

Урок 2

Подробнее о методах кластеризации

2.1. Метод К средних (K-Means)

2.1.1. Как работает K-Means

Алгоритм K-means представляет собой итеративный процесс. Пусть в начале произвольным образом выбраны центры классов. Объект относится к тому кластеру, расстояние до центра которого меньше. На каждой итерации сначала пересчитываются положения центра каждого класса как среднее арифметическое отнесенных к нему точек, а после этого объекты перераспределяются на основании новых положений центров.

2.1.2. Варианты метода K-Means

Описанный выше вариант метода K-means называется K-Means'ом Болла Холла. Существует также версия Мак Кина, в которой при каждом переходе объекта из кластера в кластер центры кластеров пересчитываются.

В ситуации, когда количество объектов очень велико, чтобы не вычислять среднее арифметическое по всем объектам, можно на каждой итерации считать положения центров классов по некоторой случайной выборке объектов. Такой вариант метода называется Mini Batch K-Means и позволяет использовать метод K-Means на огромных выборках.

Если размерность пространства признаков d велика, то операция вычисление расстояния между объектами будет вычислительно сложной (требовать порядка d элементарных операций). Решить проблему можно уменьшением числа признаков: использовать отбор признаков, метод главных компонент (PCA) или сингулярное разложение (SVD).

Сходимость метода K-means сильно зависит от выбора начальных значений центров. В случае двух кластеров можно в качестве центров выбрать две самые удаленные друг от друга точки. Для произвольного числа кластеров существует алгоритм k-means++. Это вариация метода K-means, в которой начальные приближения выбираются не случайно, а некоторым более оптимальным образом. Положение первого центра класса выбирается случайно из равномерного распределения на выборке. На каждом следующем шаге сначала вводится такое вероятностное распределение на выборке, что вероятность выбора точки пропорциональна квадрату расстояния до ближайшего до нее центра, а положение нового центра выбирается из этого распределения.

2.1.3. Пример: квантизация изображений

Допустим, необходимо понизить количество цветов в изображении. Изначально изображение содержит около 96 тысяч цветов из пространства цветов RGB.

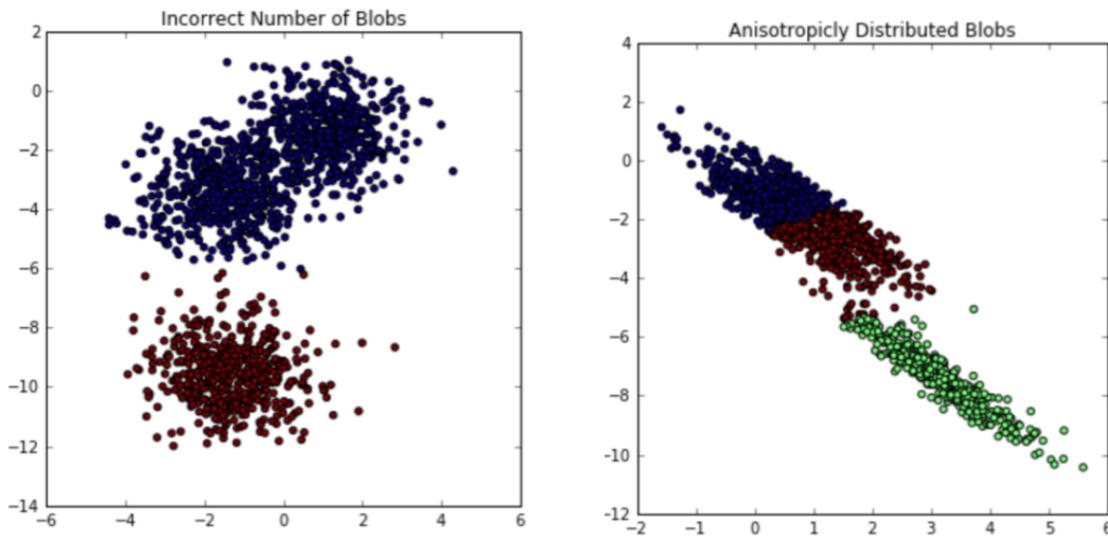
Можно уменьшить количество цветов, выбрав 64 случайных цвета, а остальные цвета на изображении заменить на ближайший в пространстве RGB цвет из них. С другой стороны, точки изображения можно рассматривать как объекты в пространстве признаков и кластеризовать их, например, с помощью K Means. Такой способ дает очень хороший результат.



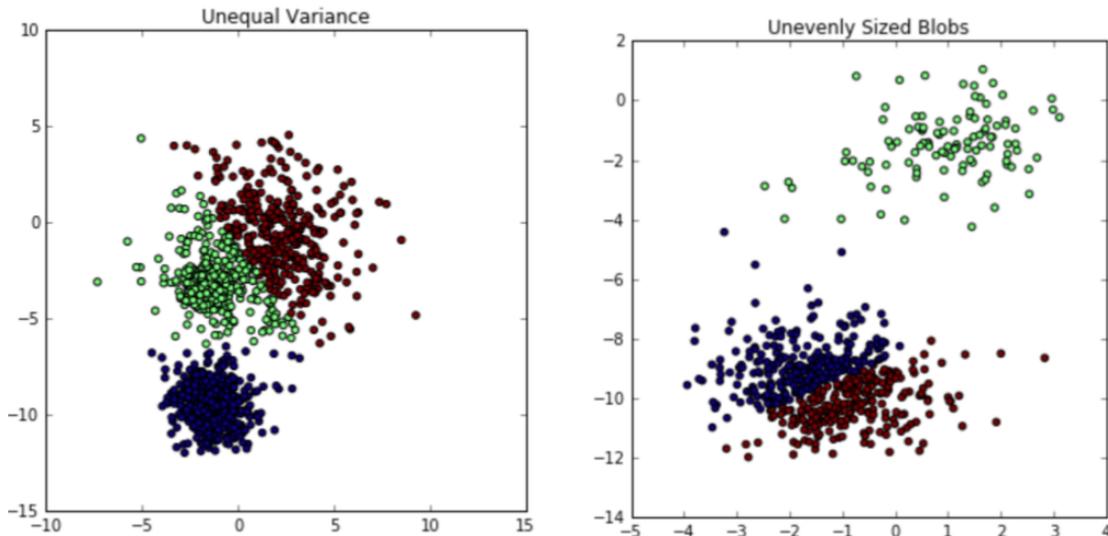
Кластеризация, таким образом, представляет собой естественный способ огрубления признаков и позволяет свести возможные варианты векторов признаков к какому то небольшому набору.

2.1.4. K-means и форма кластеров

Метод K-means может совершенно по-разному работать в зависимости от формы кластеров и от выбранного количества кластеров. Например, алгоритм может решить, что два вытянутых кластера — это одно скопление.

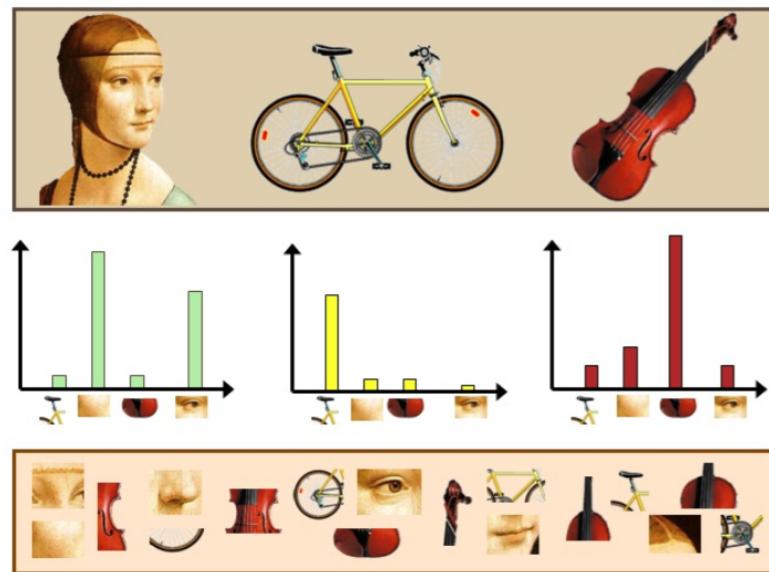


Также, в случае неодинакового разброса у кластеров, алгоритм может не очень оптимально поделить два рядом находящихся друг с другом кластера.

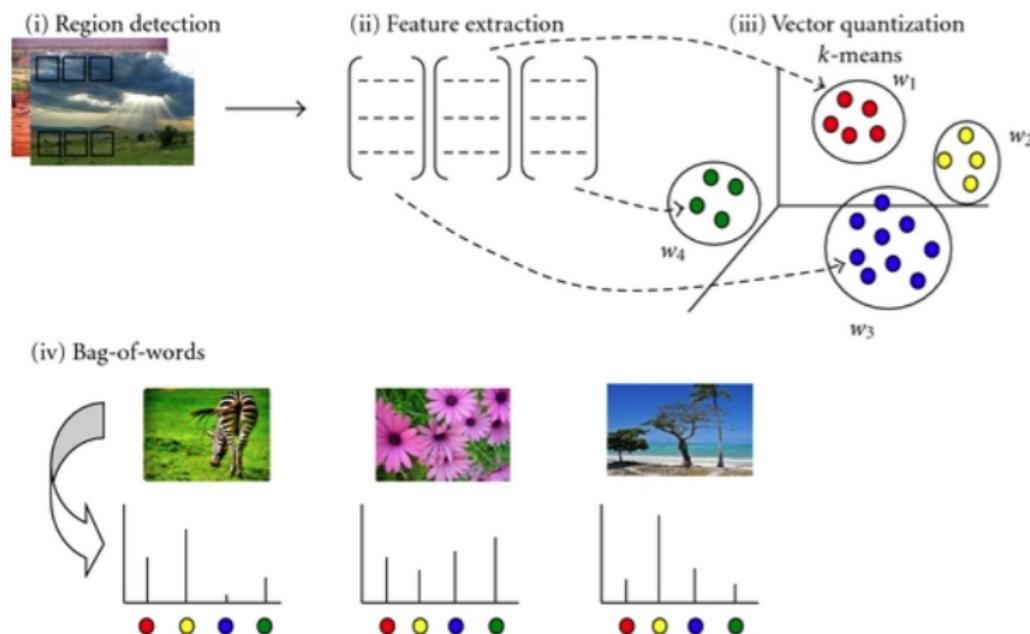


2.1.5. Пример: мешок визуальных слов

Другой пример использования K-Means — это мешок визуальных слов. До того как в анализе изображений начали активно использовать свёрточные нейросети, применяли следующий подход для классификации: нарезали изображения на множество кусочков и по аналогии с тем, как классифицируются тексты по частотам разных слов в тексте, в качестве признаков изображения выбирали частоты тех или иных фрагментов изображения.



Проблема заключалась в том, что точное совпадение фрагментов двух изображений практически невозможно. Поэтому фрагменты изображений помещались в признаковое пространство и кластеризовались с помощью K-means. Каждый из получившихся кластеров интерпретировался как «визуальное слово».



2.1.6. Функционал, который минимизирует K-means

Среднее внутрекластерное расстояние имеет следующий вид:

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

Если известны положения центров кластеров, можно записать аналогичную метрику, которая представляет собой среднее расстояние до центра кластера:

$$\Phi_0 = \sum_{y \in Y} \frac{1}{|K_y|} \sum_{i: y_i = y} \rho^2(x_i, \mu_y) \rightarrow \min.$$

Оказывается, что K-Means в варианте Мак Кина находит локальный минимум функционала Φ_0 . Это интуитивно понятно из того факта, что в K-Means итеративно минимизируется средние внутрикластерные расстояния: объект присваивается к тому кластеру, центр которого ближе, а центр кластера перемещается в среднее арифметическое векторов признаков объектов. Легко убедиться, что в одномерном пространстве выбор точки μ минимизирует среднеквадратичное отклонение от других точек кластера:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i = \operatorname{argmin}_{\mu} \frac{1}{N} \sum_{i=1}^N (\mu - x_i)^2.$$

Действительно:

$$\frac{d}{d\mu} \frac{1}{N} \sum_{i=1}^N (\mu - x_i)^2 = \frac{2}{N} \sum_{i=1}^N (\mu - x_i) = 0 \implies \mu = \frac{1}{N} \sum_{i=1}^N x_i.$$

2.2. Expectation Maximization (EM-алгоритм)

2.2.1. Постановка задачи

EM-алгоритм, как и k-means, является итеративным методом: сначала задается начальное приближение, а затем на каждом шаге форма кластеров уточняется.

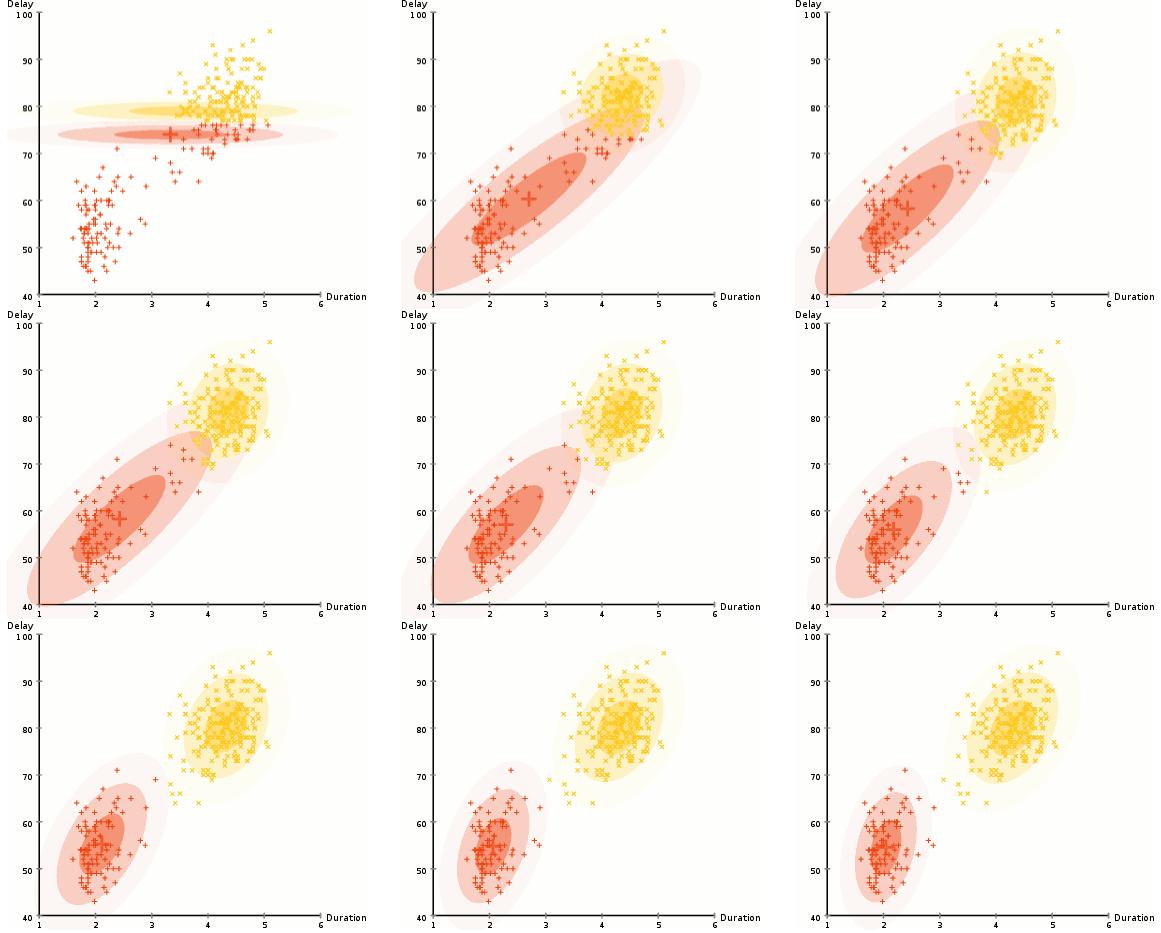


Рис. 2.1: Пример работы EM-алгоритма

2.2.2. Постановка задачи разделения смеси распределений

Пусть w_1, \dots, w_K — априорные вероятности кластеров, $p_1(x), \dots, p_K(x)$ — плотности распределения кластеров, тогда плотность распределения вектора признаков x сразу по всем кластерам равна:

$$p(x) = \sum_{j=1}^K w_j p_j(x).$$

Необходимо на основе выборки оценить параметры модели w_1, \dots, w_K , $p_1(x), \dots, p_K(x)$. Это позволит оценивать вероятность принадлежности к кластеру и, таким образом, решить задачу кластеризации. Такая задача называется задачей разделения смеси распределений:

$$p(x) = \sum_{j=1}^K w_j p_j(x), \quad p_j(x) = \varphi(\theta_j; x),$$

где θ_j — параметры распределения $p_j(x)$.

Обычно все распределения отдельных компонент берутся из одного семейства. Например, все компоненты могут иметь нормальное распределение.

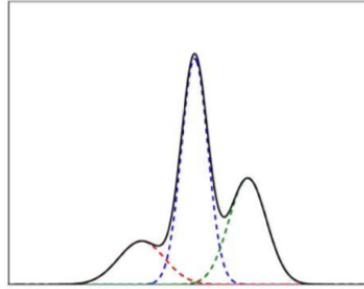


Рис. 2.2: Смесь трех нормальных распределений.

Согласно принципу максимизации правдоподобия:

$$\theta, w = \underset{\theta, w}{\operatorname{argmax}} \sum_{i=1}^K \ln p(x_i) = \underset{\theta, w}{\operatorname{argmax}} \sum_{i=1}^K \ln \left(\sum_{j=1}^K w_j p_j(x_i) \right).$$

Таким образом, имеет место задача максимизации суммы логарифмов сумм, решение которой представляет большую трудность. В таком случае полезным оказывается итеративный метод решения — EM-алгоритм.

2.2.3. EM-алгоритм

EM-алгоритм заключается в последовательном выполнении двух шагов:

- **E-шаг:** Вычисляются вспомогательные переменные:

$$g_{ji} = p(\theta_j | x_i) = \frac{w_j p_j(x_i)}{p(x_i)}.$$

- **M-шаг:** При зафиксированных g_{ji} решение задачи максимизации правдоподобия может быть найдено согласно:

$$w_j = \frac{1}{N} \sum_{i=1}^N g_{ji}, \quad \theta_j = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^N g_{ji} \ln \varphi(\theta; x).$$

Итерации происходят до сходимости.

2.2.4. Пример: 2 кластера с гауссовской плотностью

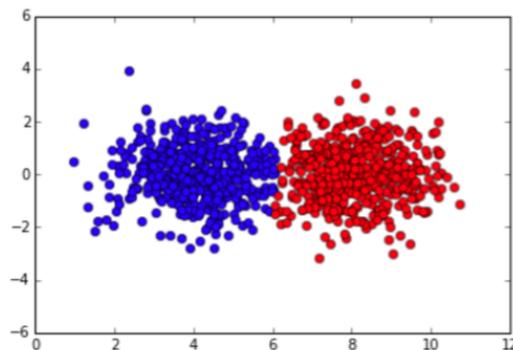


Рис. 2.3: Два получившихся сгустка можно интерпретировать как кластеры

В случае смеси гауссовых распределений:

$$p(x) = w_1 p_1(x) + w_2 p_2(x),$$

можно выписать более подробные выражения для М-шага:

- **E-шаг:**

$$g_{ji} = p(j|x_i) = \frac{w_j p_j(x_i)}{p(x_i)}$$

- **M-шаг:**

$$w_j = \frac{1}{N} \sum_{i=1}^N g_{ji}, \quad \mu_j = \frac{1}{N w_j} \sum_{i=1}^N g_{ji} x_i, \quad \Sigma_j = \frac{1}{N w_j - 1} \sum_{i=1}^N g_{ji} (x_i - \mu_j)(x_i - \mu_j)^T$$

Объект относится к кластеру j , для которого максимально значение $p(j|x_i) = g_{ij}$.

2.2.5. Простое и классическое объяснения ЕМ-алгоритма

Простое объяснение ЕМ-алгоритма заключается в том, что для построения итерационного процесса нужно ввести скрытые переменные таким образом, чтобы вычисления были как можно более простые. Оказывается, что в качестве таких скрытых переменных удобно рассмотреть (здесь сразу применена формула Байеса):

$$g_{ji} = P(j|x_i) = \frac{w_j p_j(x_i)}{\sum_{k=1}^K w_k p_k(x_i)}.$$

На втором шаге (М-шаге) решается задача максимизации правдоподобия с зафиксированными $P(j|x_i)$. Если приравнять производные по параметрам к нулю, получаются выражения:

$$w_J = \frac{1}{N} \sum_{i=1}^N g_{ji}, \quad \theta_j = \operatorname{argmax}_{\theta} \sum_{i=1}^N g_{ji} \ln \phi(\theta; x).$$

Классическое объяснение ЕМ-алгоритма (которое объясняет его название Expectation-Maximization) заключается в следующем. Пусть $L(\theta; X, Z) = p(X, Z|\theta)$ — функция правдоподобия. Тогда в ЕМ-алгоритме на Е-шаге происходит построение функции ожидаемого значения логарифма правдоподобия, зависящей от параметров θ :

$$Q(\theta|\theta^{(t)}) = E_{Z|X,\theta^{(t)}} \log L(\theta; X, Z),$$

а на М-шаге вычисляются значения параметров $\theta^{(t+1)}$, которые максимизируют $Q(\theta|\theta^{(t)})$:

$$\theta^{(t+1)} = \operatorname{argmax}_{\theta} Q(\theta|\theta^{(t)}).$$

2.2.6. Приложения ЕМ-алгоритма

ЕМ-алгоритм имеет следующие приложения:

- Оценка параметров в других вероятностных моделях (не только в смеси распределений).
- Восстановление плотности распределения.
- Классификация.

2.3. Агломеративная иерархическая кластеризация

2.3.1. Иерархическая кластеризация

Иерархическая кластеризация — кластеризация, в которой кластеры получаются вложенными друг в друга. Выделяют два способа это сделать:

- Агломеративный подход: каждый объект помещается в свой собственный кластер, которые постепенно объединяются.
- Дивизивный (англ. divisive) подход: сначала все объекты помещаются в один кластер, который затем разбивается на более мелкие кластеры.

Более распространен агломеративный подход, поэтому, когда говорят «иерархическая кластеризация», часто имеют в виду именно его. Что именно имеется в виду, к сожалению, приходится понимать из контекста.

2.3.2. Агломеративная кластеризация

На данном примере показан ход выполнения агломеративной иерархической кластеризации. Считается, что выбран некоторый способ вычисления расстояния между кластерами.

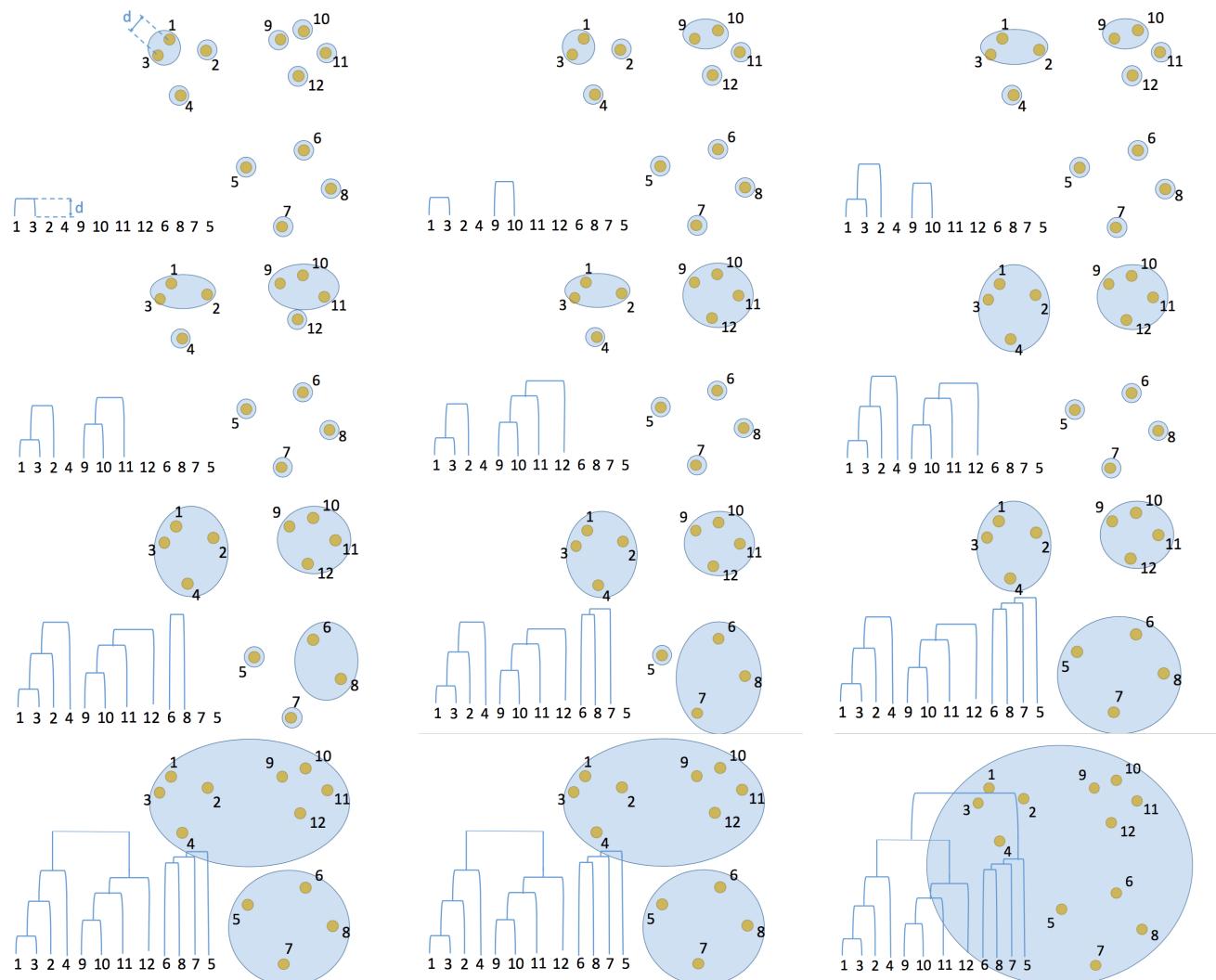


Рис. 2.4: Процесс агломеративной иерархической кластеризации (сразу представлено построение так называемой дендрограммы, о которой пойдет речь несколько позже)

Главная особенность метода агломеративной иерархической кластеризации состоит в том, что для изменения желаемого числа кластеров запускать заново алгоритм не нужно. Для этого достаточно рассмотреть дерево объединения кластеров и обрезать его на желаемом шаге.

2.3.3. Расстояние между кластерами

Расстояние между кластерами можно ввести несколькими разными способами.

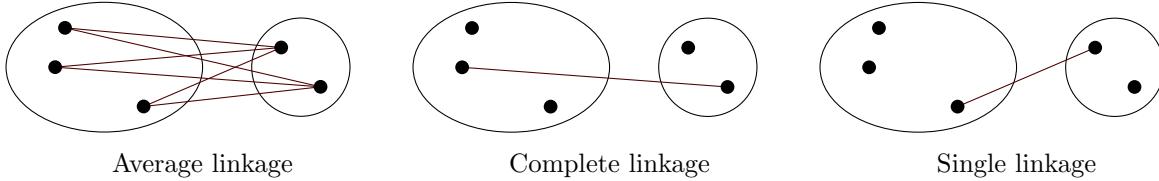


Рис. 2.5: Различные способы ввести расстояние между кластерами

С помощью формулы Ланса-Уильямса можно обобщить множество разных способов ввести расстояния между кластерами:

$$R(U \cup V, S) = \alpha_U R(U, S) + \alpha_V R(V, S) + \beta R(U, V) + \gamma |R(U, S) - R(V, S)|.$$

Эта формула выражает расстояние между кластером, которым получается в результате слияния двух кластеров U и V , и каким-то третьим кластером S . Формула позволяет рекурсивно получить расстояние между двумя сложными кластерами, если известно расстояние между простыми. Разные коэффициенты в этой формуле приводят к разным способам вычислять расстояние между кластерами, в том числе:

- Расстояние ближайшего соседа (при $\alpha_U = \alpha_V = \frac{1}{2}$, $\beta = 0$, $\gamma = -\frac{1}{2}$):

$$R^B(W, S) = \min_{w \in W, s \in S} \rho(w, s).$$

- Расстояние дальнего соседа (при $\alpha_U = \alpha_V = \frac{1}{2}$, $\beta = 0$, $\gamma = \frac{1}{2}$):

$$R^D(W, S) = \max_{w \in W, s \in S} \rho(w, s).$$

- Среднее расстояние (при $\alpha_U = \frac{|U|}{|W|}$, $\alpha_V = \frac{|V|}{|W|}$, $\beta = \gamma = 0$):

$$R^C(W, S) = \frac{1}{|W||S|} \sum_{w \in W} \sum_{s \in S} \rho(w, s).$$

- Расстояние между центрами кластеров (при $\alpha_U = \frac{|U|}{|W|}$, $\alpha_V = \frac{|V|}{|W|}$, $\beta = -\alpha_U \alpha_V$, $\gamma = 0$):

$$R^{II}(W, S) = \rho^2 \left(\sum_{w \in W} \frac{w}{|W|}, \sum_{s \in S} \frac{s}{|S|} \right).$$

- Расстояние Уорда (при $\alpha_U = \frac{|S|+|U|}{|S|+|W|}$, $\alpha_V = \frac{|S|+|V|}{|S|+|W|}$, $\beta = \frac{-|S|}{|S|+|W|}$, $\gamma = 0$):

$$R^{II}(W, S) = \frac{|S||W|}{|S| + |W|} \rho^2 \left(\sum_{w \in W} \frac{w}{|W|}, \sum_{s \in S} \frac{s}{|S|} \right).$$

2.3.4. Дендрограмма

Построение дендрограммы — очень удобный способ визуализировать иерархическую кластеризацию. Расстояние между кластерами на дендрограмме изображаются как высота дуги, которой соединяются метки кластеров (см. на рисунке выше). Также наглядно будет построить график зависимости расстояния между сливаемыми кластерами от номера итерации.

2.3.5. Пример: дендрограмма в задаче кластеризации писем по теме

Далее представлена дендрограмма построенная на реальных данных в задаче кластеризации писем по теме.

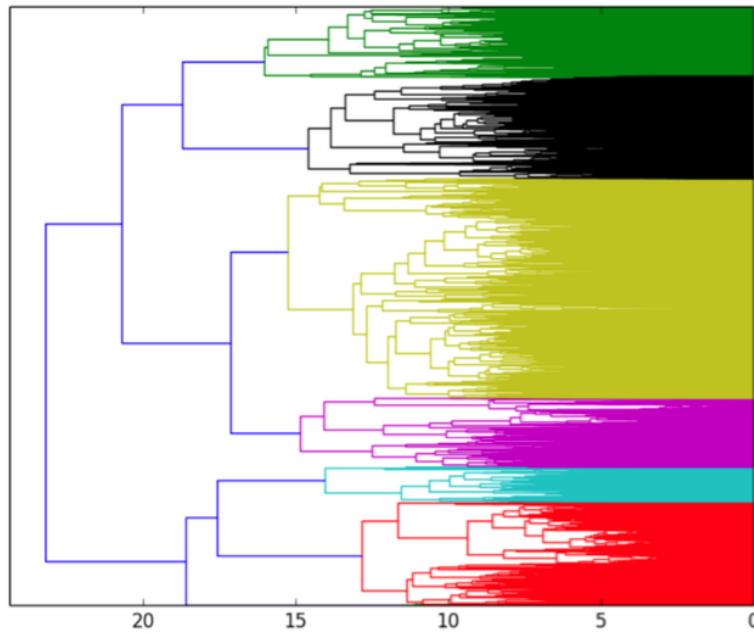


Рис. 2.6: Дендрограмма в задаче кластеризации писем по теме

Если построить дендрограмму для небольшой выборки (100 писем), то она будет иметь вид:

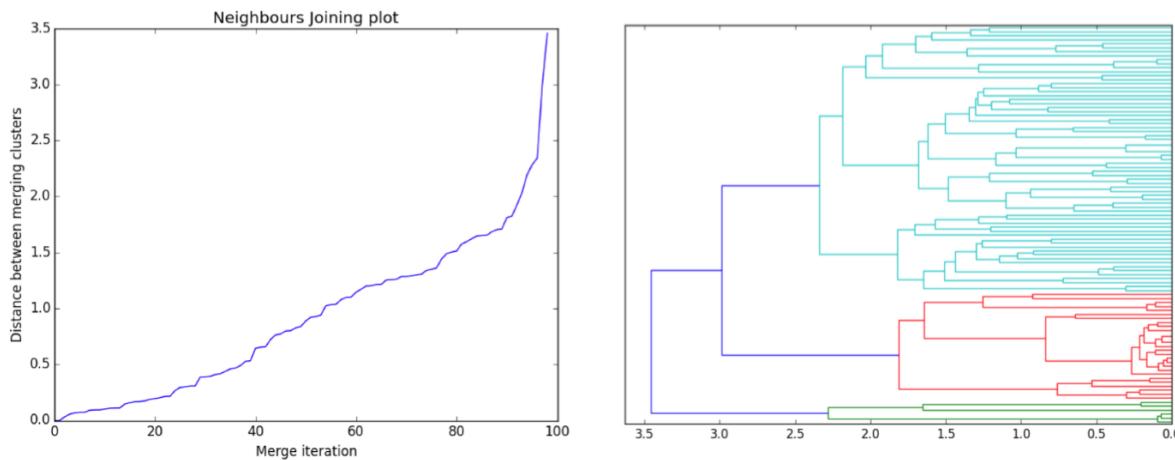


Рис. 2.7: Дендрограмма в задаче кластеризации писем по теме (на подвыборке из 100 писем) и график зависимости расстояния между сливаемыми кластерами от номера итерации

На представленном графике ближе к последним итерациям расстояние между кластерами быстро взмывает вверх, то есть существует некоторое разумное количество кластеров, которые друг от друга более-менее удалены.

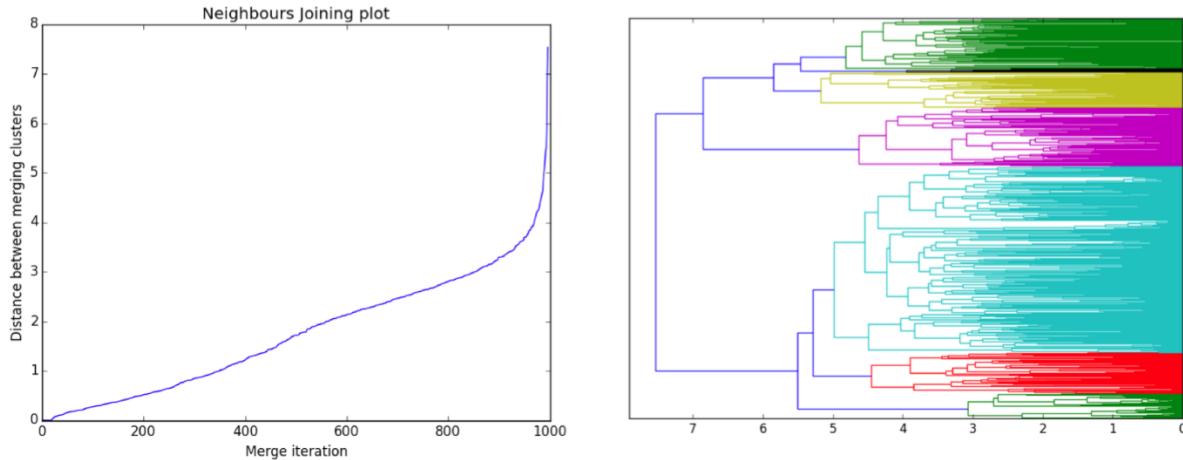


Рис. 2.8: Дендрограмма в задаче кластеризации писем по теме (на подвыборке из 1000 писем) и график зависимости расстояния между сливаемыми кластерами от номера итерации

На большей подвыборке из 1000 писем график стал более гладким, поскольку итерации стало больше, а дендрограмма получилась значительно сложнее.

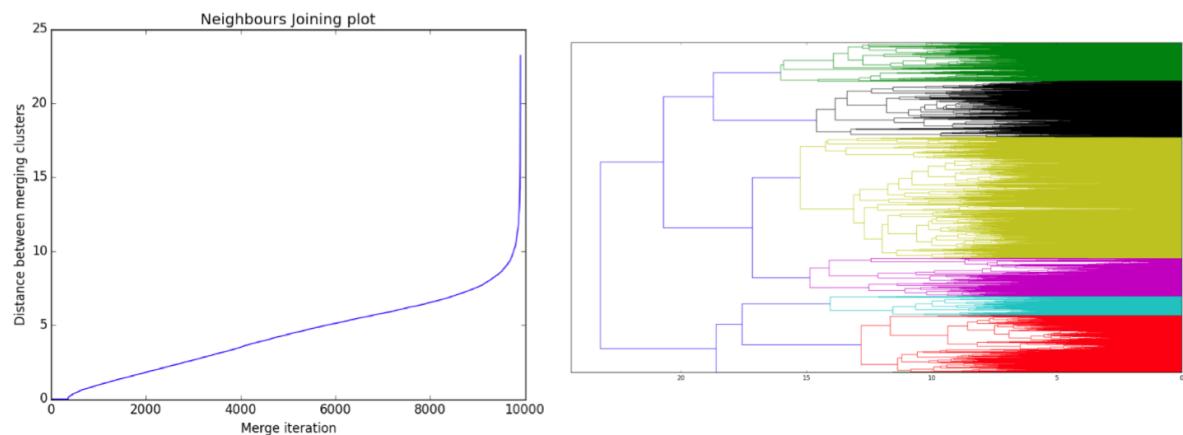


Рис. 2.9: Дендрограмма в задаче кластеризации писем по теме (на подвыборке из 10000 писем) и график зависимости расстояния между сливаемыми кластерами от номера итерации

На самой большой выборке из 10000 писем на графике есть область особенно быстрого роста. Любопытно, что изгиб на графике, за которым происходит особенно быстрый рост, возникал при разном расстоянии между сливаемыми кластерами на всех трех представленных графиках. Это связано с тем, что при исследовании разных подвыборок получается разный набор признаков, а следовательно сравнивать такие расстояния в экспериментах на разных подвыборках некорректно.

2.3.6. Перекос в размерах кластеров

Другая интересная особенность иерархической кластеризации состоит в том, что часто возникает один большой кластер и несколько небольших. Во многих задачах желательно получать кластеры более-менее похожие по размеру.

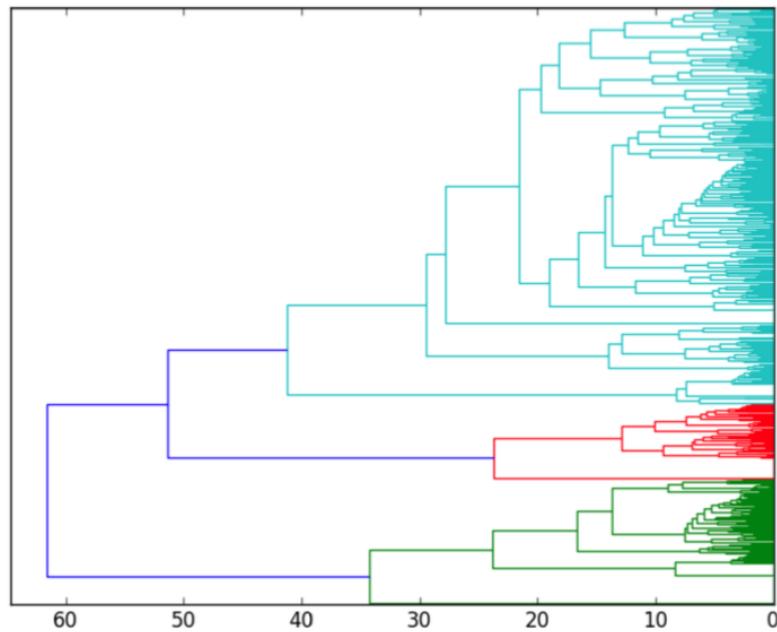


Рис. 2.10: Дендрограмма в задаче кластеризации текстов.

Это может быть связано с тем, что в признаках содержится слишком много шума. В этом случае можно попробовать понизить размерность пространства признаков, например с помощью SVD.

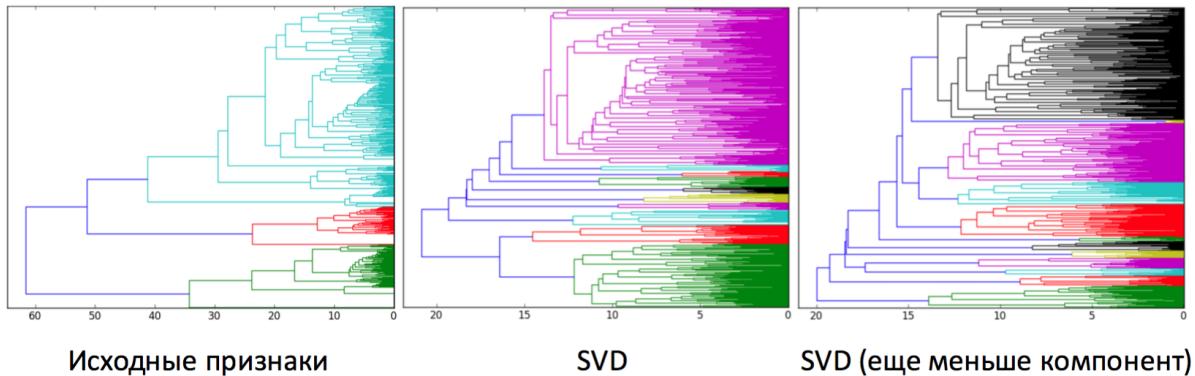


Рис. 2.11: Дендрограммы после понижения размерности признаков с помощью SVD

Если уменьшить размерность слишком сильно, то перегиб на графике зависимости расстояния между сливающимися кластерами от номера итерации пропадает совсем.

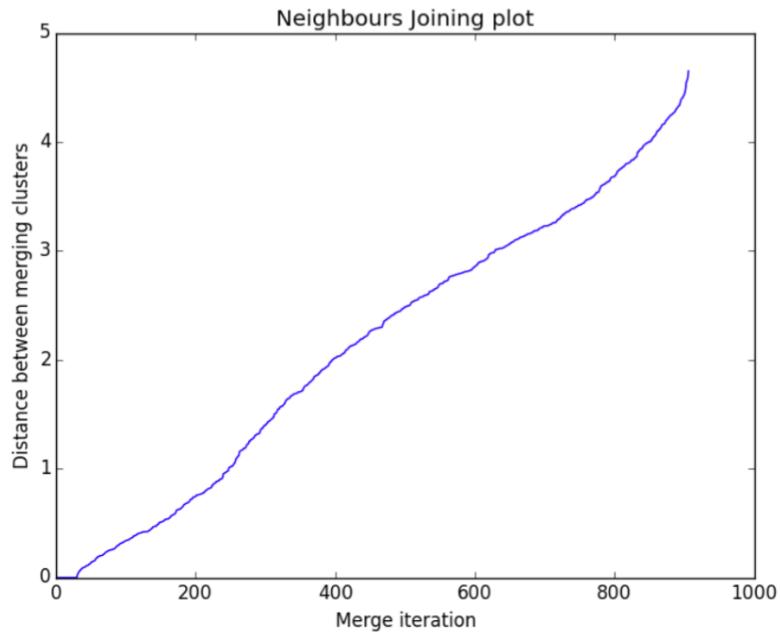


Рис. 2.12: График зависимости расстояния между сливаемыми кластерами от номера итерации

Это значит, что явно выделить кластеры уже нельзя — слишком сильно была понижена размерность пространства признаков.

2.4. Графовые методы кластеризации

2.4.1. Выделение связных компонент

Связность — это свойство, заключающееся в том, что из любой вершины графа можно попасть в любую другую вершину графа по ребрам. Связные компоненты — это подграфы в графе, которые обладают свойством связности, и в то же время никакие вершины из графа нельзя добавить в этот подграф, сохранив свойства связности.

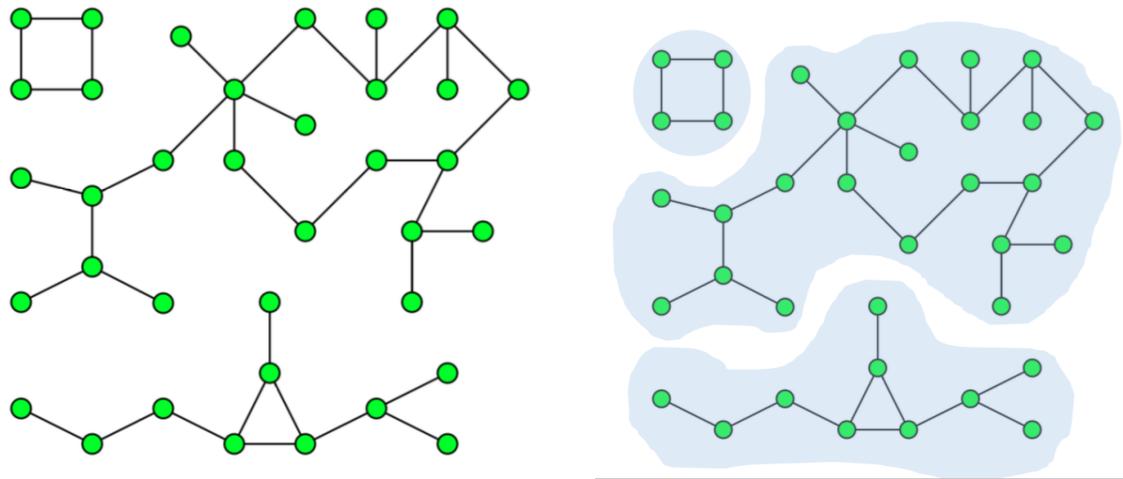


Рис. 2.13: Выделенные связные компоненты графа.

Граф, таким образом, может иметь одну или несколько связных компонент.

2.4.2. Кластеризация по компонентам связности

Кластеризация по компонентам связности происходит следующим образом: соединяются ребрами те объекты, расстояние между которыми меньше R , а затем в получившемся графе выделяются компоненты связности. Если граф получился связный (то есть компонента связности единственна), следует взять меньшее значение R . Однако непонятно, какое значение R нужно выбрать, чтобы получить конкретное значение числа кластеров K . Решить эту проблему позволяет другой простой графовый подход.

2.4.3. Минимальное оствовное дерево

Остовным деревом называется такой связный граф без циклов (дерево), в который входя все вершины исходного графа.

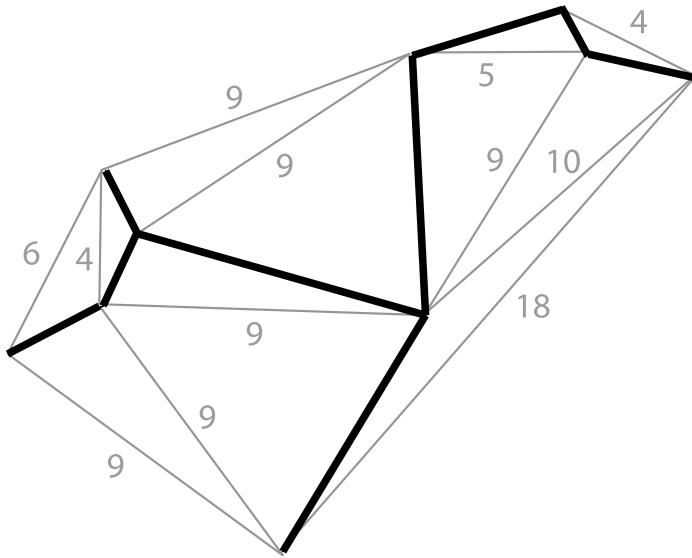


Рис. 2.14: Минимальное оствовное дерево

Минимальное оствовное дерево в связанным взвешенным неориентированном графе — это оствовное дерево этого графа, имеющее минимальный возможный вес, где под весом дерева понимается сумма весов входящих в него рёбер.

Минимальное оствовное дерево можно построить с помощью алгоритма Крускала (Kruskal):

1. Вначале текущее множество рёбер устанавливается пустым.
2. Пока это возможно, проводится следующая операция: из всех рёбер, добавление которых к уже имеющемуся множеству не вызовет появление в нём цикла, выбирается ребро минимального веса и добавляется к уже имеющемуся множеству.
3. Когда таких рёбер больше нет, алгоритм завершён.

Доказательство корректности алгоритма в данном курсе не приводится.

Чтобы решить задачу кластеризация с помощью оствовного дерева, нужно построить взвешенный граф, в котором вершины — это объекты, а веса ребер — это расстояния между объектами. Если необходимо построить K классов, то необходимо удалить $K - 1$ ребро с максимальными весами. Получившийся граф будет состоять из K компонент связности, каждую из которых можно интерпретировать как кластер.

2.5. Методы, основанные на плотности

2.5.1. Идея density-based методов

Идея density-based методов заключается в том, чтобы рассматривать плотность точек в окрестности каждого объекта выборки. Если в окрестности радиуса R с центром в некоторой точке выборки находится N или более других точек выборки, то такая точка считается основной. Здесь R и N — параметры алгоритма. Если точек меньше, чем N , но в окрестности рассматриваемой точки содержится основная точка, то такая точка называется граничной. В ином случае точка считается шумовой.

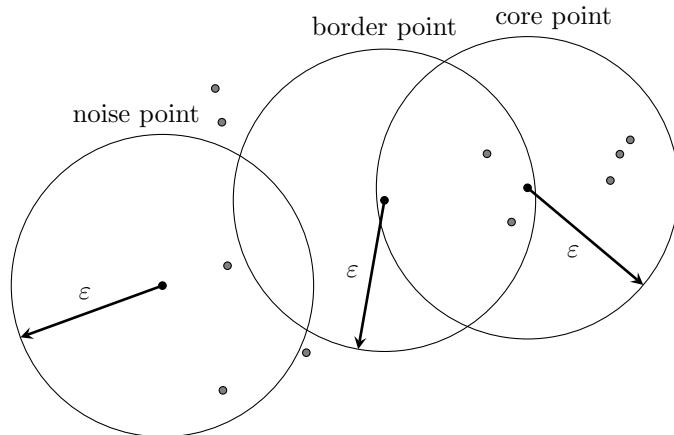


Рис. 2.15: Принцип работы алгоритма DBSCAN: изображены основная, пограничная и шумовая точки

2.5.2. Алгоритм DBSCAN

DBSCAN — это один из **density-based** методов, который состоит из следующих шагов:

1. Разделить точки на основные, пограничные и шумовые.
2. Отбросить шумовые точки.
3. Соединить основные точки, которые находятся на расстоянии ε друг от друга. В результате получается граф.
4. Каждую группу соединенных основных точек объединить в свой кластер (то есть выделить связные компоненты в получившемся графе).
5. Отнести пограничные точки к соответствующим им кластерам.

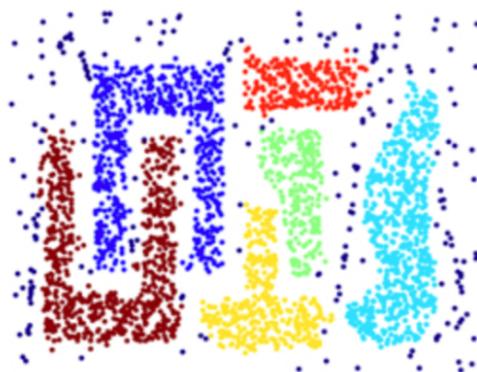


Рис. 2.16: Пример работы алгоритма DBSCAN

Как видно на последнем примере, DBSCAN хорошо справляется с нетривиальными формами кластеров и успешно отделяет шумовые точки, которые могли бы сильно испортить работу других алгоритмов кластеризации. Но DBSCAN часто неправильно определяет количество кластеров, особенно когда несколько кластеров расположены слишком близко друг к другу.

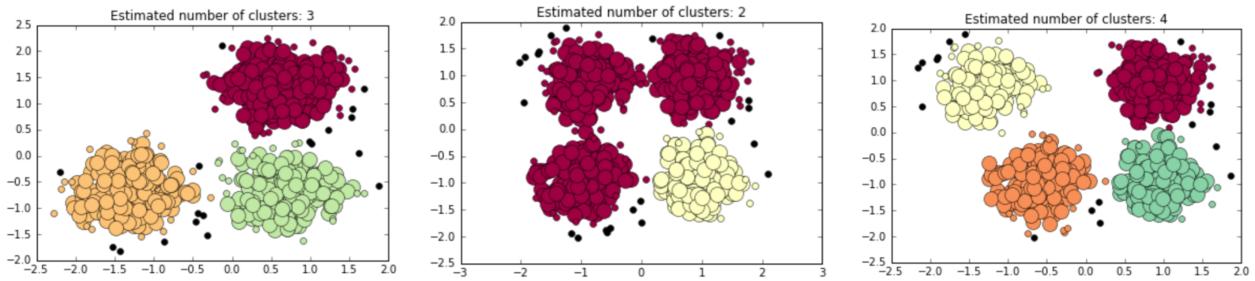
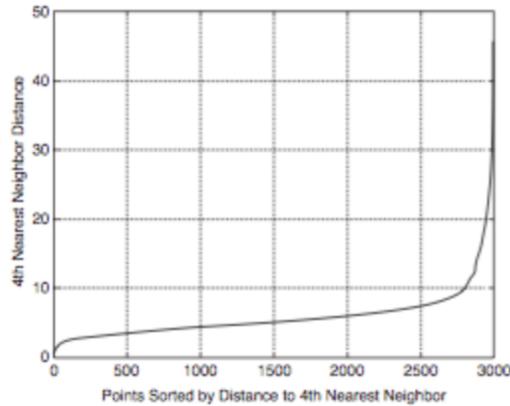


Рис. 2.17: Алгоритм DBSCAN правильно определил число кластеров (рис. слева). Ситуация, когда 3 из 4 кластеров слились воедино (рис. по центру). Эту ситуацию можно было бы избежать, сдвинув один из кластеров влево (рис. справа).

2.5.3. Подбор параметров в DBSCAN

Чтобы подобрать параметры в методе DBSCAN, имеет смысл построить график, по оси y у которого отложено расстояние до k -го соседа, а по оси x — количество точек, расстояние до k -го соседа которых меньше.



Начиная с некоторого номера на этом графике происходит резкий рост расстояния, а значит можно естественным способом выбрать расстояние, выбираемое в качестве радиуса R окрестности. Число k в данном случае есть пороговое количество точек в окрестности некоторой точки, необходимое, чтобы считать ее основной.

Таким образом, подбор параметров в методе DBSCAN заключается в рассмотрении графиков для разных значений k и определении оптимального значения R для каждого из них. Выбрать следует такую пару параметров k и R , чтобы количество шумовых точек было минимальным.

2.6. Оценка качества и рекомендации по решению задачи кластеризации

2.6.1. Внутрикластерное и межкластерное расстояния

Среднее внутрикластерное расстояние имеет следующий вид:

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

Если известны положения центров кластеров, можно записать аналогичную метрику, которая представляет собой среднее расстояние до центра кластера:

$$\Phi_0 = \sum_{y \in Y} \frac{1}{|K_y|} \sum_{i:y_i=y} \rho^2(x_i, \mu_y) \rightarrow \min.$$

Абсолютно аналогично вводится среднее межкластерное расстояние (теперь объекты берутся из разных кластеров):

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

Если известны центры кластеров, в качестве более простого варианта этой метрики можно взять (где μ — среднее арифметическое центров кластеров):

$$\Phi_1 = \sum_{y \in Y} \rho^2(\mu_y, \mu) \rightarrow \max.$$

Учесть значения обоих функционалов в одной метрике можно, если рассмотреть частное:

$$F_0/F_1 \rightarrow \min, \quad \text{или} \quad \Phi_0/\Phi_1 \rightarrow \min.$$

Но так или иначе эти функционалы имеют ряд недостатков. Например, с помощью таких функционалов нельзя подобрать количество кластеров. Действительно, при использовании таких метрик лучшим вариантом будет отнести каждую точку к своему собственному кластеру, так как в этом случае среднее внутрикластерное расстояние будет равно нулю.

2.6.2. Коэффициент силуэта для объекта

Коэффициент силуэта — метрика качества, которая позволяет выбрать количество кластеров. Коэффициент силуэта для некоторого фиксированного объекта определяется следующим образом:

$$s = \frac{b - a}{\max(a, b)},$$

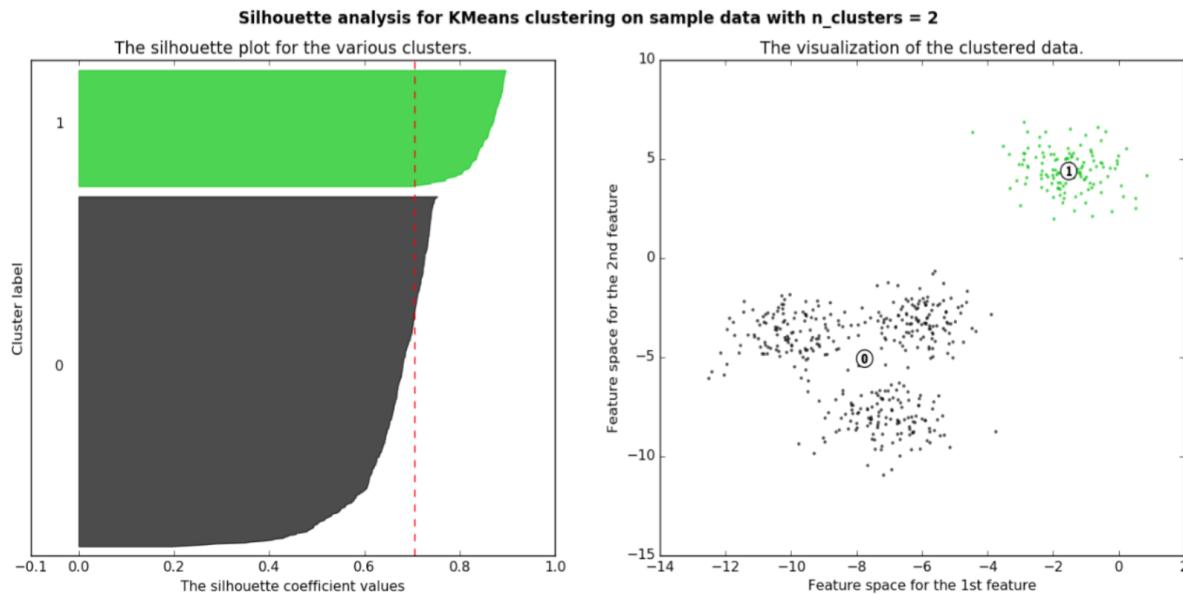
где a — среднее расстояние от данного объекта до других объектов из того же кластера, b — среднее расстояние от данного объекта до объектов из ближайшего другого кластера.

Обычно коэффициент силуэта положителен, но, вообще говоря, может меняться в пределах от -1 до 1 . Если объект находится вблизи границы кластера, и близко к нему расположен другой небольшой кластер, коэффициент силуэта для этого объекта получится очень маленьким.

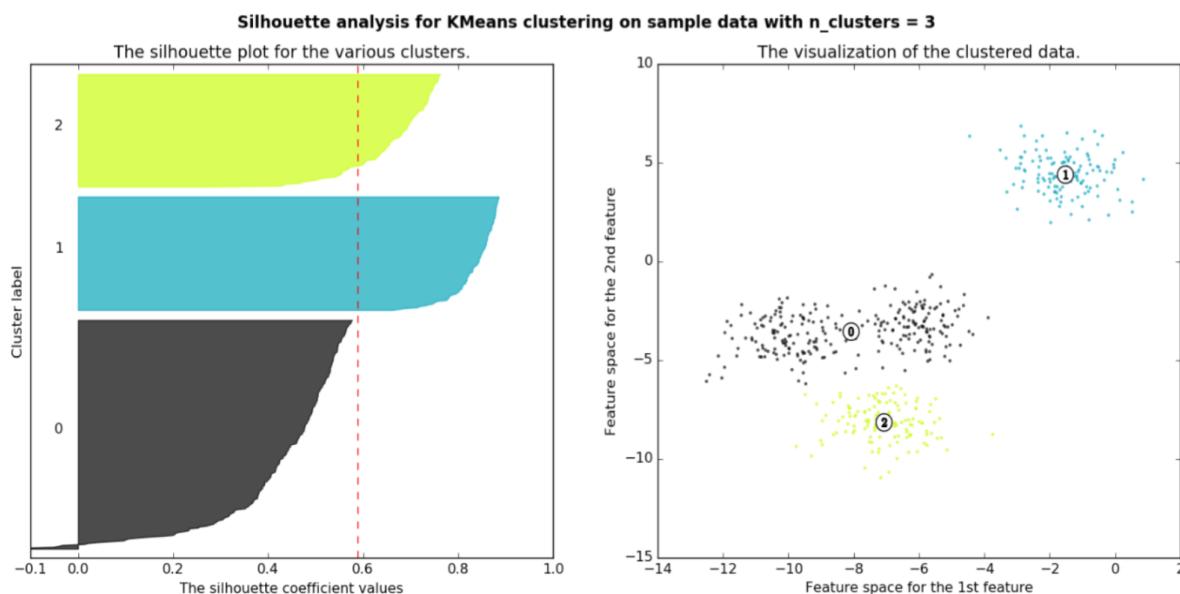
2.6.3. Использование коэффициента силуэта для оценки числа кластеров

Обычно вычисляют среднее значение коэффициента силуэта, а также для каждого кластера строят график, показывающий количество точек с различными значениями коэффициента силуэта. Среднее значение на таких графиках отмечается пунктирной линией. Для наглядности выборка будет изображаться справа от графика.

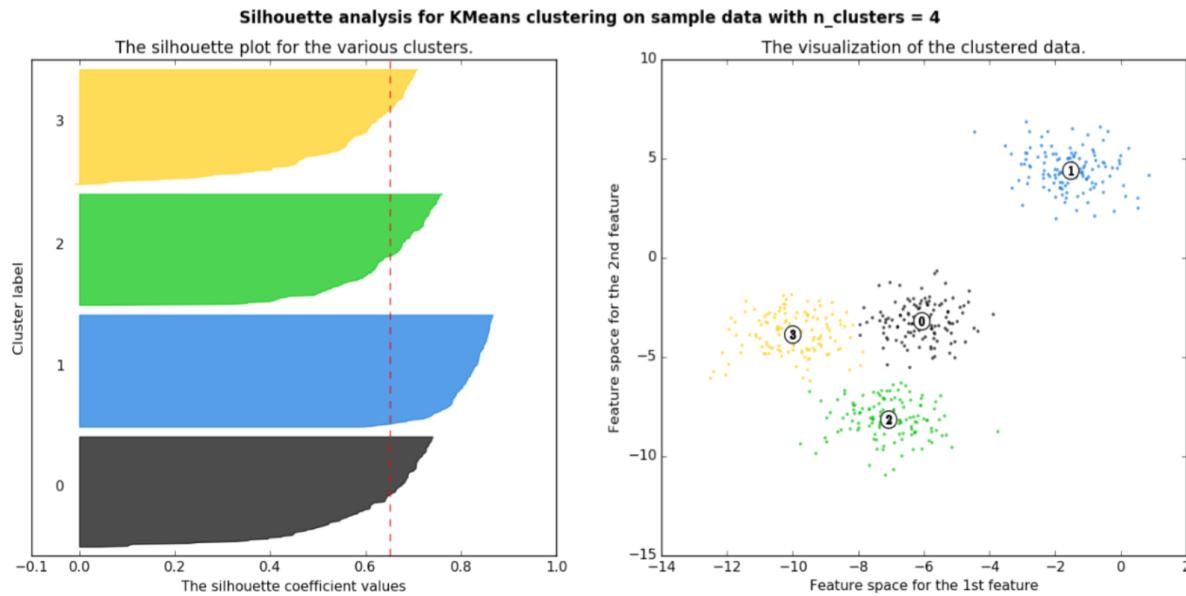
В случае двух кластеров в обоих из них есть точки, коэффициент силуэта для которых больше его среднего значения по всей выборке. Так же разброс значений по кластерам не очень большой.



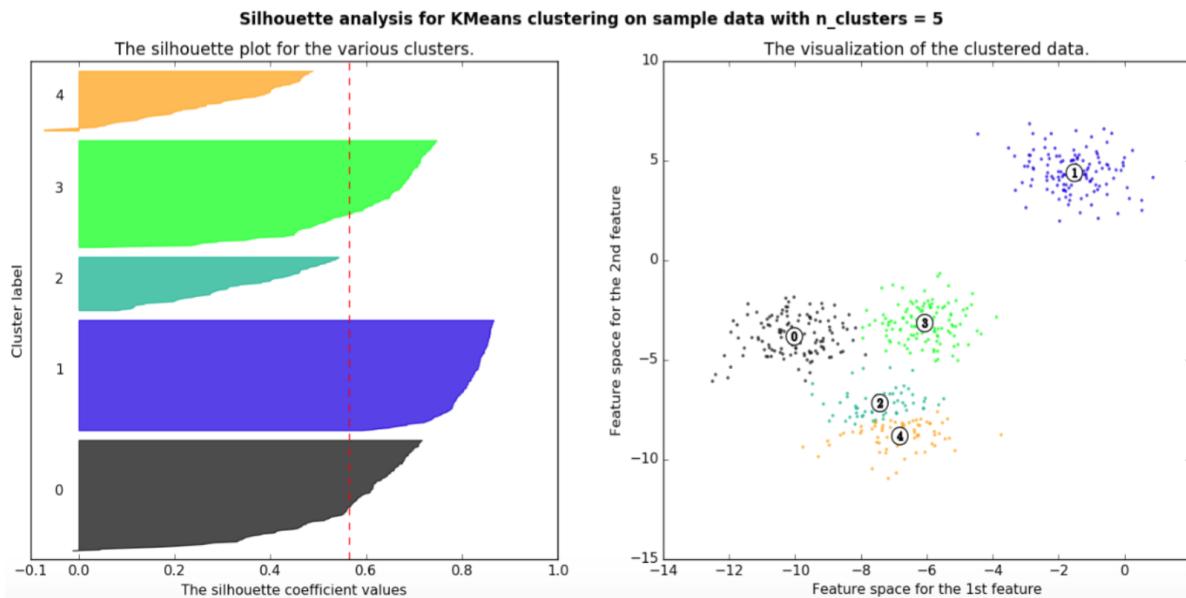
В случае трех кластеров разброс значений коэффициента силуэта в различных кластерах уже большой. При этом в третьем кластере все значения оказались меньше среднего значения по всей выборке. Такая кластеризация выглядит не очень убедительно и такое число кластеров лучше не выбирать.



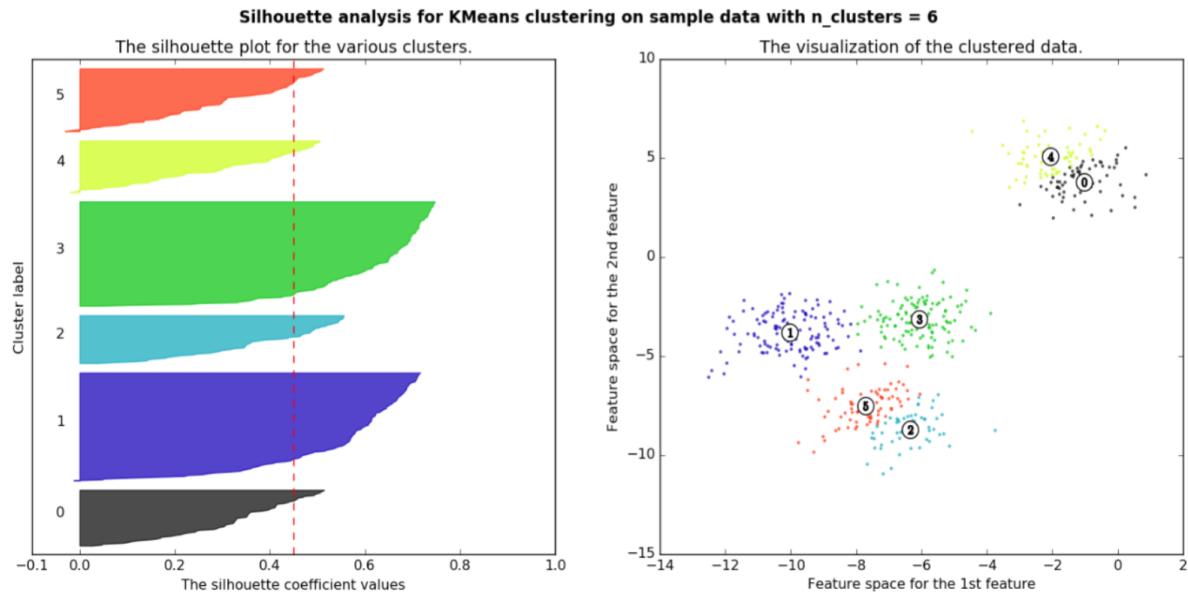
В случае четырех кластеров разброс значений коэффициента силуэта в различных кластерах небольшой, а также во всех кластерах есть точки, значения коэффициентов силуэта в которых больше среднего значения по выборке. Кластеризация с таким числом кластеров выглядит очень хорошо, в чем можно убедиться по визуализации на выборке. В реальных задачах часто признаков очень много и так просто визуализировать на плоскости не получается.



Если попробовать взять число кластеров равным пяти, то разброс по кластерам становится еще большие и опять же есть кластеры, у которых коэффициент меньше среднего значения по всей выборке.

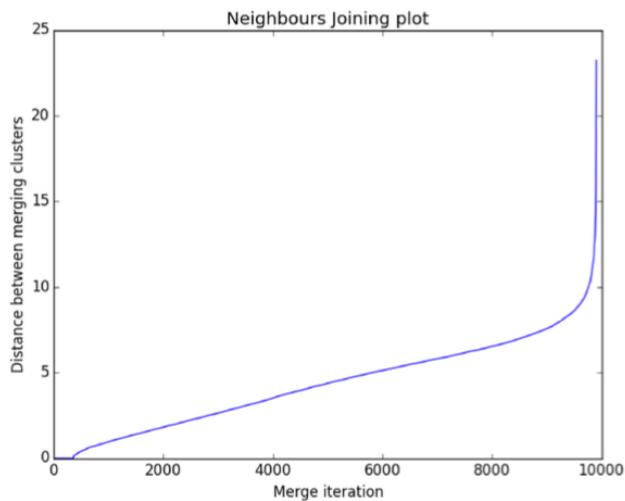


И в случае шести кластеров опять же получаем довольно большой разброс.



2.6.4. Проверка наличия кластерной структуры

Также полезно уметь проверять наличие кластерной структуры. Один из способов это проверить уже был озвучен — необходимо построить график расстояния, на котором происходит слияние, от номера итерации при агломеративной иерархической кластеризации.



В произвольном случае поступают следующим образом: генерируют p случайных точек из равномерного распределения и p случайных точек из обучающей выборки. После этого вычисляется так называемая статистика Хопкинса:

$$H = \frac{\sum_{i=1}^p w_i}{\sum_{i=1}^p u_i + \sum_{i=1}^p w_i},$$

w_i — расстояние от i -ой случайной точки до ближайшей случайной, u_i — расстояние от i -ой точки из выборки до другой ближайшей точки из выборки. Если статистика получается близкой к $1/2$, это значит, что выборка более-менее равномерно заполняет пространство признаков. Если же статистика получается вблизи нуля, это означает, что точки как-то группируются.

2.6.5. Выбор признаков

Кроме того, нужно уметь выбирать хорошие признаки для задачи кластеризации. Все метрики, которые обсуждались ранее, используют расстояние, которое зависит от выбора признаков. Поэтому использовать такие метрики для выбора признаков не получается. Вообще говоря, хотелось бы иметь возможность сравнивать качество кластеризации в зависимости от выбора признаков.

Пусть известна разметка, то есть к каким классам можно было бы отнести объекты выборки, причем этой разметки недостаточно для обучения классификатора. Тогда ее можно использовать для оценки качества кластеризации, например использовать метрику точности (accuracy). Другой подход состоит в использовании однородности, полноты и V-меры:

$$h = 1 - \frac{H(C|K)}{H(C)}, \quad c = 1 - \frac{H(K|C)}{H(K)}, \quad v = 2 \cdot \frac{h \cdot c}{h + c},$$

где $H(C)$ — энтропия класса, $H(K)$ — энтропия кластера, $H(C|K)$ — энтропия класса при условии кластера, $H(K|C)$ — энтропия кластера при условии класса. V-мера, как это видно из выражения, представляет собой среднее гармоническое однородности и полноты.

Однородность будет максимальной, если кластер состоит только из объектов одного класса, а полнота — если все объекты из класса принадлежат к одному кластеру.

Энтропия — мера неопределенности, мера незнания о том, какая будет конкретная реализация случайной величины:

$$H = - \sum_i p_i \log p_i.$$

Энтропия для класса вычисляется следующим образом:

$$H(C) = - \sum_{c=1}^{|C|} \frac{n_c}{n} \log \left(\frac{n_c}{n} \right), \quad P(c) = \frac{n_c}{n}.$$

Энтропия для класса при условии кластера вычисляется следующим образом:

$$H(C|K) = - \sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{n_{c,k}}{n} \log \left(\frac{n_{c,k}}{n_k} \right), \quad P(c|k) = \frac{n_{c,k}}{n}.$$

Если классы идеально совпадают с кластерами, то соответствующие условные энтропии $H(K|C)$ и $H(C|K)$ будут равняться нулю, а следовательно $h = c = 1$.

2.6.6. Привлечение ассессоров для оценки качества

Если разметки, о которой шла речь выше, нет, то можно:

- Использовать метрики без разметки
- Создать разметку с помощью ассессоров и использовать ее
- Предложить ассессорам отвечать на вопросы вида «допустимо ли эти объекты относить в один/разные кластеры». И, используя их ответы на уже готовой кластеризации, оценить ее качество.