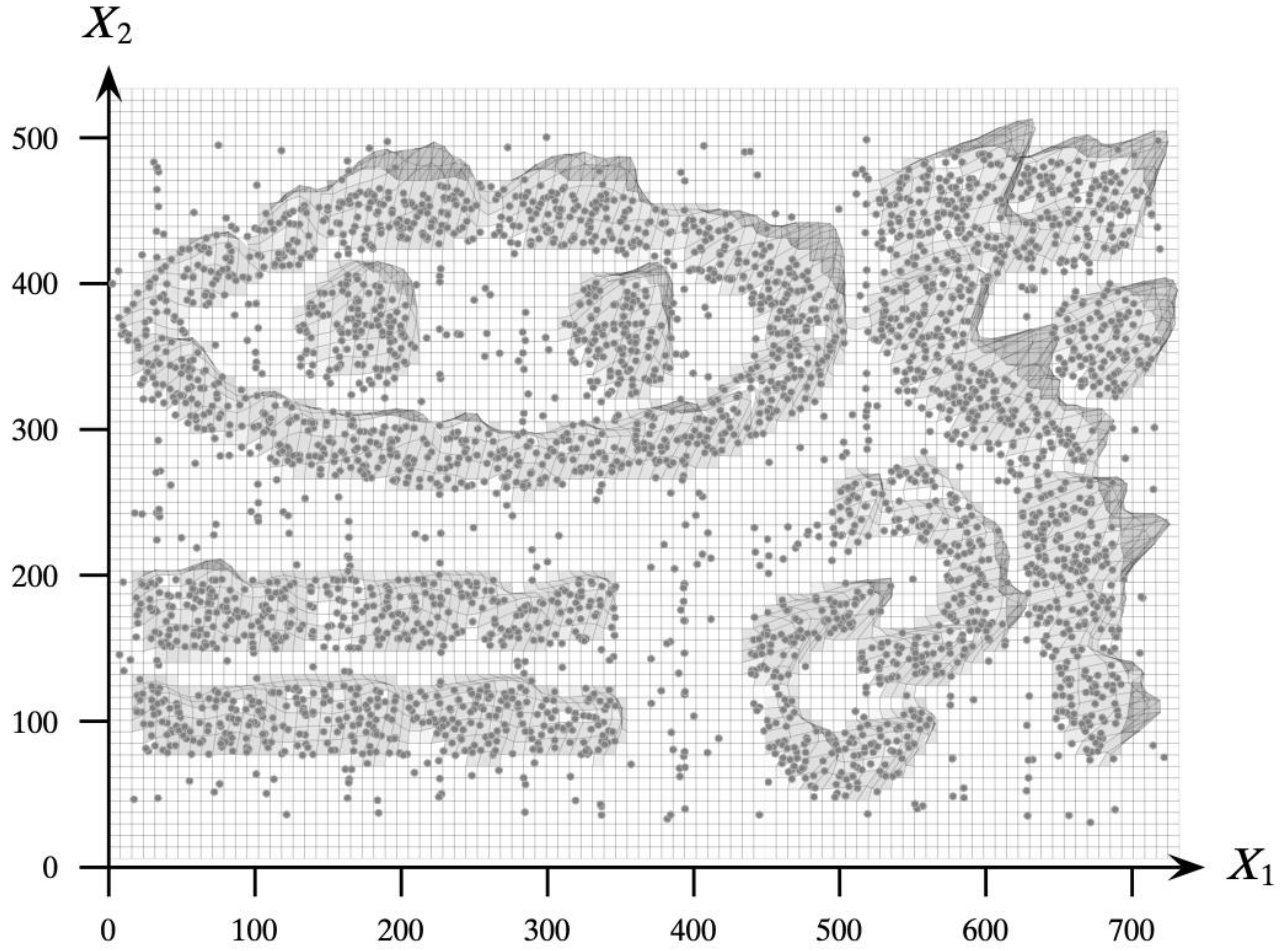
Пример 15.8

На рисунке 15.10 показана кластеризация DENCLUE для двухмерного набора данных Iris, включающего атрибуты длины и ширины чашелистника. Результаты были получены при $h = 0{,}2$ и $\xi = 0{,}08$ с использованием ядра Гаусса. Кластеризация получается путем определения порога функции плотности вероятности на рисунке 15.7b при $\xi = 0{,}08$. Два пика соответствуют двум последним кластерам. В то время как *iris setosa* хорошо разделена, трудно разделить два других типа *Ирисов*.



Пример 15.9

На рисунке 15.11 показаны кластеры, полученные с помощью DENCLUE на основе набора данных по плотности из рисунка 15.1. Используя параметры h = 10 и $\xi = 9.5 \times 10^{-5}$, с гауссовым ядром, мы получаем восемь кластеров. Рисунок получен путем разрезания функции плотности на значение плотности ξ; отображаются только области выше этого значения. Все кластеры идентифицированы правильно, за исключением двух полукруглых кластеров в правом нижнем углу, которые кажутся объединенными в один кластер.



DENCLUE: Особые случаи

Можно показать, что DBSCAN является частным случаем подхода к кластеризации на основе общей оценки плотности ядра, DENCLUE. Если мы положим $h = \in$ и $\xi = minpts$, то использование дискретного ядра DENCLUE дает точно такие же кластеры, как DBSCAN. Каждый аттрактор плотности соответствует центральной точке, а набор связанных основных точек определяет аттракторы кластера на основе плотности. Также можно показать, что K-средних является частным случаем кластеризации на основе плотности для соответствующих значений h и ξ, с аттракторами плотности, соответствующими центроидам кластеров. Кроме того, стоит отметить, что подход на основе плотности может создавать иерархические кластеры путем изменения порога ξ. Например, уменьшение ξ может привести к слиянию нескольких кластеров, обнаруженных при более высоких значениях порогов. В то же время это также может приводить к новым кластерам, если пиковая плотность удовлетворяет более низкому значению ξ.

Вычислительная сложность

Время выполнения для DENCLUE является наименьшим из-за стоимости процесса восхождения. Для каждой точки $\boldsymbol{x} \in \mathbf{D}$ поиск аттрактора плотности занимает время $O(nt)$, где t - максимальное количество итераций подъема на холм. Это связано с тем, что каждая итерация занимает $O(n)$ времени для вычисления суммы функции влияния по всем точкам $\boldsymbol{x}_i \in$ D. Таким образом, общая стоимость вычисления аттракторов плотности составляет $O(n^2 t)$. Мы предполагаем, что при разумных значениях h и ξ имеется всего несколько аттракторов плотности, то есть $|A| = m \ll n$. Стоимость поиска максимально достижимых подмножеств аттракторов составляет $O(m^2)$, а конечные кластеры могут быть получены за $O(n)$ время.

Критерии достоверности и статистики кластеризации

Внешний: внешние меры проверки используют критерии, которые не являются неотъемлемыми для набора данных. Это может быть в форме предшествующих или определенных экспертами знаний о кластерах, например, меток классов для каждой точки.

Внутренний: меры внутренней проверки используют критерии, которые выводятся из самих данных. Например, мы можем использовать внутрикластерные и межкластерные расстояния для получения показателей компактности кластера (например, насколько похожи точки в одном кластере?) И разделения (например, насколько далеко друг от друга находятся точки в разных кластерах?).

Относительный: меры относительной проверки направлены на прямое сравнение различных кластеров, обычно полученных с помощью разных настроек параметров для одного и того же алгоритма.

ВНЕШНИЕ МЕРЫ

Пусть $\boldsymbol{D} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ - набор данных, состоящий из n точек в d-мерном пространстве, разбитых на k кластеров. Пусть $y_i \in \{1, 2, \dots, k\}$ обозначают истинную кластеризацию или информацию о метке кластера для каждой точки. Истинная кластеризация задается как $T = \{T_1, T_2, \dots, T_k\}$, где кластер T_j состоит из всех точек с меткой j, т.е. $T_j = \{\boldsymbol{x}_i \in \boldsymbol{D} | \, y_i = j\}$. Кроме того, пусть $C = \{C_1, C_2, \dots, C_r\}$ обозначает кластеризацию того же набора данных в r кластеров, полученных с помощью некоторого алгоритма кластеризации, и пусть $\hat{y}_i \in \{1, 2, \dots, r\}$ обозначает метку кластера для $\boldsymbol{x}_i$.

Для ясности в дальнейшем мы будем называть T истинным *разделением*, а каждый T_i – разделом. Мы будем называть C кластеризацией, причем каждый C_i назовем кластером. Поскольку предполагается, что достоверная информация известна, обычно методы кластеризации будут выполняться с правильным числом кластеров, то есть с $r = k$. Однако, чтобы сохранить более общее обсуждение, мы позволяем r отличаться от k.

Меры внешней оценки пытаются зафиксировать степень, в которой точки из одного раздела появляются в одном кластере, и степень, в которой точки из разных разделов сгруппированы в разные кластеры. Обычно существует компромисс между этими двумя целями, который либо явно фиксируется мерой, либо подразумевается при ее вычислении. Все внешние меры полагаются на $r \times k$ *таблицу сопряженности* $\mathbf{N}$, которая индуцируется кластеризацией C и истинным разделением T, определяемая следующим образом

$$\boldsymbol{N}(i,j) = n_{ij} = |C_i \cap T_j|$$

$$\boldsymbol{N}(i,j) = n_{ij} = |C_i \cap T_j|$$

Другими словами, счетчик n_{ij} обозначает количество точек, общих для кластера C_i и истинного разделения $\boldsymbol{T}_j$. Далее, для ясности положим $n_i = |C_i|$ обозначает количество точек в кластере C_i, и пусть $m_j = |T_j|$ обозначает количество точек в разбиении T_j. Таблица сопряженности может быть вычислена из T и C за время $O(n)$, исследуя метки разделов и кластеров, y_i и $\widehat{y_i}$, для каждой точки $\boldsymbol{x}_i \in \boldsymbol{D}$ и увеличивая соответствующий счетчик $n_{y_i\widehat{y_i}}$.

Соответствующие меры

Чистота определяет степень, в которой кластер C_i содержит объекты только из одного раздела. Другими словами, он измеряет, насколько «чистым» является каждый кластер. Чистота кластера C_i определяется как

$$purity_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\}$$

Чистота кластеризации C определяется как взвешенная сумма значений чистоты по кластерам:

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} \max_{j=1}^{k} \{n_{ij}\}$$

где отношение n_i обозначает долю точек в кластере C_i. Чем выше чистота C, тем лучше согласие с истинным разделением.

Чистота

Максимальное значение чистоты - 1, когда каждый кластер содержит точки только из одного раздела. Когда $r = k$, значение чистоты 1 указывает на идеальную кластеризацию с взаимно-однозначным соответствием между кластерами и разделами.

Однако чистота может быть равна 1 даже при $r > k$, когда каждый из кластеров является подмножеством разбиения на основе истинности. Когда $r < k$, чистота никогда не может быть 1, потому что хотя бы один кластер должен содержать точки из более чем одного раздела.

Максимальное соответствие

Мера максимального соответствия выбирает отображение между кластерами и разделами, так что сумма числа общих точек (n_{ij}) максимизируется, при условии, что только один кластер может соответствовать заданному разделу.

Мы рассматриваем таблицу сопряженности как полный взвешенный двудольный граф $G = (V, E)$, где каждое разбиение и кластер является узлом, то есть $V = C \cup T$, и существует ребро $(C_i, T_j) \in E$ с весом $w(C_i, T_j) = n_{ij}$ для всех $C_i \in C$ и $T_j \in T$. Совпадение M в G – это подмножество E, такое, что ребра в M попарно несмежные, то есть они не имеют общей вершины. Максимальная мера соответствия определяется как *максимальное соответствие веса* в G:

$$match = \underset{M}{argmax} \left\{ \frac{w(M)}{n} \right\}$$

где вес совпадающего M – это просто сумма всех весов ребер в M, заданная как $w(M) = \sum_{e \in M} w(e)$.

Максимальное соответствие

$$match = \underset{M}{argmax} \left\{ \frac{w(M)}{n} \right\}$$

Максимальное совпадение может быть вычислено за время $O\left(|V|^2 \cdot |E|\right) = O((r + k)^2 rk)$, что эквивалентно $O(k^4)$, если $r = O(k)$.

F-Мера

Учитывая кластер C_i, пусть j_i обозначает разбиение, содержащее максимальное количество точек из C_i, то есть $j_i = max^{k}_{j=1}\{n_{ij}\}$. *Точность* кластера C_i такая же, как и его чистота:

$$prec_i = \frac{1}{n_i} \underset{j=1}{\overset{k}{max}} \{n_{ij}\} = \frac{n_{ij_i}}{n_i}$$

Он измеряет долю точек в C_i от мажоритарного разбиения T_{ij}.

Отзыв кластера C_i определяется как

$$recall_i = \frac{n_{ij_i}}{|T_{j_i}|} = \frac{n_{ij_i}}{m_{j_i}}$$

где $m_{j_i} = |T_{j_i}|$. Это измеряет долю точки в разделе T_{j_i}, совместно используемую с кластером C_i.

F-Мера

F-мера – это среднее гармоническое значение точности и отзыва для каждого кластера. Следовательно, F-мера для кластера C_i имеет вид

$$F_i = \frac{2}{\frac{1}{prec_i} + \frac{1}{recall_i}} = \frac{2 \cdot prec_i \cdot recall_i}{prec_i + recall_i} = \frac{2n_{ij_i}}{n_i + m_{j_i}} \quad (17.1)$$

F-мера для кластеризации C – это среднее значение F-меры по кластерам:

$$F = \frac{1}{r}\sum_{i=1}^{r} F_i$$

Таким образом, F-мера пытается сбалансировать точность и отзыв значения по всем кластерам. Для идеальной кластеризации, когда $r = k$, максимальное значение F-меры равно 1

Меры, основанные на энтропии

Условная энтропия

Энтропия кластеризации C определяется как

$$H(C) = -\sum_{i=1}^{r} p_{C_i} \log p_{C_i}$$

где $p_{C_i} = \frac{n_i}{n}$ – вероятность кластера C_i. Аналогично энтропия разбиения T определяется как

$$H(T) = -\sum_{j=1}^{k} p_{T_j} \log p_{T_j}$$

где p_{T_j} – вероятность разбиения T_j.

Специфическая для кластера энтропия T, то есть условная энтропия T относительно кластера C_i, определяется как

$$H(T|C_i) = -\sum_{j=1}^{k} \left(\frac{n_{ij}}{n_i}\right) \log\left(\frac{n_{ij}}{n_i}\right)$$

Условная энтропия T с учетом кластеризации C затем определяется как взвешенная сумма:

$$H(T|C) = \sum_{i=1}^{r} \frac{n_i}{n} H(T|C_i) = -\sum_{i=1}^{r}\sum_{j=1}^{k} \frac{n_{ij}}{n} \log\left(\frac{n_{ij}}{n_i}\right) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log\left(\frac{p_{ij}}{p_{C_i}}\right) \quad (17.2)$$

где $p_{ij} = \frac{n_{ij}}{n}$ – вероятность того, что точка в кластере i также принадлежит разделу j.

Чем больше членов кластера разбито на различные разделы, тем выше условная энтропия. Для идеальной кластеризации значение условной энтропии равно нулю, тогда как наихудшее возможное значение условной энтропии – $\log k$. Далее, расширяя уравнение (17.2), мы видим, что

$$H(T|C) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\left(\log p_{ij} - \log p_{C_i}\right) = -\left(\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log p_{ij}\right) + \sum_{i=1}^{r}\left(\log p_{C_i}\sum_{j=1}^{k} p_{ij}\right) =$$

$$= -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log p_{ij} + \sum_{j=1}^{k} p_{C_i}\log p_{C_i} = H(C,T) - H(C) \quad (17.3)$$

где $H(T|C) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log p_{ij}$ – совместная энтропия C и T.

Нормализованная Взаимная Информация

Взаимная информация пытается количественно определить количество общей информации между кластеризацией C и разделением T, и она определяется как

$$I(C,T) = \sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log\left(\frac{p_{ij}}{p_{C_i} \cdot p_{T_j}}\right) \quad (17.4)$$

Он измеряет зависимость между наблюдаемой совместной вероятностью p_{ij} C и T и ожидаемой совместной вероятностью $p_{C_i} \cdot p_{T_j}$ в предположении независимости. Когда C и T являются независимыми тогда $p_{ij} = p_{C_i} \cdot p_{T_j}$, и, таким образом, $I(C,T) = 0$. Однако верхней границы взаимной информации не существует.

Заметим, что $I(C,T) = H(C) + H(T) - H(C,T)$. Используя уравнение (17.3), получаем:

$$I(C,T) = H(T) - H(T|C)$$

$$I(C,T) = H(C) - H(C|T)$$

Поскольку $H(C|T) \geq 0$ и $H(T|C) \geq 0$, имеем неравенства $I(C,T) \leq H(C)$ и $I(C,T) \leq H(T)$. Мы можем получить нормализованную версию взаимной информации, рассматривая отношения $I(C,T)$ / $H(C)$ и $I(C,T)$ / $H(T)$, оба из которых могут превзойти большинство из них. *Нормализованная взаимная информация* (NMI) определяется как среднее геометрическое этих двух соотношений:

$$NMI(C,T) = \sqrt{\frac{I(C,T)}{H(C)} \cdot \frac{I(C,T)}{H(T)}} = \frac{I(C,T)}{\sqrt{H(C) \cdot H(T)}}$$

Значение NMI находится в диапазоне [0,1]. Значения, близкие к 1, указывают на хорошую кластеризацию.

Вариация информации

Этот критерий основан на взаимной информации между кластеризацией C и истинным разбиением T и их энтропии; это определяется как

$$VI(C,T) = \big(H(T) - I(C,T)\big) + \big(H(C) - I(C,T)\big) = H(T) + H(C) - 2I(C,T) \quad (17.5)$$

Вариация информации (VI) равна нулю только тогда, когда C и T идентичны. Таким образом, чем ниже значение VI, тем лучше кластеризация C.

Используя эквивалентность $I(C,T) = H(T) - H(T|C) = H(C) - H(C|T)$, мы также можем выразить уравнение (17.5) как

$$VI(C,T) = H(T|C) + H(C|T)$$

Наконец, учитывая, что $H(T|C) = H(T|C) - H(C)$, другое выражение для VI дается как

$$VI(C,T) = 2H(T,C) - H(T) - H(T)$$

Парные меры

Учитывая кластеризацию C и исходное разбиение T, парные меры используют информацию о разделах и метках кластера по всем парам точек. Пусть $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{D}$ - любые две точки, $i \neq j$.

Пусть y_i обозначает истинную метку разбиения, а $\hat{y}_i$ обозначает метку кластера для точки $\boldsymbol{x}_i$.

Если и $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$ принадлежат одному кластеру, то есть $\hat{y}_i = \hat{y}_j$, мы называем это *положительным* событием, а если они не принадлежат одному кластеру, то есть $\hat{y}_i \neq \hat{y}_j$, мы называем это *отрицательным* событием.

В зависимости от того, согласованы ли метки кластера и метки разделов, можно рассмотреть четыре возможности:

- Истинно положительные значения: $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$ принадлежат одному и тому же разделу в T, и они также находятся в одном кластере в C. Это истинно положительная пара, потому что положительное событие, $\hat{y}_i = \hat{y}_j$, соответствует основной истине, $y_i = y_j$. Количество истинно положительных пар определяется как

$$TP = \left|\left\{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) : y_i = y_j \; and \; \hat{y}_i = \hat{y}_j\right\}\right|$$

- Ложно отрицательные: $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$ принадлежат одному и тому же разделу в T, но они не принадлежат одному и тому же кластеру в C. То есть отрицательное событие, $\hat{y}_i \neq \hat{y}_j$, не соответствует истине, $y_i = y_j$. Таким образом, эта пара является ложноотрицательной, а количество всех ложноотрицательных пар определяется как

$$FN = \left|\left\{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) : y_i \neq y_j \; and \; \hat{y}_i \neq \hat{y}_j\right\}\right|$$

- Ложно положительные: $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$ не принадлежат одному и тому же разделу в T, но они принадлежат одному и тому же кластеру в C. Эта пара является ложноположительной, потому что положительное событие $\hat{y}_i = \hat{y}_j$ на самом деле ложно, т. Е. не согласуется с разделением основной истины, которое указывает, что $y_i \neq y_j$. Количество ложноположительных пар определяется как

$$FP = \left|\left\{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) : y_i \neq y_j \; and \; \hat{y}_i = \hat{y}_j\right\}\right|$$

- Истинно отрицательные: $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$ не принадлежат одному и тому же разделу в T, и они не принадлежат одному и тому же кластеру в C. Таким образом, эта пара является истинно отрицательным, то есть $\hat{y}_i \neq \hat{y}_j$ и $y_i \neq y_j$. Количество таких истинно отрицательных пар определяется как

$$TN = \left|\left\{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) : y_i \neq y_j \; and \; \hat{y}_i \neq \hat{y}_j\right\}\right|$$

Поскольку имеется $N = \binom{n}{2} = \frac{n(n-1)}{2}$ пар точек, мы имеем следующую идентичность:

$$N = TP + FN + FP + TN \quad (17.6)$$

Наивное вычисление предыдущих четырех случаев времени $O(n^2)$. Однако их можно вычислить более эффективно, используя таблицу сопряженности $\boldsymbol{N} = \{n_{ij}\}$, где $1 \leq i \leq r$ и $1 \leq j \leq k$. Количество истинных положительных результатов определяется как

$$TP = \sum_{i=1}^{r}\sum_{j=1}^{k}\binom{n_{ij}}{2} = \sum_{i=1}^{r}\sum_{j=1}^{k}\frac{n_{ij}(n_{ij}-1)}{2} = \frac{1}{2}\left(\sum_{i=1}^{r}\sum_{j=1}^{k}n_{ij}^2 - \sum_{i=1}^{r}\sum_{j=1}^{k}n_{ij}\right) =$$

$$= \frac{1}{2}\left(\left(\sum_{i=1}^{r}\sum_{j=1}^{k}n_{ij}^2\right) - n\right) \quad (17.7)$$

Чтобы вычислить общее количество ложноотрицательных результатов, мы удаляем количество истинных положительных результатов из числа пар, принадлежащих одному разделу. Поскольку две точки $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$, принадлежащие одному и тому же разделу, имеют $y_i = y_j$, если мы удалим истинные положительные точки, то есть пары с $\hat{y}_i = \hat{y}_j$, у нас останутся пары, для которых $\hat{y}_i \neq \hat{y}_j$, то есть ложно негативные. Таким образом, мы имеем

$$FN = \sum_{j=1}^{k} \binom{m_j}{2} - TP = \frac{1}{2}\left(\sum_{j=1}^{k} m_j^2 - \sum_{j=1}^{k} m_j - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2 + n\right) =$$

$$= \frac{1}{2}\left(\sum_{j=1}^{k} m_j^2 - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) \quad (17.8)$$

Количество ложных срабатываний можно получить аналогичным образом, вычтя количество истинных срабатываний из количества пар точек, находящихся в одном кластере:

$$FP = \sum_{i=1}^{r} \binom{n_i}{2} - TP = \frac{1}{2}\left(\sum_{i=1}^{r} n_i^2 - \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) \quad (17.9)$$

Наконец, количество истинно негативных срабатываний может быть получено с помощью уравнения (17.6) следующим образом:

$$TN = N - (TP + FN + FP) = \frac{1}{2}\left(n^2 - \sum_{i=1}^{r} n_i^2 - \sum_{j=1}^{k} m_j^2 + \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^2\right) \quad (17.10)$$

Коэффициент Жаккара

Коэффициент Жаккара измеряет долю истинно положительных пар точек, но после игнорирования истинных отрицательных. Это определяется следующим образом:

$$Jaccard = \frac{TP}{TP + FN + FP} \quad (17.11)$$

Для идеальной кластеризации C (т.е. полного согласия с разбиением T) коэффициент Жаккара имеет значение 1, так как в этом случае нет ложных срабатываний или ложных отрицаний. Коэффициент Жаккара асимметричен с точки зрения истинных положительных и отрицательных сторон, поскольку он игнорирует истинные отрицательные стороны. Другими словами, он подчеркивает сходство в терминах пар точек, которые принадлежат друг другу как при кластеризации, так и при разбиении по основанию, но не учитывает пары точек, которые не принадлежат друг другу.

Статистика Рэнда

Статистика Рэнда измеряет долю истинно положительных и истинно отрицательных результатов по всем парам точек; это определяется как

$$Rand = \frac{TP + FN}{N} \quad (17.12)$$

Симметричная статистика Rand измеряет долю пар точек, в которых совпадают *C* и *T*. Кластеризация префектов имеет значение 1 для статистики.

Мера Фаулкса-Мальлоу

Определяет общую *попарную точность* и значения *попарного отзыва* для кластеризации C следующим образом:

$$prec = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Точность измеряет долю истинных или правильно сгруппированных пар точек по сравнению со всеми парами точек в одном кластере. С другой стороны, отзыв измеряет долю правильно помеченных пар точек по сравнению со всеми парами точек в том же разделе.

Мера Фаулкса-Мальлоу

Мера Фаулкса-Маллльлоу (FM) определяется как среднее геометрическое для попарной точности и отзыва.

$$FM = \sqrt{prec \cdot recall} = \frac{TP}{\sqrt{(TP + FN)(TP + FP)}} \quad (17.13)$$

Мера FM также асимметрична с точки зрения истинных положительных и отрицательных сторон, поскольку игнорирует истинные отрицательные стороны. Его максимальное значение также равно 1, что достигается при отсутствии ложных срабатываний или отрицательных результатов.

Корреляционные меры

Пусть **X** и **Y** две симметричные матрицы размера $n \times n$, и пусть $N = \binom{n}{2}$. Пусть **x**, **y** $\in \mathbb{R}^N$ обозначают векторы, полученные линеаризацией верхних треугольных элементов (исключая главную диагональ) **X** и **Y** (например, построчно) соответственно. Пусть μ_X обозначает поэлементное среднее *x*, заданное как

$$\mu_X = \frac{1}{N}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \boldsymbol{X}(i,j) = \frac{1}{N}\boldsymbol{x}^T\boldsymbol{x}$$

и пусть $\mathbf{z}_x$ обозначает центрированный вектор **x**, определенный как

$$\boldsymbol{z}_x = \boldsymbol{x} - \mathbf{1}\cdot\mu_X$$

где $\mathbf{1} \in \mathbb{R}^N$ – вектор всех единиц. Аналогично, пусть μ_Y будет поэлементным средним **y**, а $\mathbf{z}_y$ - центрированным вектором **y**.

Статистика Гильберта

Статистика Гильберта определяется как усредненное поэлементное произведение между **X** и **Y**

$$\Gamma = \frac{1}{N}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \boldsymbol{X}(i,j)\cdot\boldsymbol{Y}(i,j) = \frac{1}{N}\boldsymbol{x}^T\boldsymbol{y} \quad (17.14)$$

Нормализованная статистика Гильберта определяется как поэлементная корреляция между **X** и **Y**

$$\Gamma_n = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(\boldsymbol{X}(i,j)-\mu_X)(\boldsymbol{Y}(i,j)-\mu_Y)}{\sqrt{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(\boldsymbol{X}(i,j)-\mu_X)^2 \quad \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(\boldsymbol{Y}[\boldsymbol{i}]-\mu_Y)^2}} = \frac{\sigma_{XY}}{\sqrt{\sigma_X^2\sigma_Y^2}}$$

σ_X^2 и σ_Y^2 - дисперсии, а σXY - ковариация векторов x и y, определяемых как

$$\sigma_X^2 = \frac{1}{N}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(\boldsymbol{X}(i,j)-\mu_X)^2 = \frac{1}{N}\boldsymbol{z}_x^T\boldsymbol{z}_x = \frac{1}{N}\|\boldsymbol{z}_x\|^2$$

$$\sigma_Y^2 = \frac{1}{N}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(\boldsymbol{Y}(i,j)-\mu_Y)^2 = \frac{1}{N}\boldsymbol{z}_y^T\boldsymbol{z}_y = \frac{1}{N}\|\boldsymbol{z}_y\|^2$$

$$\sigma_{XY}^2 = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(\boldsymbol{X}(i,j)-\mu_X)(\boldsymbol{Y}(i,j)-\mu_Y) = \frac{1}{N}\boldsymbol{z}_x^T\boldsymbol{z}_y$$

Таким образом, нормализованная статистика Гильберта может быть переписана как

$$\Gamma_n = \frac{\boldsymbol{z}_x^T \boldsymbol{z}_y}{\|\boldsymbol{z}_x\| \cdot \|\boldsymbol{z}_y\|} = \cos\theta \quad (17.15)$$

где θ – угол между двумя центрированными векторами $\boldsymbol{z}_x$ и $\boldsymbol{z}_y$. Отсюда сразу следует, что Γ_n изменяется от -1 до $+1$.

Когда **X** и **Y** являются произвольными матрицами размера $n \times n$, приведенные выше выражения можно легко изменить, чтобы охватить все n^2 элементов двух матриц. (Нормализованная) статистика Гильберта может использоваться в качестве меры внешней оценки с соответствующим образом определенными матрицами **X** и **Y**, как описано ниже.

Дискретизированная статистика Гильберта

Пусть **T** и **C** - матрицы размера $n \times n$, определенные как

$$\boldsymbol{T}(i,j) = \begin{cases} 1 \text{ если } y_i = y_j, i \neq j \\ 0 \text{ иначе} \end{cases}$$

$$\boldsymbol{C}(i,j) = \begin{cases} 1 \text{ если } \hat{y}_i = \hat{y}_j, i \neq j \\ 0 \text{ иначе} \end{cases}$$

Кроме того, пусть **t**, **c** $\in \mathbb{R}^N$ обозначает N-мерные векторы, составляющие верхние треугольные элементы (исключая диагональ) **T** и **C** соответственно, где $N = \binom{n}{2}$ обозначает количество различных пар точек. Наконец, пусть $\mathbf{z}_t$ и $\mathbf{z}_c$ обозначают центрированные векторы **t** и **c**.

Дискретизированная статистика Гильберта вычисляется по формуле (17.14),

положив **x** = **t** и **y** = **c**:

$$\Gamma = \frac{1}{N}\boldsymbol{t}^T\boldsymbol{c} = \frac{TP}{N} \quad (17.16)$$

Поскольку *i*-й элемент **t** равен 1 только тогда, когда *i*-я пара точек принадлежит одному и тому же разделу, и аналогично *i*-й элемент **c** равен 1 только тогда, когда *i*-я пара точек также принадлежит к тому же кластеру, точечное произведение $\boldsymbol{t}^T\boldsymbol{c}$ – это просто количество истинно положительных значений, и, таким образом, значение эквивалентно доле всех пар, которые являются истинно положительными. Отсюда следует, что чем выше соответствие между истинным разбиением *T* и кластеризацией *C*, тем выше значение.

Нормализованная дискретная статистика Гильберта

Нормализованная версия дискретизированной статистики Гильберта – это просто корреляция между **t**

и **c** $\Gamma_n = \frac{\boldsymbol{z}_t^T \boldsymbol{z}_c}{\|\boldsymbol{z}_t\| \cdot \|\boldsymbol{z}_c\|} = \cos\theta \quad (17.17)$

Обратите внимание, что $\mu_T = \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t}$ – это доля пар точек, принадлежащих одному и тому же разделу, то есть с $y_i = y_j$, независимо от того, совпадает ли $\hat{y}_i$ с $\hat{y}_j$ или нет. Таким образом, мы имеем

$$\mu_T = \frac{\boldsymbol{t}^T\boldsymbol{t}}{N} = \frac{TP + FN}{N}$$

Аналогичным образом, $\mu_C = \frac{1}{N}\boldsymbol{c}^T\boldsymbol{c}$ – это доля пар точек, принадлежащих одному кластеру, то есть с $\hat{y}_i = \hat{y}_j$, независимо от того, совпадает ли y_i и y_j или нет, так что

$$\mu_C = \frac{c^T c}{N} = \frac{TP + FP}{N}$$

Подставляя их в числитель в формуле (17.17), получаем

$$\boldsymbol{z}_t^T \boldsymbol{z}_c = (\boldsymbol{t} - \mathbf{1} \cdot \mu_T)^T (\boldsymbol{c} - \mathbf{1} \cdot \mu_C) = \boldsymbol{t}^T \boldsymbol{c} - \mu_C \boldsymbol{t}^T \mathbf{1} - \mu_T \boldsymbol{c}^T \mathbf{1} + \mathbf{1}^T \mathbf{1} \mu_T \mu_C =$$

$$= \boldsymbol{t}^T \boldsymbol{c} - N\mu_C \mu_T - N\mu_T \mu_C + N\mu_T \mu_C = \boldsymbol{t}^T \boldsymbol{c} - N\mu_T \mu_C =$$

$$= TP - N\mu_T \mu_C \quad (17.18)$$

где $\mathbf{1} \in \mathbb{R}^N$ – вектор всех единиц. Мы также использовали тождества $\boldsymbol{t}^T \mathbf{1} = \boldsymbol{t}^T \boldsymbol{t}$ и $\boldsymbol{c}^T \mathbf{1} = \boldsymbol{c}^T \boldsymbol{c}$. Аналогичным образом мы можем вывести

$$\|\boldsymbol{z}_t\| = \boldsymbol{z}_t^T \boldsymbol{z}_t = \boldsymbol{t}^T \boldsymbol{t} - N\mu_T^2 = N\mu_T - N\mu_T^2 = N\mu_T(1 - \mu_T) \quad (17.19)$$

$$\|\boldsymbol{z}_c\| = \boldsymbol{z}_c^T \boldsymbol{z}_c = \boldsymbol{c}^T \boldsymbol{c} - N\mu_C^2 = N\mu_C - N\mu_C^2 = N\mu_C(1 - \mu_C) \quad (17.20)$$

Подключая уравнения (17.18), (17.19) и (17.20) в уравнение. (17.17) нормализованная дискретизированная статистика Гильберта может быть записана как

$$\Gamma_n = \frac{\frac{TP}{N} - \mu_T \mu_C}{\sqrt{\mu_T \mu_C (1 - \mu_T)(1 - \mu_C)}} \quad (17.21)$$

поскольку $\mu_T = \frac{TP+FN}{N}$ нормализованная статистика Γ_n может быть вычислена с использованием только значений *TP*, *FN* и *FP*. Максимальное значение Γ_n = +1 получается, когда нет ложных срабатываний или отрицательных результатов, то есть, когда *FN* = *FP* = 0. Минимальное значение Γ_n = −1 - это когда нет истинных положительных и отрицательных результатов, то есть когда *TP* = *TN* = 0.

Подключая уравнения (17.18), (17.19) и (17.20) в уравнение. (17.17) нормализованная дискретизированная статистика Гильберта может быть записана как

$$\Gamma_n = \frac{\frac{TP}{N} - \mu_T \mu_C}{\sqrt{\mu_T \mu_C (1 - \mu_T)(1 - \mu_C)}} \quad (17.21)$$

поскольку $\mu_T = \frac{TP+FN}{N}$ нормализованная статистика Γ_n может быть вычислена с использованием только значений *TP*, *FN* и *FP*. Максимальное значение Γ_n = +1 получается, когда нет ложных срабатываний или отрицательных результатов, то есть, когда *FN* = *FP* = 0. Минимальное значение Γ_n = −1 - это когда нет истинных положительных и отрицательных результатов, то есть когда *TP* = *TN* = 0.

ВНУТРЕННИЕ МЕРЫ

Меры внутренней оценки не прибегают к прямому разделению, которое является типичным сценарием при кластеризации набора данных. Следовательно, для оценки качества кластеризации внутренние меры должны использовать понятия внутрикластерного сходства или компактности, в отличие от понятий межкластерного разделения, обычно с компромиссом для достижения этих двух целей. Внутренние меры основаны на *матрице расстояний* $n \times n$, также называемой *матрицей близости*, всех попарных расстояний между n точками:

$$\boldsymbol{W} = \{\delta(x_i, x_j)\}_{i,j=1}^{n} \quad (17.22)$$

где

$$\delta(x_i, x_j) = \|x_i - x_j\|_2$$

Евклидово расстояние между x_i, $x_j \in D$, хотя можно использовать и другие метрики расстояния. Поскольку **W** симметрична и $\delta(x_i, x_j) = 0$, обычно во внутренних мерах используются только верхние треугольные элементы **W** (исключая диагональ).

Матрицу близости **W** можно также рассматривать как матрицу смежности взвешенного полного графа G по n точкам, то есть с узлами $V = \{x_i \mid x_j \in D\}$, ребрами $E = \{(x_i, x_j) \mid x_i, x_j \in D\}$,, а веса ребер $w_{ij} = W(i, j)$ для всех x_i, $x_j \in D$. Таким образом, существует тесная связь между мерами внутренней оценки и задачами кластеризации графов, которые мы рассмотрели в главе 16.

Что касается внутренних мер, мы предполагаем, что у нас нет доступа к исходному разделению. Вместо этого мы предполагаем, что нам дана кластеризация $C = \{C_1, \dots, C_k)$, состоящая из $r = k$ кластеров, причем кластер C_i содержит $n_i = |C_i|$ точек. Пусть $\hat{y}_i \in \{1, 2, \dots, k\}$ обозначает метку кластера для точки x_i. Кластеризацию C можно рассматривать как k-образное разрез в G, потому что $C_i \neq 0$ для всех i, $C_i \cap C_j \neq 0$ для всех i, j и $\cup_i C_i = V$. Учитывая подмножества S, R $\subset$ V, определим W (S, R) как сумму весов на всех ребрах с одной вершиной в S и другой в R, заданных как

$$W(S, R) = \sum_{x_i \in S} \sum_{x_j \in R} w_{ij}$$

Также, если S $\subseteq$ V, обозначим через $\bar{S}$ дополнительный набор вершин, то есть $\bar{S}$ = V - S.

Внутренние меры основаны на различных функциях внутрикластерных и межкластерных весов. В частности, обратите внимание, что сумма всех внутрикластерных весов по всем кластерам задается как

$$W_{in} = \frac{1}{2}\sum_{i=1}^{k} W(C_i, C_i) \quad (17.23)$$

Мы делим на 2, потому что каждое ребро в C_i учитывается дважды при суммировании $W(C_i, C_i)$. Также обратите внимание, что сумма всех межкластерных весов задается как

$$W_{out} = \frac{1}{2}\sum_{i=1}^{k} W\left(C_i, \overline{C_i}\right) = \sum_{i=1}^{k-1}\sum_{j>1} W\left(C_i, C_j\right) \quad (17.24)$$

Здесь мы тоже делим на 2, потому что каждое ребро учитывается дважды при суммировании по кластерам.

Количество отдельных внутрикластерных ребер, обозначенных N_{in}, и межкластерных ребер, обозначенных N_{out}, задается как

$$N_{in} = \sum_{i=1}^{k} \binom{n_i}{2} = \frac{1}{2} \sum_{i=1}^{k} n_i(n_i - 1)$$

$$N_{out} = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} n_i \cdot n_j = \frac{1}{2} \sum_{i=1}^{k} \sum_{\substack{j=i+1 \\ j \neq i}}^{k} n_i \cdot n_j$$

Отметим, что общее количество различных пар точек N удовлетворяет тождеству

$$N = N_{in} + N_{out} = \binom{n}{2} = \frac{1}{2} n(n-1)$$

BetaCV Мера

Мера BetaCV — это отношение среднего внутрикластерного расстояния к среднему межкластерному расстоянию:

$$BetaCV = \frac{W_{in}/N_{in}}{W_{out}/N_{out}} = \frac{N_{out}}{N_{in}} \cdot \frac{W_{out}}{W_{in}} = \frac{N_{out}}{N_{in}} \frac{\sum_{i=1}^{k} W(C_i, C_i)}{\sum_{i=1}^{k} W\left(C_i, \overline{C_i}\right)}$$

Чем меньше отношение BetaCV, тем лучше кластеризация, поскольку это указывает на то, что внутрикластерные расстояния в среднем меньше, чем межкластерные.

C-index

Пусть $W_{min}(N_{in})$ будет суммой наименьших расстояний N_{in} в матрице близости $\mathbf{W}$, где N_{in} - общее количество внутрикластерных ребер или пар точек. Пусть $W_{max}(N_{in})$ будет суммой наибольших расстояний N_{in} в $\mathbf{W}$.

C-index измеряет, в какой степени кластеризация объединяет точки N_{in}, которые являются ближайшими в k кластерах. Он определяется как

$$Cindex = \frac{W_{in} - W_{min}(N_{in})}{W_{max}(N_{in}) - W_{min}(N_{in})}$$

где W_{in} - сумма всех внутрикластерных расстояний [ур. (17.23)]. C-index находится в диапазоне [0,1]. Чем меньше C-index, тем лучше кластеризация, поскольку он указывает на более компактные кластеры с относительно меньшими расстояниями внутри кластеров, а не между кластерами.

Нормализованная мера разреза

Нормализованная цель разреза [Ур. (16.17)] для кластеризации графов также может использоваться в качестве меры оценки внутренней кластеризации:

$$NC = \sum_{i=1}^{k} \frac{W(C_i, \bar{C}_i)}{vol(C_i)} = \sum_{i=1}^{k} \frac{W(C_i, \bar{C}_i)}{W(C_i, V)}$$

где $vol(C_i) = W(C_i, V)$ - объем кластера C_i, то есть суммарные веса на ребрах с хотя бы одним концом в кластере. Однако, поскольку мы используем матрицу близости или расстояния W вместо матрицы сходства A, чем выше нормализованное значение разреза, тем лучше.

Чтобы убедиться в этом, воспользуемся наблюдением, что $W(C_i, V) = W(C_i, C_i) + W(C_i, \overline{C_i})$, так что

$$NC = \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{W(C_i, C_i) + W(C_i, \overline{C_i})} = \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{\frac{W(C_i, C_i)}{W(C_i, \overline{C_i})} + 1}$$

Мы можем видеть, что *NC* максимизируется, когда отношения $\frac{W(C_i, C_i)}{W(C_i, \overline{C_i})}$ (по k кластерам) настолько малы, насколько это возможно, что происходит, когда внутрикластерные расстояния намного меньше по сравнению с межкластерными расстояниями, то есть когда кластеризация хорошая. Максимально возможное значение NC равно *k*.

Модульность

Модульность для кластеризации графов [Ур. (16.26)] также может использоваться как внутренняя мера:

$$Q = \sum_{i=1}^{k} \left(\frac{W(C_i, C_i)}{W(V,V)} - \left(\frac{W(C_i, V)}{W(V,V)} \right)^2 \right)$$

$$W(V,V) = \sum_{i=1}^{k} W(C_i, V) = \sum_{i=1}^{k} W(C_i, C_i) + \sum_{i=1}^{k} W\left(C_i, \bar{C}_i\right) = 2(W_{in} + W_{out})$$

Последний шаг следует из Ур. (17.23) и (17.24). Модульность измеряет разницу между наблюдаемой и ожидаемой долей весов на краях внутри кластеров. Поскольку мы используем матрицу расстояний, чем меньше показатель модульности, тем лучше кластеризация, что указывает на то, что внутрикластерные расстояния ниже ожидаемых.

Индекс Данна

Индекс Данна определяется как соотношение между минимальным расстоянием между парами точек из разных кластеров и максимальным расстоянием между парами точек из одного кластера. Более формально у нас есть

$$Dunn = \frac{W_{out}^{min}}{W_{in}^{max}}$$

Где W_{out}^{min} - минимальное межкластерное расстояние:

$$W_{out}^{min} = min_{i,j>i}\{w_{ab} | x_a \in C_i, x_b \in C_j\}$$

И W_{in}^{max} - максимальное внутрикластерное расстояние:

$$W_{in}^{max} = max_i\{w_{ab} | x_a, x_b \in C_i\}$$

Индекс Данна

Чем больше индекс Данна, тем лучше кластеризация, потому что это означает, что даже самое близкое расстояние между точками в разных кластерах намного больше, чем самое дальнее расстояние между точками в одном кластере. Однако индекс Данна может быть нечувствительным, поскольку минимальные межкластерные и максимальные внутрикластерные расстояния не охватывают всю информацию о кластеризации.

Индекс Дэвиса – Болдина

Пусть μ_i обозначает кластерное среднее, заданное как

$$\mu_i = \frac{1}{n_i} \sum_{x_j \in C_j} x_j \quad (17.25)$$

Далее, пусть σ_{μ_i} обозначает дисперсию или разброс точек вокруг среднего кластера, заданный как

$$\sigma_{\mu_i} = \sqrt{\frac{\sum_{x_j \in C_j} \delta\left(x_j, \mu_i\right)^2}{n_i}} = \sqrt{var(C_i)}$$

где $var(C_i)$ - полная дисперсия [Ур. (1.4)] кластера C_i.

Мера Дэвиса – Болдина для пары кластеров C_i и C_j определяется как отношение

$$DB_{ij} = \frac{\sigma_{\mu_i} + \sigma_{\mu_j}}{\delta(\sigma_{\mu_i}, \sigma_{\mu_j})}$$

DB_{ij} измеряет, насколько компактны кластеры по сравнению с расстоянием между средними значениями кластеров. Индекс Дэвиса – Болдина тогда определяется как

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\{DB_{ij}\}$$

То есть для каждого кластера C_i мы выбираем кластер C_j, который дает наибольшее отношение DB_{ij}. Чем меньше значение DB, тем лучше кластеризация, поскольку это означает, что кластеры хорошо разделены (т. Е. Расстояние между средними значениями кластеров большое), и каждый кластер хорошо представлен своим средним значением (т.е. имеет небольшой разброс).

Коэффициент Силуэта

Коэффициент силуэта является мерой сплоченности и разделения кластеров и основан на разнице между средним расстоянием до точек в ближайшем кластере и до точек в том же кластере. Для каждой точки x_i вычисляем коэффициент силуэта s_i как

$$s_i = \frac{\mu_{out}^{min}(x_i) - \mu_{in}(x_i)}{max\{\mu_{out}^{min}(x_i), \mu_{in}(x_i)\}} \quad (17.26)$$

где $\mu_{in}(x_i)$ - среднее расстояние от x_i до точек в собственном кластере $\hat{y}_i$:

$$\mu_{in}(x_i) = \frac{\sum_{x_j \in C_{\hat{y}_j}, j \neq i} \delta(x_i, x_j)}{n_{\hat{y}_i} - 1}$$

а $\mu_{out}^{min}(x_i)$ - среднее расстояние от xi до точек ближайшего кластера:

$$\mu_{out}^{min}(x_i) = \min_{j \neq \hat{y}_i} \left\{ \frac{\sum_{y \in C_j} \delta(x_i, y)}{n_j} \right\}$$

Значение s_i точки лежит в интервале [−1, + 1]. Значение, близкое к +1, указывает, что x_i гораздо ближе к точкам в собственном кластере и далеко от других кластеров. Значение, близкое к нулю, указывает на то, что x_i близко к границе между двумя кластерами. Наконец, значение, близкое к -1, указывает, что x_i намного ближе к другому кластеру, чем его собственный кластер, и, следовательно, точка может быть неправильно кластеризована.

Коэффициент силуэта определяется как среднее значение s_i по всем точкам:

$$SC = \frac{1}{n}\sum_{i=1}^{n} s_i \quad (17.27)$$

Значение, близкое к +1, указывает на хорошую кластеризацию.

Статистика Гильберта

Статистика Гильберта [Ур. (17.14)], и его нормализованная версия Γ_n [Eq. (17.15)], могут использоваться как меры внутренней оценки, если $\mathbf{X} = \mathbf{W}$ является парной матрицей расстояний, а также путем определения $\mathbf{Y}$ как матрицы расстояний между кластерами:

$$Y = \left\{ \delta \left(\mu_{\hat{y}_i}, \mu_{\hat{y}_j} \right) \right\}_{i,j=1}^{n} \quad (17.28)$$

Поскольку $\mathbf{W}$ и $\mathbf{Y}$ симметричны, и Γ, и Γ_n вычисляются по их верхним треугольным элементам.

Относительные меры

Относительные меры используются для сравнения различных кластеров, полученных путем варьирования разных параметров для одного и того же алгоритма, например, для выбора количества кластеров k.

Коэффициент силуэта (Silhouette Coefficient)

Коэффициент силуэта [Ур. (17.26)] для каждой точки s_j, и среднее значение SC [уравнение. (17.27)], можно использовать для оценки количества кластеров в данных. Подход состоит из построения значений s_j в порядке убывания для каждого кластера и отметки общего значения SC для конкретного значения k, а также значений SC по кластерам:

$$SC_i = \frac{1}{n_i} \sum_{x_j \in C_i} s_j$$

Затем мы можем выбрать значение k, которое дает наилучшую кластеризацию, при этом многие точки имеют высокие значения s_j в каждом кластере, а также высокие значения для SC и $SC_i (1 \leq i \leq k)$.

Индекс Калински – Харабаса

Учитывая набор данных $D = \{x_i\}_{i=1}^{n}$, матрица рассеяния для D имеет вид

$$S = n\Sigma = \sum_{j=1}^{n} (x_j - \mu)(x_j - \mu)^T$$

где $\mu = \sum_{j=1}^{n} x_j$- среднее значение, а Σ - ковариационная матрица. Матрицу разброса можно разложить на две матрицы: $S = S_W + S_B$, где S_W — это матрица внутрикластерного рассеяния, а S_B - это матрица межкластерного рассеяния, заданная как

$$S_W = \sum_{i=1}^{k} \sum_{x_j \in C_i} (x_j - \mu_i)(x_j - \mu_i)^T$$

$$S_B = \sum_{i=1}^{k} n_i (\mu_i - \mu)(\mu_i - \mu)^T$$

$\mu_i = \frac{1}{n_i} \sum_{x_j \in C_i} x_j$ - среднее значение для кластера C_i

Критерий отношения дисперсии Калински – Харабаса (CH) для данного значения k определяется следующим образом

$$CH(k) = \frac{tr(S_B)/(k-1)}{tr(S_W)/(n-k)} = \frac{n-k}{k-1} \cdot \frac{tr(S_B)}{tr(S_W)}$$

где $tr(S_W)$ и $tr(S_B)$ — это следы (сумма диагональных элементов) внутрикластерных и межкластерных матриц рассеяния.

Для хорошего значения k мы ожидаем, что разброс внутри кластера будет меньше по сравнению с разбросом между кластерами, что должно привести к более высокому значению $CH(k)$. С другой стороны, нам не нужно очень большое значение k; таким образом, член $\frac{n-k}{k-1}$ ставить в невыгодное положение большие значения k. Мы могли бы выбрать значение k, которое максимизирует $CH(k)$. В качестве альтернативы, мы можем построить график значений CH и ожидать большого увеличения значения с последующим небольшим усилением или его отсутствием. Например, мы можем выбрать значение $k > 3$, которое минимизирует член

$$\Delta(k) = \big(CH(k+1) - CH(k)\big) - \big(CH(k) - CH(k-1)\big)$$

Интуиция заключается в том, что мы хотим найти значение k, для которого $CH(k)$ намного выше, чем $CH(k-1)$, и есть только небольшое улучшение или уменьшение значения $CH(k+1)$.

Статистика разрыва

Статистика разрыва сравнивает сумму внутрикластерных весов W_{in} [Ур. (17.23)] для различных значений k с их ожидаемыми значениями, предполагающими отсутствие явной структуры кластеризации, что формирует нулевую гипотезу.

Пусть C_k будет кластеризацией, полученной для указанного значения k с использованием выбранного алгоритма кластеризации. Пусть $W_{in}^k(D)$ обозначает сумму внутрикластерных весов (по всем кластерам) для C_k во входном наборе данных D. Мы хотим вычислить вероятность наблюдаемого значения W_k при нулевой гипотезе о том, что точки случайным образом размещаются в то же пространстве данных, что и D. К сожалению, распределение выборки W_{in} неизвестно. Кроме того, оно зависит от количества кластеров k, количества точек n и других характеристик D.

Чтобы получить эмпирическое распределение для W_{in} , мы прибегаем к моделированию процесса выборки методом Монте-Карло. То есть мы генерируем t случайных выборок, содержащих n случайно распределенных точек в том же d-мерном пространстве данных, что и входной набор данных D.

То есть для каждого измерения D , скажем X_j , мы вычисляем его диапазон $[min(X_j), max(X_j)]$ и генерируем значения для n точек (для j-го измерения) равномерно случайным образом в заданном диапазоне. Обозначим через $R_i \in \mathbb{R}^{n\ x\ d}, 1 \leq\ i \leq\ t,\ i$ -й образец.

Пусть $W_{in}^{k}(R_i)$ обозначает сумму внутрикластерных весов для данной кластеризации R_i в k кластеров. Из каждого образца набора данных R_i мы генерируем кластеризацию для различных значений k, используя тот же алгоритм, и записываем внутрикластерные значения $W_{in}^{k}(R_i)$. Пусть $\mu_W(k)$ и $\mu_W(k)$ обозначают среднее и стандартное отклонение этих внутрикластерных весов для каждого значения k, заданного как

$$\mu_W(k) = \frac{1}{t}\sum_{l=1}^{t} \log W_{in}^{k}(R_i)$$

$$\sigma_W(k) = \sqrt{\frac{1}{t}\sum_{i=1}^{t}\left(\log W_{in}^{k}(R_i) - \mu_W(k)\right)^2}$$

где мы используем логарифм значений W_{in}, так как они могут быть довольно большими.

Статистика разрыва для данного k тогда определяется как

$$gap(k) = \mu_W(k) - \log W_{in}^k\,(D)$$

Он измеряет отклонение наблюдаемого значения W_{in}^k от его ожидаемого значения при нулевой гипотезе. Мы можем выбрать значение k, которое дает статистику наибольшего разрыва, поскольку оно указывает на структуру кластеризации вдали от равномерного распределения точек. Более надежный подход - выбрать k следующим образом:

$$k^* = \arg\min_k\{gap(k) \geq gap(k+1) - \sigma_W(k+1)\}$$

То есть мы выбираем наименьшее значение k так, чтобы статистика разрыва находилась в пределах одного стандартного отклонения разрыва при $k + 1$.

Стабильность кластера

Основная идея стабильности кластера заключается в том, что кластеризация, полученная из нескольких наборов данных, отобранных из того же базового распределения, что и D, должна быть аналогичной или «стабильной». Подход кластерной стабильности можно использовать для поиска хороших значений параметров для данного алгоритма кластеризации; мы сосредоточимся на задаче поиска хорошего значения k, правильного количества кластеров.

Пусть $\mathcal{C}_{\mathrm{k}}(\mathrm{D_i})$ обозначает кластеризацию, полученную из выборки $\mathrm{D_i}$, для данного значения k. Затем метод сравнивает расстояние между всеми парами кластеризации $\mathcal{C}_{\mathrm{k}}(\mathrm{D_i})$ и $\mathcal{C}_{\mathrm{k}}(\mathrm{D_j})$ с помощью некоторой функции расстояния. Некоторые из показателей внешней кластерной оценки можно использовать в качестве меры расстояния, задав, например, $\mathcal{C} = \mathcal{C}_{\mathrm{k}}(\mathrm{D_i})$ и $T = \mathcal{C}_{\mathrm{k}}(D_j)$ или наоборот.

По этим значениям мы вычисляем ожидаемое попарное расстояние для каждого значения k. Наконец, значение k^*, которое показывает наименьшее отклонение между кластеризацией, полученной из повторно дискретизированных наборов данных, является лучшим выбором для k, поскольку оно демонстрирует наибольшую стабильность.

Зададим

$$D_{ij} = D_i \cap D_j = \left\{ m^a \text{ экземпляр } x_a \;\middle|\; x_a \in D, m^a = \min\left\{ m_i^a, m_j^a \right\} \right\} \qquad \textbf{(17.30)}$$

То есть общий набор данных D_{ij} создается путем выбора минимального количества экземпляров точки x_a в $\mathrm{D_i}$ или $\mathrm{D_j}$.

Алгоритм кластеризации устойчивости для выбора k

Стабильность кластера (A, t, k^{max}, D):

$1\ n \leftarrow |D|$

// Генерация t образцов

2 for $i = 1,2, \ldots, t$ **do**

3 $\mathrm{D_i} \leftarrow$ образец n точек из D с заменой

// Генерация кластеризации для разных значений k

4 for $i = 1,2, \ldots, t$ **do**

5 **for** $k = 2,3, \ldots, k^{\max}$ **do**

6 $\mathcal{C}_{\mathrm{k}}(\mathrm{D_i}) \leftarrow$ кластер Di в k в k кластеров с использованием алгоритма A

// Вычислить среднюю разницу между кластеризацией для каждого k

7 foreach *pair* $\mathrm{D_i}, \mathrm{D_j}$ *with* $\mathrm{j} > \mathrm{i}$ **do**

8 $\mathrm{D_{ij}} \leftarrow \mathrm{D_i} \cap \mathrm{D_j}$ //создать общий набор данных используя Ур. (17.30)

9 **for** $\mathrm{k} = 2,3, \ldots, k^{\max}$ **do**

10 $d_{ij}(k) \leftarrow d\left(\mathcal{C}_{\mathrm{k}}(D_{\mathrm{i}}), \mathcal{C}_{\mathrm{k}}\left(D_{\mathrm{j}}\right), D_{ij}\right)$// расстояние между кластерами

11 for $k = 2,3, \ldots, k^{\max}$ **do**

12 $\mu_{\mathrm{d}}(\mathrm{k}) \leftarrow \frac{2}{\mathrm{t}(\mathrm{t}-1)} \sum_{\mathrm{i}=1}^{\mathrm{t}} \sum_{\mathrm{j}>\mathrm{i}} \mathrm{d_{ij}}(\mathrm{k})$ // ожидаемое попарное расстояние

// Выбор лучшего k

13 $k^* \leftarrow \arg\min_k\{\mu_d(k)\}$

Алгоритм 17.1 показывает псевдокод метода стабильности кластеризации для выбора наилучшего значения k. Он принимает в качестве входных данных алгоритм кластеризации A, количество выборок t, максимальное количество кластеров k^{max} и входной набор данных D. Сначала он генерирует t выборок начальной загрузки и кластеризует их, используя алгоритм A. Затем он вычисляет расстояние между кластерами для каждой пары наборов данных Di и Dj для каждого значения k.

Наконец, метод вычисляет ожидаемое попарное расстояние $\mu_d(k)$ в строке 12. Мы предполагаем, что функция расстояния кластеризации d является симметричной. Если d не является симметричным, то ожидаемую разницу следует вычислять по всем упорядоченным парам, то есть $\mu_d(k) = \frac{1}{r(t-1)} \sum_{i=1}^{r} \sum_{j \neq i} d_{ij}(k)$

Тенденция к кластеризации

Тенденция к кластеризации или кластеризация направлена на определение того, есть ли в наборе данных D какие-либо значимые группы для начала. Обычно это сложная задача, учитывая разные определения того, что значит быть кластером, например, секционированный, иерархический, на основе плотности, на основе графа и так далее.

Даже если мы исправим тип кластера, определение соответствующей нулевой модели (например, модели без какой-либо структуры кластеризации) для данного набора данных все равно будет сложной задачей.

Кроме того, если мы определим, что данные являются кластеризованными, тогда мы все еще сталкиваются с вопросом, сколько существует кластеров. Тем не менее, все же стоит оценить возможность кластеризации набора данных; мы рассмотрим некоторые подходы, чтобы ответить на вопрос, являются ли данные кластеризованными или нет.

Пространственная гистограмма

Один из простых подходов - сопоставить d-мерную пространственную гистограмму входного набора данных D с гистограммой из выборок, случайно сгенерированных в том же пространстве данных. Пусть $X_1, X_2, \ldots X_d$ обозначают d измерений. Учитывая b, количество ячеек для каждого измерения, мы делим каждое измерение X_j на b ячеек равной ширины и просто подсчитываем, сколько точек лежит в каждой из ячеек d-размерности b^d. Из этой пространственной гистограммы мы можем получить эмпирическую функцию совместной вероятности и массы (EPMF - empirical joint probability mass function) для набора данных D, которая является приближением неизвестной совместной функции плотности вероятности. EPMF задается как

$$f(i) = P\left(x_j \in \text{ cell i } i\right) = \frac{\left|\left\{x_j \in cell\ i\right\}\right|}{n}$$

где $\mathrm{i} = (i_1, i_2, \ldots, i_d)$ обозначает индекс ячейки, а i_j обозначает индекс ячейки по измерению X_j.

Затем мы генерируем t случайных выборок, каждая из которых содержит n точек в том же d-мерном пространстве, что и входной набор данных D. То есть для каждого измерения X_j мы вычисляем его диапазон $[min(X_j), max(X_j)]$ и генерируем значения равномерно случайным образом в пределах данного диапазона. Обозначим через R_j j-ю такую случайную выборку. Затем мы можем вычислить соответствующий EPMF $g_j(i)$ для каждого R_j $1 \leq j \leq t$.

Наконец, мы можем вычислить, насколько распределение f отличается от g_j (для $j = 1, \ldots, t$), используя дивергенцию Кульбака – Лейблера (KL) от f к g_j, определяемую как

$$KL(f \mid g_j) = \sum_i f(i) \log\left(\frac{f(i)}{g_j(i)}\right) \qquad \textbf{17.31}$$

Дивергенция KL равна нулю только тогда, когда f и g_j - одинаковые распределения. Используя эти значения расхождения, мы можем вычислить, насколько набор данных D отличается от случайного набора данных.

Основное ограничение этого подхода заключается в том, что по мере увеличения размерности количество ячеек (b^d) увеличивается экспоненциально, а при фиксированном размере выборки n большинство ячеек будет пустым или будет иметь только одну точку, что затрудняет оценку расхождение. Метод также чувствителен к выбору параметра b. Вместо гистограмм и соответствующих EPMF мы также можем использовать методы оценки плотности (см. Раздел 15.2), чтобы определить совместную функцию плотности вероятности (PDF) для набора данных D и посмотреть, чем она отличается от PDF для случайных наборов данных. Однако проклятие размерности также создает проблемы для оценки плотности.

Распределение расстояния

Вместо того, чтобы пытаться оценить плотность, другой подход к определению кластеризации состоит в сравнении попарных расстояний между точками от D с расстояниями от случайно сгенерированных выборок R_j из нулевого распределения. То есть мы создаем EPMF из матрицы близости W для D [Ур. (17.22)] путем объединения расстояний в интервалы b:

$$f(i) = P\big(\, w_{pq} \in bin\ i \mid x_p, x_q \in D, p < q \,\big) = \frac{\big|\{w_{pq} \in bin\ i\}\big|}{n(n-1)/2}$$

Точно так же для каждого из образцов R_j мы можем определить EPMF для попарных расстояний, обозначенных g_j. Наконец, мы можем вычислить KL-расходимости между f и g_j, используя ур. (17.31). Ожидаемое расхождение указывает на степень, в которой D отличается от нулевого (случайного) распределения.

Статистика Хопкинса

Статистика Хопкинса — это тест с разреженной выборкой на пространственную случайность. Учитывая набор данных D, содержащий n точек, мы генерируем t случайных подвыборок R_i по m точек в каждой, где $m \ll n$. Эти образцы берутся из того же пространства данных, что и D, и генерируются равномерно случайным образом по каждому измерению. Кроме того, мы также генерируем t подвыборок из m точек непосредственно из D, используя выборку без замены. Обозначим через D_i $i-$ю прямую подвыборку. Затем мы вычисляем минимальное расстояние между каждой точкой $x_j \in D_i$ и точками в D

$$\delta_{min}(x_j) = \min_{x_i \in D, x_i \neq x_j} \{\delta(x_j, x_i)\}$$

Точно так же мы вычисляем минимальное расстояние $\delta_{min}(y_j)$ между точкой $y_j \in R_i$ и точками в D.

Статистика Хопкинса (в d-измерениях) для i -й пары выборок R_i и D_i затем определяется как

$$HS_i = \frac{\sum_{y_j \in R_i} \left(\delta_{min}(y_j)\right)^d}{\sum_{y_j \in R_i} \left(\delta\, min(y_j)\right)^d + \sum_{x_j \in D_i} \left(\delta_{min}(x_j)\right)^d}$$

Эта статистика сравнивает распределение ближайших соседей случайно сгенерированных точек с таким же распределением для случайных подмножеств точек из D. Если данные хорошо сгруппированы, мы ожидаем, что значения $\delta_{min}(x_j)$ будут меньше по сравнению со значениями $\delta_{min}(y_j)$ и в этом случае HSi стремится к 1. Если расстояние до обоих ближайших соседей одинаково, тогда HS_i принимает значения, близкие к 0.5, что указывает на то, что данные, по существу, случайны и явной кластеризации нет. Наконец, если значения $\delta_{min}(x_j)$ больше по сравнению со значениями $\delta_{min}(y_j)$, то HS_i стремится к 0 и указывает на точечное отталкивание без кластеризации. Затем из t различных значений HS_i мы можем вычислить среднее значение и дисперсию статистики, чтобы определить, является ли D кластеризируемым или нет.