ЦИФРОВАЯ КУЛЬТУРА
УНИВЕРСИТЕТ ИТМО

Задача кластеризации

Высшая Школа Цифровой Культуры
Университет ИТМО
dc@itmo.ru

Содержание

1 Введение

1.1 Мотивировка

Здравствуйте, уважаемые слушатели! Сегодня лекция будет посвящена одному из направлений «Обучения без учителя» – задаче кластеризации. Начнем же мы, конечно, с того, что попробуем выяснить: а в чем заключается задача кластеризации и зачем ее решать?

Давайте объясним суть на примере. Представим, что мы отправились за покупками в какой-нибудь гипермаркет. В гипермаркете, как известно, продается множество различных товаров: там можно найти и фрукты, и овощи, и мясо, и молочную продукцию, бытовую химию, а кое-где даже чайники, щетки стеклоочистителей, гладильные доски и другие полезные вещи. В то же время, все эти товары вряд ли лежат кучей прямо перед входом, не так ли? Товары разделены на какие-то группы или отделы (о чем часто свидетельствуют вывески под потолком вдоль ряда полок), причем, в зависимости от гипермаркета, как сами группы, так и их количество, могут быть разными. Более того, на территории магазина группы могут быть расположены поразному друг относительно друга. Решение относительно группировки товаров и расположения этих групп в магазине принимает отдел мерчендайзинга. Но из каких соображений?

Во-первых, разделение на группы должно быть интуитивно понятно покупателю: вряд ли кто-то будет искать молоко на соседней полке с куриными грудками (хотя такую группу и можно назвать, например, «продукты животного происхождения», но неужели это сразу приходит на ум?). Во-вторых, количество групп не должно быть слишком большим: отдельные группы «яблоки», «груши», «персики» находить будет куда сложнее, чем большую группу «фрукты» (или даже «фрукты и овощи»); совершенно ясно, что внутри этой укрупненной группы мы и сами быстро поймем, где лежат персики, а где груши, и выберем то, что нам по душе. Кстати, может быть мы и вовсе внезапно (!) предпочтем апельсины, так как именно сегодня они уж больно хорошо выглядят на прилавке.

Последнее наблюдение может подводить и к мысли, что расположение групп товаров в магазине (хотя это и несколько другая задача) – немаловажная деталь. Например, почти всегда торты и конфеты расположены рядом с отделом кофе и чая: люди, покупая чай, достаточно часто протягивают руку на соседнюю полку, чтобы купить что-то «сладенькое». Если бы конфет рядом не было, то соблазн был бы менее велик, конфет покупалось бы меньше, а значит и магазин нес бы потери в прибыли.

Ну а при чем тут кластеризация? Да вот при чем. В приведенном примере товары – это объекты, подлежащие кластеризации, а группы или отделы

– это кластеры, на которые они разбиты. Заранее как количество кластеров, так и их состав неизвестны, однако хочется верить в то, что наиболее близкие по типу товары должны находиться как можно ближе друг к другу (образовывать группу, кластер). Кроме того, чем больше объекты различаются, тем дальше друг от друга они должны находиться, впрочем, как и кластеры, которым они принадлежат, хотя это и не всегда так, но об этом чуть позже.

Давайте теперь ответим на вопрос: чем же все-таки обучение с учителем отличается от обучения без учителя? Мы видим, что в отличие от задач ветки обучения с учителем, в рассмотренном примере нам не даны правильные ответы, называемые нами откликами. Это и является глобальным отличием кластеризации от классификации: при решении задачи кластеризации набор классов (точнее – кластеров) изначально неизвестен, он должен быть определен либо алгоритмом, либо, что чаще, исследователем.

Подытожив, можно сказать, что задача кластеризации – это задача разделения множества объектов на группы, внутри которых находятся похожие объекты. Уже из приведенного примера ясно, что эта задача далеко не всегда имеет очевидное (и единственное) решение. Посмотрите, например, на рисунок 1. На какое количество кластеров вы бы разбили представленные объекты? На два, на четыре, на шесть или на какое-то другое количество? Конечно, правильного ответа на поставленный вопрос не существует. Определение количества кластеров – это сложная задача, которая часто требует вовлечения экспертного мнения для правильной интерпретации полученных результатов.

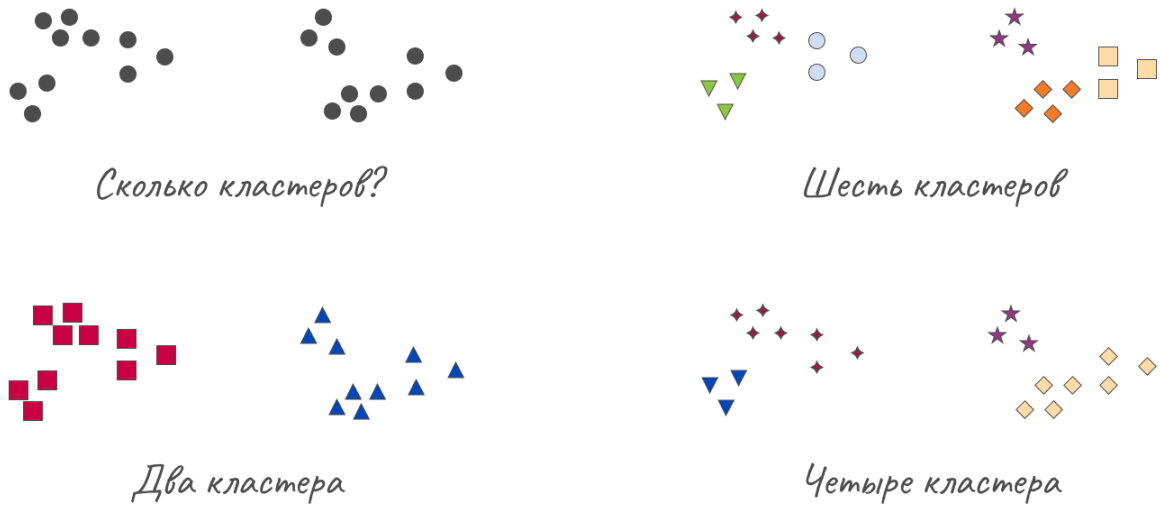

Рис. 1: Неоднозначность кластеризации

Разобравшись с концепцией задачи кластеризации, полезно понять и следующее: продукты в магазине, точки на плоскости – это объекты, характеристики которых нам часто видны и понятны. На практике же объекты могут

характеризоваться большим количеством признаков-предикторов и не поддаваться визуализации. Именно поэтому работу на разделение объектов на кластеры и отдают на откуп машинам, а аналитику остается не менее интригующая задача: проинтерпретировать полученное разбиение.

Что же, давайте теперь обратимся к вопросу о том, как машины разбирают предложенные данные на похожие и непохожие.

1.2 Виды кластеров и множественность алгоритмов

Наверное, у вас уже сформировалось интуитивное представление о задаче кластеризации: насыпанные каким-то образом точки нужно разделить на кучки, внутри которых точки «близки» друг к другу. Понятно, что такая вольная (и не совсем точная) постановка задачи дает волю фантазии, именно поэтому и способов решения задачи кластеризации довольно много. На рисунке 2 представлено шесть различных конфигураций «насыпанных» точек и результаты работы алгоритмов по разделению их на кластеры (разные цвета точек отвечают разным кластерам). Смотрите, насколько по-разному алгоритмы справляются с одной и той же задачей.

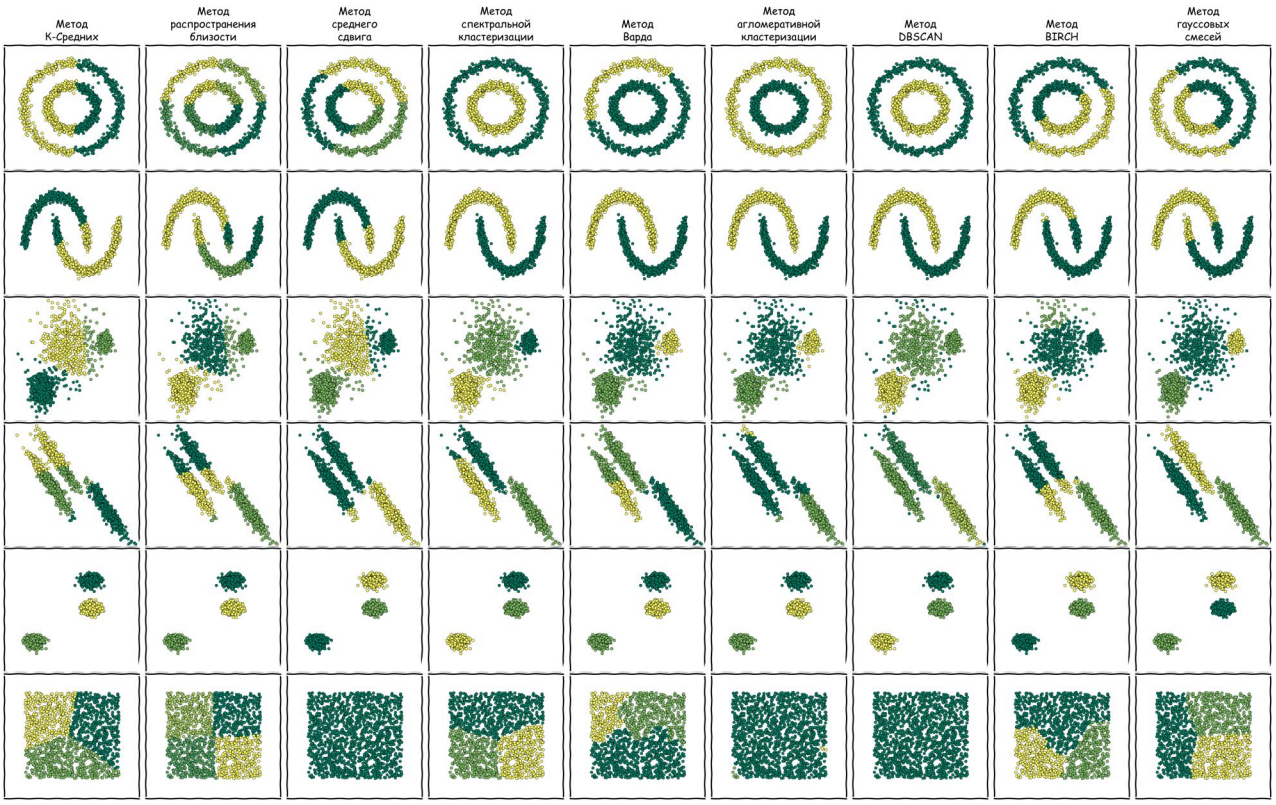

Рис. 2: Сравнение алгоритмов кластеризации

Давайте теперь проанализируем полученные результаты. Наверное, интуитивное представление о кластерах и «близости» внутри этих кластеров до этого момента больше всего походило на картину, соответствующую пятой конфигурации. Здесь кластеры представляют собой, как говорят «друг от друга отделенные плотные шаровые сгустки», и, как легко видеть, все представленные алгоритмы хорошо справляются с задачей. С точки зрения математики представленная ситуация характеризуется так: расстояние между объектами внутри кластера, как правило, много меньше, чем расстояние

между кластерами. Можно смело утверждать, что на настоящий момент кластеризация отделенных друг от друга плотных шаровых сгустков не представляет труда, большинство алгоритмов справляются с этой задачей легко и просто, а главное – качественно.

Теперь посмотрите на третью конфигурацию. Казалось бы, здесь тоже есть шаровые сгустки, и даже плотные, однако видно, что теперь они «плохо разделены» – между ними есть так называемые перемычки. Видно, что эти перемычки начинают мешать некоторым алгоритмам. Для большей наглядности можно привести и такую иллюстрацию для описываемой ситуации, см. рисунок 3. Перемычки между кластерами являются чем-то вроде мостов между островами и сильно усложняют работу алгоритмов. С точки зрения математики такая ситуация характеризуется так: расстояние между объектами внутри кластера сопоставимо с расстоянием между кластерами. Впрочем, многие алгоритмы кластеризации умеют справляться с подобными ситуациями.

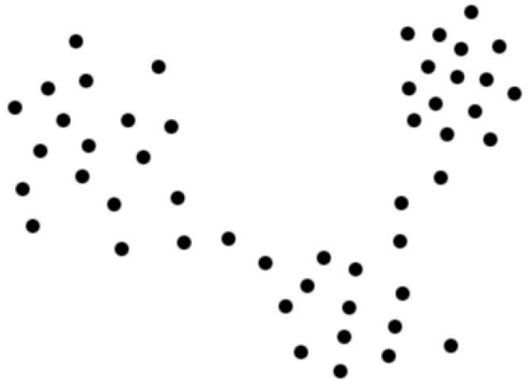

Рис. 3: Кластеры и перемычки между ними

Теперь обратимся к первой, второй и четвертой конфигурациям. А что тут? Перед нами явное опровержение интуитивным (и, честно говоря, несколько наивным) представлениям о том, что кластеры – это шаровые сгустки. Несмотря на то, что объекты, согласно нашему визуальному представлению, легко делятся на кластеры с четкой границей, алгоритмам это разделение совершенно неочевидно. Кластеры такой конфигурации называются ленточными. По своей сути, они представляют собой многообразия меньшей размерности, чем пространство, в котором они расположены: объекты внутри одного кластера могут быть расположены весьма далеко друг от друга, и это расстояние может быть весомо больше, чем расстояние между кластерами. Современные алгоритмы умеют находить ленточные кластеры, хотя это и не очень просто, об этом мы поговорим чуть позже.

Наконец, а что с последней конфигурацией? Наверное, можно сказать, что все объекты принадлежат одному кластеру, а может быть разумнее сказать и следующее: рассматриваемые признаки объектов не дают никакой полезной для кластеризации информации – кластеров просто нет. В этом примере особенно интересно, что многие алгоритмы все-таки пытаются эту массу разделить на какие-то части. Что же, это их право.

На практике случаются и другие ситуации, не нашедшие отражения в рассматриваемом примере. Например, кластеры могут быть окружены некоторым фоном-шумом, рисунок 4, накладываться друг на друга, рисунок 5, или вообще образовываться вовсе не по сходству, а по каким-то другим типам зависимостей, рисунок 6, могут быть и другие ситуации.

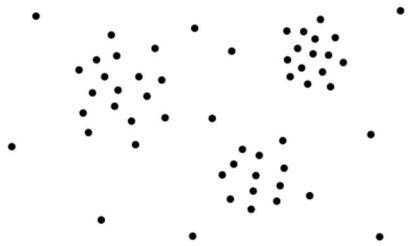

Рис. 4: Кластеры на фоне шума

Рис. 5: Пересекающиеся кластеры

Рис. 6: Другая зависимость «похожести»

Именно из-за такого разнообразия возможных ситуаций, выбор метода кластеризации – чуть ли не основная задача, которая стоит перед аналитиком-исследователем. Почему не основная? Потому что еще нужно определиться с тем, как измерять расстояния между объектами и между кластерами. Но обо всем по порядку. Итак, первый метод, который мы будем рассматривать – это метод K-средних (K-means).

2 Метод K-средних

2.1 Предварительные сведения и выбор расстояния

Перед тем как начать обсуждение алгоритма K-средних, дадим формальное определение понятию кластера.

Определение 2.1.1 *Пусть $X \neq \varnothing$ – множество всех рассматриваемых объектов. Непустые множества $C_1, C_2, \ldots, C_K \subset X$ называются кластерами, если их объединение совпадает с X, а пересечение любых двух из них (конечно, с различными индексами) есть пустое множество, то есть*

$$X = \bigcup_{i=1}^{K} C_i, \quad \textit{причем} \quad C_i \cap C_{i'} = \varnothing \quad \textit{при} \quad i \neq i', \quad i, i' \in \{1, 2, ..., K\}.$$

Замечание 2.1.1 *В математике семейство множеств $C_1, C_2, \ldots, C_K$ из предыдущего определения называют разбиением множества X.*

Классически будем считать, что у нас имеется набор данных $X = (x_1, x_2, ..., x_n)$ объема n с числовыми признаками, где

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}), \quad i \in \{1, 2, ..., n\},$$

и этот набор данных нам и требуется разбить на кластеры. На первый взгляд задача распределения n объектов по K кластерам выглядит не очень простой, особенно если K и n достаточно велики, ведь существует K^n вариантов построения искомого разбиения. Перебор всех возможных вариантов в поисках «оптимального», конечно, не самое разумное, а зачастую и вычислительно невозможное решение. Да и что вообще значит – «оптимальный вариант»?

Интуиция подсказывает нам, что кластеризация тем лучше, чем более «похожи» объекты в рамках одного кластера. Похожесть, видимо, эквивалентна близости объектов с точки зрения расстояния между ними. Если

вспомнить случай отделенных плотных шаровых сгустков, о которых мы говорили ранее, то можно сформулировать следующую эмпирическую характеристику «оптимальности»: в случае «оптимальной» кластеризации расстояние между любыми двумя объектами внутри одного кластера невелико по сравнению с расстоянием между объектами, находящимися в двух разных кластерах. Значит, можно прийти к следующему умозаключению: разумно считать, что кластеризация проведена удачно, если среднее значение попарных расстояний между объектами каждого кластера мало. Вот он и критерий!

Замечание 2.1.2 *Еще раз обратим ваше внимание на то, что озвученный «критерий» – это не рецепт на все случаи жизни. Это – вольно сформулированный принцип довольствования близостью в алгоритме K-средних. Вспоминая результаты работы алгоритмов, изображенные на рисунке 2, видно, что, в частности, когда сгустки плохо отделены, такой критерий уже работает не очень хорошо, а в случае ленточных кластеров или шума – не работает вовсе. Поэтому область применения алгоритма K-средних не очень велика: она ограничивается весьма жесткими предположениями на форму и расположение кластеров – плотные хорошо разделенные шаровые сгустки.*

Итак, разобравшись с идейной стороной вопроса, перейдем к формальным определениям.

Определение 2.1.2 *Пусть* $X = \{x_1, x_2, ..., x_n\}$ *– множество рассматриваемых объектов,* $C_1, C_2, ..., C_K$ *– разбиение* X*,* d *– функция расстояния, заданная на* X*.*

Внутрикластерным расстоянием $W(C_k)$ *в кластере* C_k *называют сумму попарных расстояний между всеми объектами этого кластера, то есть*

$$W(C_k) = \sum_{x_i, x_{i'} \in C_k} d(x_i, x_{i'}), \quad k \in \{1, 2, ..., K\}.$$

Теперь совершенно ясно, как ввести понятие среднего внутрикластерного расстояния.

Определение 2.1.3 *В обозначениях предыдущего определения, средним внутрикластерным расстоянием в кластере* C_k *называется величина, равная*

$$\frac{W(C_k)}{|C_k|}, \quad k \in \{1, 2, ..., K\},$$

где $|C_k|$ *– количество объектов, принадлежащих кластеру* C_k*.*

Итак, чтобы минимизировать средние внутрикластерные расстояния, можно попробовать минимизировать их сумму по всем кластерам, то есть минимизировать следующее выражение:

$$\sum_{k=1}^{K} \frac{W(C_k)}{|C_k|} = \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{x_i, x_{i'} \in C_k} d(x_i, x_{i'}).$$

Осталось определиться с функцией расстояния. В качестве последней мы будем использовать квадрат евклидова расстояния.

Замечание 2.1.3 *Напомним, что евклидово расстояние между объектами*

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \quad и \quad x_{i'} = (x_{i'1}, x_{i'2}, \ldots, x_{i'p})$$

определяется следующим образом:

$$d_E(x_i, x_{i'}) = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}.$$

Квадрат евклидова расстояния вычисляется, как

$$d_E^2(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2.$$

При таком выборе функции расстояния близкие объекты (евклидово расстояние между которыми меньше единицы) вносят меньший вклад в минимизируемую сумму, а далекие (евклидово расстояние между которыми больше единицы) – больший. Само минимизируемое выражение выглядит следующим образом:

$$\sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{x_i, x_{i'} \in C_k} d_E^2(x_i, x_{i'}).$$

Замечание 2.1.4 *Обратим внимание на один технический момент. Отслеживать изменение значения заявленной минимизируемой функции*

$$\sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{x_i, x_{i'} \in C_k} d_E^2(x_i, x_{i'})$$

достаточно сложно, так как при изменении принадлежности объекта кластеру во внешней сумме меняется не только числитель дроби, но и знаменатель, причем сразу у двух слагаемых: у слагаемого, отвечающего кластеру из которого ушел объект, и у слагаемого, отвечающего кластеру, в который пришел объект.

Снова обращаясь к геометрическим соображениям (шаровые сгустки) становится понятно, что, скорее всего, полезной характеристикой кластера является точка, являющаяся центром масс объектов, принадлежащих кластеру, – так называемый центроид кластера.

Определение 2.1.4 *Центроидом* $\overline{x}_k = (\overline{x}_{k1}, \ldots, \overline{x}_{kp})$ *кластера* C_k *называется объект, координаты которого вычисляются следующим образом:*

$$\overline{x}_{kj} = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_{ij}, \ j = \{1, 2, \ldots, p\}.$$

Оказывается, справедлива следующая лемма: при использовании в качестве функции расстояния квадрата евклидова расстояния, среднее внутрикластерное расстояние в кластере C_k равно удвоенной сумме квадратов расстояний от объектов кластера C_k до центроида этого кластера $\overline{x}_k$.

Лемма 2.1.1 *В рамках обозначений, введенных ранее, справедливо равенство:*

$$\frac{1}{|C_k|} \sum_{x_i, x_{i'} \in C_k} d_E^2(x_i, x_{i'}) = 2 \sum_{x_i \in C_k} d_E^2(x_i, \overline{x}_k).$$

Доказательство. Для простоты и наглядности обозначений предположим, что кластер C_k содержит n объектов $x_1, x_2, ..., x_n$,

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}), \quad i \in \{1, 2, ..., n\},$$

тогда

$$\frac{1}{|C_k|} \sum_{x_i, x_{i'} \in C_k} d_E^2(x_i, x_{i'}) = \frac{1}{n} \sum_{i,j=1}^{n} d_E^2(x_i, x_j) = \frac{1}{n} \sum_{i,j=1}^{n} \sum_{t=1}^{p} (x_{it} - x_{jt})^2.$$

Кроме того,

$$2 \sum_{x_i \in C_k} d_E^2(x_i, \overline{x}_k) = 2 \sum_{i=1}^{n} \sum_{t=1}^{p} (x_{it} - \overline{x}_{kt})^2.$$

Итого, нужно доказать, что

$$\frac{1}{n} \sum_{i,j=1}^{n} \sum_{t=1}^{p} (x_{it} - x_{jt})^2 = 2 \sum_{i=1}^{n} \sum_{t=1}^{p} (x_{it} - \overline{x}_{kt})^2.$$

Пусть $t \in \{1, 2, ..., p\}$, справедлива цепочка преобразований:

$$\frac{1}{n} \sum_{i,j=1}^{n} (x_{it} - x_{jt})^2 = \frac{1}{n} \sum_{i,j=1}^{n} (x_{it}^2 - 2x_{it}x_{jt} + x_{jt}^2) = \sum_{i=1}^{n} x_{it}^2 - \frac{2}{n} \sum_{i,j=1}^{n} x_{it}x_{jt} + \sum_{j=1}^{n} x_{jt}^2 =$$

$$= 2\left(\sum_{i=1}^{n} x_{it}^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_{it}\right)^2\right) = 2n\left(\frac{1}{n}\sum_{i=1}^{n} x_{it}^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_{it}\right)^2\right).$$

Заметим, что последнее выражение – это умноженная на $2n$ выборочная дисперсия набора данных $X_t = (x_{1t}, x_{2t}, ..., x_{nt})$ – t-тых координат исходного набора данных. Значит, в стандартных обозначениях,

$$2n\left(\frac{1}{n}\sum_{i=1}^{n} x_{it}^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_{it}\right)^2\right) = 2n\left(\overline{X_t^2} - \overline{X_t}^2\right) = 2n \cdot S^2(X_t).$$

Пользуясь определением выборочной дисперсии, получим

$$2n \cdot S^2(X_t) = 2n \cdot \frac{1}{n}\sum_{i=1}^{n}(x_{it} - \overline{x}_{kt})^2 = 2\sum_{i=1}^{n}(x_{it} - \overline{x}_{kt})^2.$$

Итого, при $t \in \{1, 2, ..., p\}$ справедливо равенство

$$\frac{1}{n}\sum_{i,j=1}^{n}(x_{it} - x_{jt})^2 = 2\sum_{i=1}^{n}(x_{it} - \overline{x}_{kt})^2.$$

Для завершения доказательства осталось просуммировать эти равенства по t. □

Итого, используя лемму, сумма средних внутрикластерных расстояний переписывается в виде

$$\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{x_i, x_{i'} \in C_k} d_E^2(x_i, x_{i'}) = 2\sum_{k=1}^{K}\sum_{x_i \in C_k} d_E^2(x_i, \overline{x}_k) = 2\sum_{k=1}^{K}\sum_{x_i \in C_k}\sum_{j=1}^{p}(x_{ij} - \overline{x}_{kj})^2.$$

Значит, минимизация первого выражения эквивалентна минимизации последнего. Так как двойка, стоящая перед суммой, не зависит от объектов и не влияет на минимизацию, впредь мы будем рассматривать (и минимизировать) лишь следующее выражение:

$$\sum_{k=1}^{K}\sum_{x_i \in C_k}\sum_{j=1}^{p}(x_{ij} - \overline{x}_{kj})^2.$$

Итого, задача минимизации суммы средних внутрикластерных «расстояний» свелась к задаче минимизации суммы по всем кластерам квадратов расстояний от объектов кластера до центроида соответствующего кластера. Центроид на каждой итерации является чем-то вроде инварианта, относительно которого очень удобно следить за процессом кластеризации. Что же, теперь мы готовы сформулировать способ кластеризации методом K-средних.

2.2 Описание алгоритма

Итак, пришло время сформулировать алгоритм применения метода K-средних. Пусть имеется набор данных $X = (x_1, x_2, ..., x_n)$ объема n с числовыми признаками, где

$$x_i = (x_{i1}, x_{i2}, ..., x_{ip}), \quad i \in \{1, 2, ..., n\}.$$

1. Выбирается число $K \in \mathbb{N}$.

2. **Инициализация.** Каждый объект случайным образом относят к какому-то кластеру из набора $C_1, C_2, ..., C_K$.

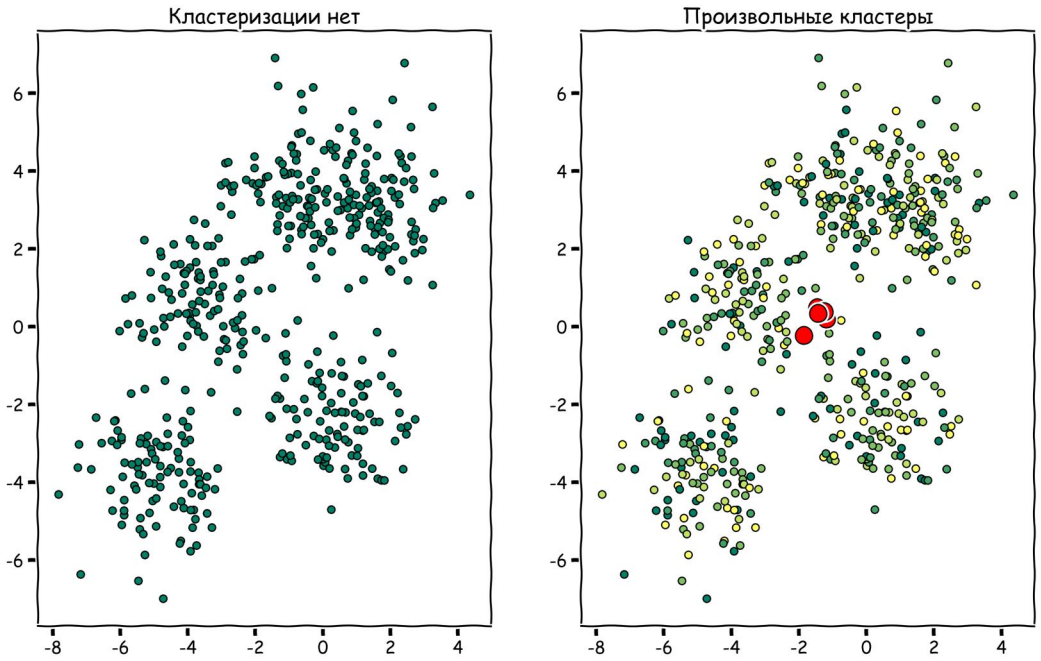

Рис. 7: Инициализация

3. **Нахождение центроидов.** Для каждого кластера C_k находят координаты центроида:

$$\overline{x}_k = (\overline{x}_{k1}, \overline{x}_{k2}, ..., \overline{x}_{kp}), \quad \overline{x}_{kj} = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_{ij},$$

где $k \in \{1, 2, \ldots, K\}$, $j \in \{1, 2, \ldots, p\}$.

4. **Вычисление квадратов расстояний до центроидов.** Находят квадрат евклидова расстояния от i-го объекта до центроида каждого

кластера:

$$d_E^2(x_i, \overline{x}_k) = \sum_{j=1}^{p} (x_{ij} - \overline{x}_{kj})^2,$$

$$k \in \{1, 2, \ldots, K\}, \quad i \in \{1, 2, \ldots, n\}.$$

5. **Перераспределение.** Объект x_i относят к кластеру с наиболее близким к нему центроидом, то есть $x_i \in C_{k^*}$, где k^* – любое из решений задачи

$$\operatorname*{Arg\,min}_{k \in \{1,2,\ldots,K\}} d_E^2(x_i, \overline{x}_k).$$

Отметим, что если среди решений поставленной задачи есть номер текущего кластера, то объект не меняет своей кластерной принадлежности.

6. Шаги 2 – 6 повторяются, пока объекты не перестанут перераспределяться по кластерам

На рисунке 7 слева представлены исходные данные: каждый объект обладает двумя числовыми предикторами, число K из визуальных соображений выбрано равным пяти. На том же рисунке справа сначала произведена инициализация: каждый объект случайным образом отнесен к одному из пяти возможных кластеров (разным кластерам соответствуют разные цвета), а затем вычислены (и нарисованы) центроиды каждого из кластеров – им отвечают жирные красные точки в центре картинки. На рисунке 8 видно расположение центроидов и принадлежность объектов к кластерам после первой, второй и так далее до шестой итераций. Скорее всего, финальный вариант кластеризации никого не удивил – он был ожидаем с самого начала.

Замечание 2.2.1 *Отметим важное замечание. Так как алгоритм K-средних использует для определения близости понятие расстояния, то признаки кластеризуемых объектов перед началом кластеризации имеет смысл либо стандартизировать, либо нормировать.*

Замечание 2.2.2 *Отметим (пока лишь на словах), что описанный алгоритм сходится (то есть существует шаг, после которого объекты перестают перераспределяться по кластерам). Почему? Давайте посмотрим. На этапе перераспределения объектов по новым кластерам выражение*

$$\sum_{k=1}^{K} \sum_{x_i \in C_k} \sum_{j=1}^{p} (x_{ij} - \overline{x}_{kj})^2,$$

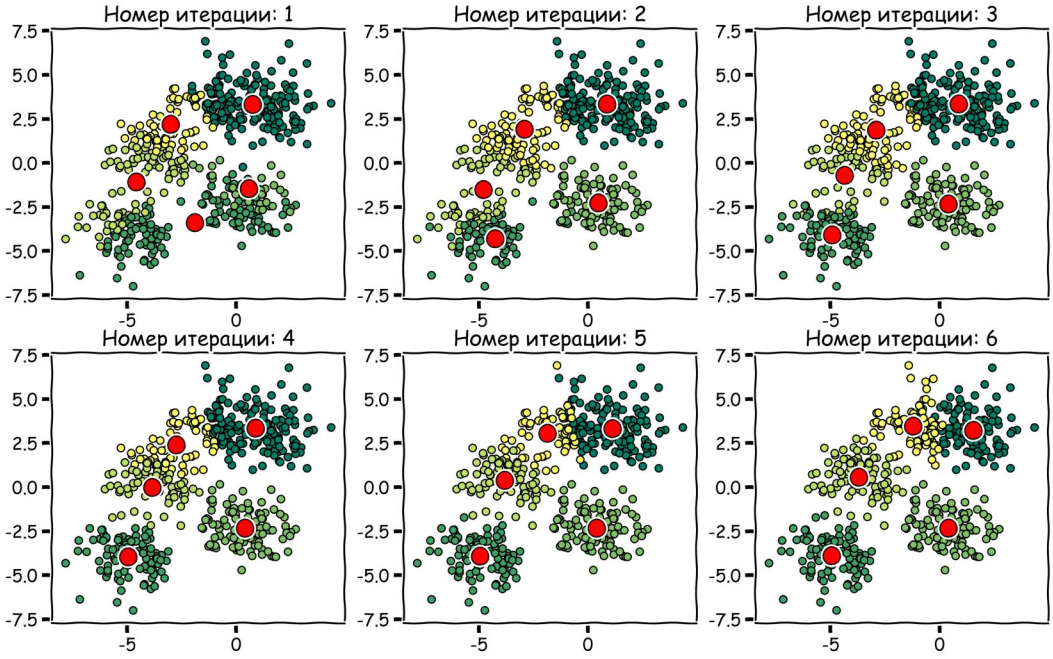

Рис. 8: Итерации кластеризации

которое мы минимизируем, не увеличивается, так как объекты, если и меняют кластер, переходят в тот, центроид которого ближе. В итоге квадрат расстояния до старого кластера (большее «расстояние») меняется на квадрат расстояния до нового кластера (меньшее «расстояние»). Нахождение новых центроидов на каждой итерации, в свою очередь, как минимум не увеличивает сумму квадратов расстояний до центроида внутри каждого кластера.

Определение 2.2.1 *Ситуация, когда алгоритм сошелся, то есть когда объекты перестали менять свою кластерную принадлежность, называется локальным оптимумом.*

Почему локальным, спросите вы? Дело в том, что результат кластеризации методом K-средних очень чувствителен к начальной инициализации (иными словами – к выбору начальных центроидов). Меняя начальную инициализацию, вообще говоря, меняется и конечная кластеризация. Продемонстрируем это на синтетическом примере. Рассмотрим один и тот же набор данных. Каждый объект обладает двумя атрибутами и может быть изображен на координатной плоскости. Проведем кластеризацию, используя приведенный алгоритм, распределяя объекты на $K = 5$ кластеров. Отличие в каждой из четырех ситуаций заключается лишь в изначальной инициализации объектов (шаг 2). Результаты представлены на рисунке 9.

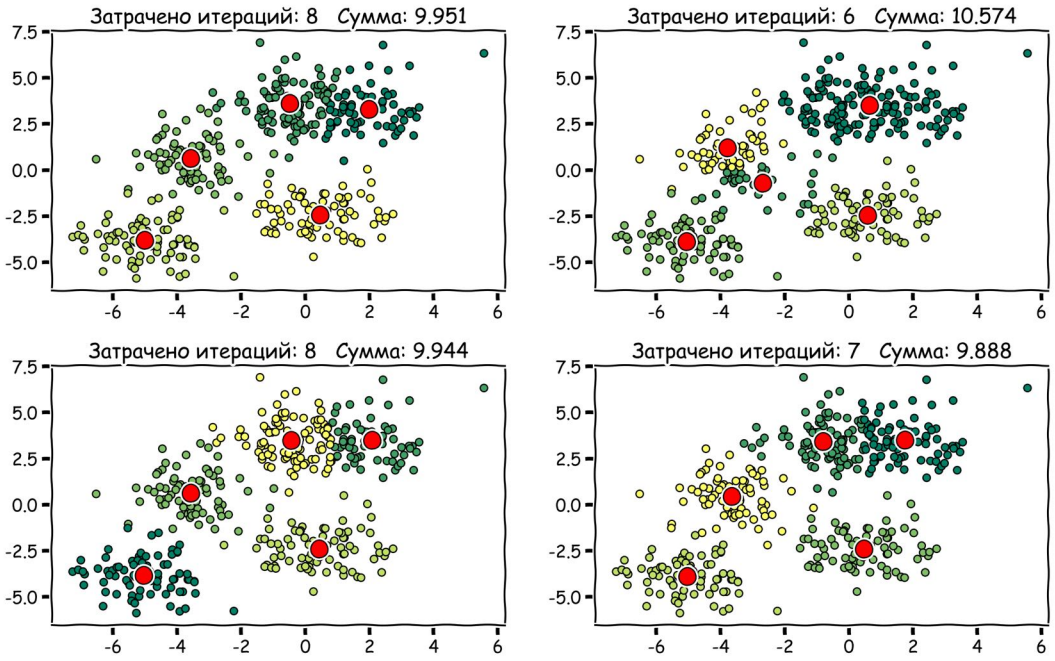

Рис. 9: Различные варианты инициализации

Как можно заметить, несмотря на то, что исходные объекты одни и те же, в результате работы алгоритма кластеризовались они по-разному (и дело вовсе не в цветах, а в качественном составе кластеров). Особенно заметно отличие верхней правой конфигурации от трех других. Впрочем, если приглядеться, легко найти несоответствия и в остальных конфигурациях. Возникли эти несоответствия из-за того, что кластеры, хоть и четко видны человеческому (и компьютерному) глазу, плохо отделены – они соединены мостами или шумом, что мешает однозначной кластеризации. Впрочем, мы об этом подробно говорили еще во введении. Кроме того, причиной может быть неудачная начальная инициализация. О том, что это такое, мы поговорим чуть позже. Сейчас же обратимся к расчетному примеру, чтобы детально разобрать все этапы алгоритма.

2.3 Пример: хруст и сладость продуктов

Разберем описанный алгоритм, как говорят, «на пальцах», а для этого рассмотрим уже знакомый по предыдущим лекциям пример про сладость и хруст различной пищи. Будем считать, что задача заключается не в классифицировании нового объекта, а в выделении групп объектов по уже имеющимся данным. Естественно, эту задачу легко решить устно, основываясь на жизненном опыте, но мы, как нормальные герои, конечно же пойдем в обход. В конце концов, нам важно понять принцип, а не решить конкретную

гастрономическую задачу. Исходные данные представлены в таблице. Для простоты вычислений, не будем нормировать или стандартизировать данные. Снова подчеркнем, что на практике это шаг является обязательным.

Номер	**Продукт**	**Сладость**	**Хруст**
1	банан	10	1
2	апельсин	7	4
3	виноград	8	3
4	креветка	2	2
5	бекон	1	5
6	орехи	3	3
7	сыр	2	1
8	рыба	3	2
9	огурец	2	8
10	яблоко	9	8
11	морковь	4	10
12	сельдерей	2	9
13	салат	3	7
14	груша	8	7
15	перец	6	9

Первым и, наверное, самым важным встает вопрос: на какое же число кластеров мы будем делить исходные объекты? На большом количестве данных с заведомо неизвестным числом кластеров имеет смысл провести несколько экспериментов, чтобы подобрать более-менее оптимальное число, если, конечно, это возможно в рамках поставленной задачи (нет строгих ограничений по времени и по ресурсам). Мы тоже могли бы решить эту задачу с различными значениями, но, так как данные нам интуитивно понятны, давайте рассмотрим распределение по трем кластерам (ожидаемо – овощи, фрукты и протеины) и посмотрим, что у нас получится.

Согласно описанному алгоритму, после выбора количества кластеров необходимо произвести инициализацию – отнести каждый объект исходных данных к одному из трех кластеров случайным образом. Объекты и присвоенные им номера кластеров можно увидеть в таблице:

Номер	**Продукт**	**Сладость**	**Хруст**	**Кластер**
1	банан	10	1	2
2	апельсин	7	4	2
3	виноград	8	3	2
4	креветка	2	2	3
5	бекон	1	5	1
6	орехи	3	3	1
7	сыр	2	1	3
8	рыба	3	2	1
9	огурец	2	8	2
10	яблоко	9	8	1
11	морковь	4	10	2
12	сельдерей	2	9	2
13	салат	3	7	2
14	груша	8	7	1
15	перец	6	9	2

Далее находим центроиды каждого кластера. Кластеру 1 принадлежат следующие точки: $(1,5),(3,3),(3,2),(9,8),(8,7)$. Найдем координаты $\overline{x}_{11},\overline{x}_{12}$ центроида этого кластера:

$$\overline{x}_{11}=\frac{1+3+3+9+8}{5}=4.8,$$

$$\overline{x}_{12}=\frac{5+3+2+8+7}{5}=5.$$

Аналогичным образом найдем координаты $\overline{x}_{k1},\overline{x}_{k2}$, $k\in\{2,3\}$, центроидов кластеров C_2 и C_3.

$$\overline{x}_{21}=\frac{10+7+8+2+4+2+3+6}{8}=5.25,$$

$$\overline{x}_{22}=\frac{1+4+3+8+10+9+7+9}{8}=6.375,$$

$$\overline{x}_{31}=\frac{2+2}{2}=2,$$

$$\overline{x}_{32}=\frac{2+1}{2}=1.5.$$

Проиллюстрируем полученное на рисунке 10: красным изображены точки-центроиды, исходные данные раскрашены в разные цвета в зависимости от изначальной инициализации.

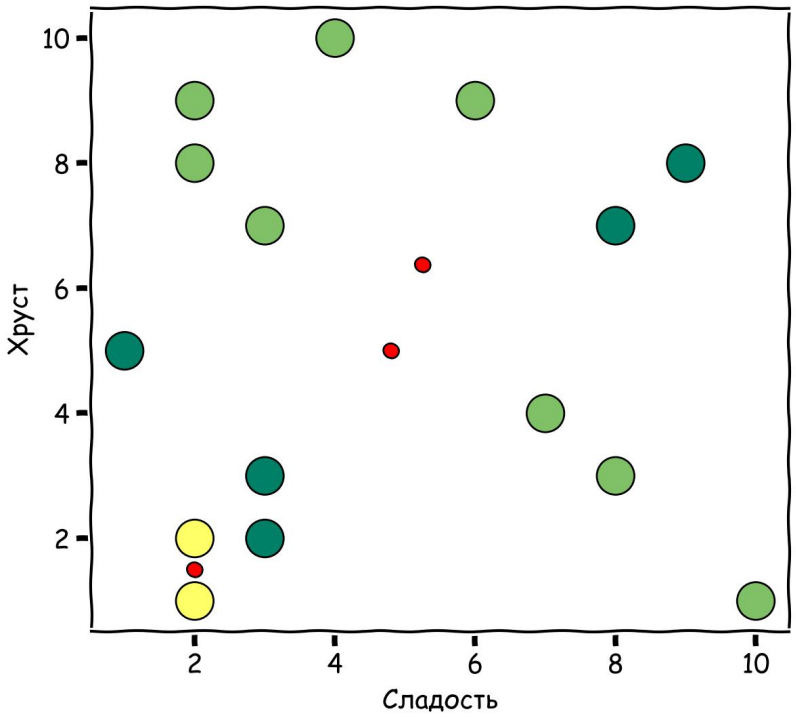

Рис. 10: Инициализация и нахождение центроидов

Двигаемся дальше. Теперь, согласно алгоритму, необходимо вычислить квадраты расстояний от каждой точки до всех трех центроидов и поместить объект в тот кластер, квадрат расстояния до центроида которого окажется наименьшим. Точка под кодовым словом «банан» является первой записью в таблице ($i = 1$) и имеет координаты $(10, 1)$. Находим квадраты расстояний до центроидов $(4.8, 5), (5.25, 6.375), (2, 1.5)$:

$$d_E^2(x_1, \overline{x}_1) = (10 - 4.8)^2 + (1 - 5)^2 = 43.04,$$
$$d_E^2(x_1, \overline{x}_2) = (10 - 5.25)^2 + (1 - 6.375)^2 = 51.453125,$$
$$d_E^2(x_1, \overline{x}_3) = (10 - 2)^2 + (1 - 1.5)^2 = 64.25.$$

Наименьшим получился квадрат расстояния до центроида кластера 1, тем самым объект «банан» теперь принадлежит кластеру 1. Аналогичную процедуру проделываем со всеми остальными объектами. Заполним таблицу новыми значениями соответствующих кластеров.

Продукт	**Сладость**	**Хруст**	**Кластер**
банан	10	1	1
апельсин	7	4	1
виноград	8	3	1
креветка	2	2	3
бекон	1	5	3
орехи	3	3	3
сыр	2	1	3
рыба	3	2	3
огурец	2	8	2
яблоко	9	8	1
морковь	4	10	2
сельдерей	2	9	2
салат	3	7	2
груша	8	7	1
перец	6	9	2

Далее повторяем шаг 3 и находим новые центроиды полученных кластеров.

$$\overline{x}_{11} = \frac{10+7+8+9+8}{5} = 8.4,$$
$$\overline{x}_{12} = \frac{1+4+3+8+7}{5} = 4.6,$$
$$\overline{x}_{21} = \frac{2+4+2+3+6}{5} = 3.4,$$
$$\overline{x}_{22} = \frac{8+10+9+7+9}{5} = 8.6,$$
$$\overline{x}_{31} = \frac{2+1+3+2+3}{5} = 2.2,$$
$$\overline{x}_{32} = \frac{2+5+3+1+2}{5} = 2.6.$$

Повторяя шаг 4, находим квадрат расстояния от каждого объекта до новых центроидов и назначаем объекту тот кластер, квадрат расстояния до центроида которого оказался наименьшим. Выполняя расчеты, можно заметить, что ни один объект не изменяет своей принадлежности кластеру, а значит кластеризация завершена. Таким образом, объекты распределены по трем кластерам, в каждом из которых объекты обладают схожими признаками. Ну а так как мы заведомо знаем, что это были за объекты, можно сказать, что разбиение получилось верным, объекты распределились правильно (рисунок 11).

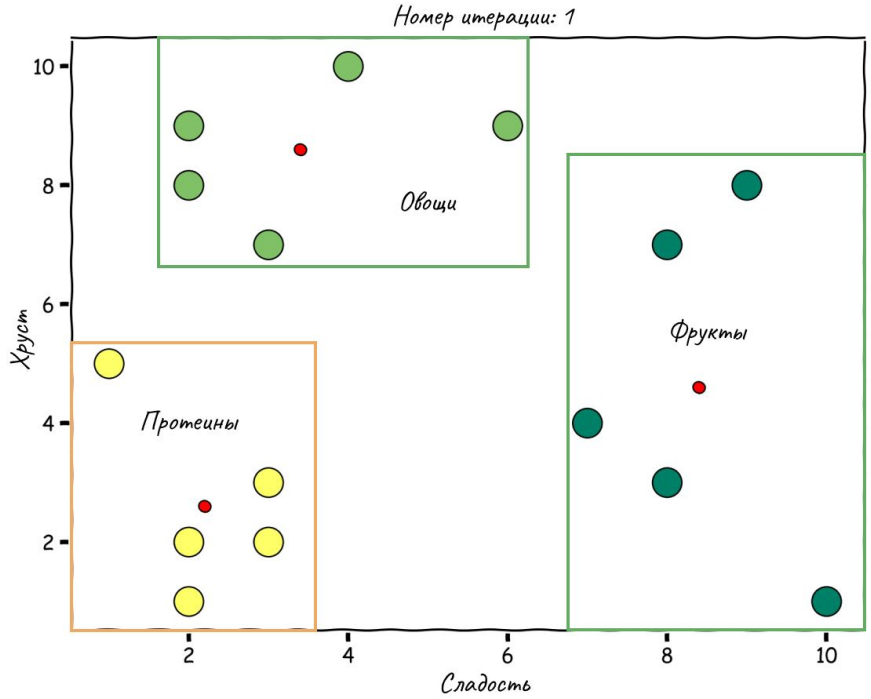

Рис. 11: Результаты кластеризации

2.4 Различные способы начальной инициализации. Сходимость метода

Инициализация методами K-means++ и наибольшим удалением

Итак, как мы уже отмечали ранее, результат кластеризации методом K-средних, вообще говоря, зависит от начальной инициализации, то есть от начального распределения рассматриваемых данных на кластеры. Метод, предложенный в алгоритме, интуитивно понятен и базируется вот на каких соображениях: если наши кластеры – плотные, хорошо разделенные шаровые сгустки, имеющие примерно одинаковый объем, то после случайной инициализации центроиды, будучи центрами масс, окажутся расположенными где-то посередине между «облаков данных», см. рисунок 12. Тогда каждое облако, конечно, чуть ли не сразу будет отнесено к ближайшему (конкретному) центроиду, и, видимо, буквально за одну итерацию алгоритм произведет безошибочную кластеризацию. Конечно, такая стерильная ситуация бывает не всегда.

Посмотрите на рисунок 13. Прекрасно видно, что теперь один кластер имеет существенно больший объем, чем два других. При случайной начальной инициализации на три кластера, все три центроида окажутся притянутыми к верхнему облаку. Совершенно ясно, что теперь два верхних центроида будут пытаться разбить большое облаков на две части, а третий центроид встанет где-то посередине между двумя маленькими облаками, объединив их

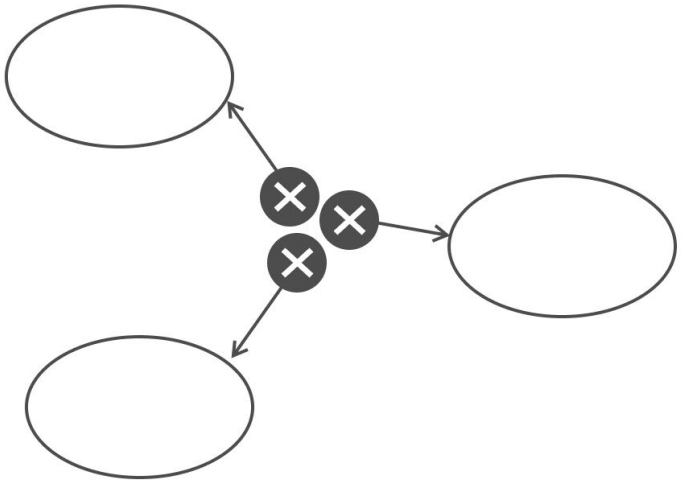

Рис. 12: Изначальный вариант инициализации – хороший вариант

в один кластер. Такая ситуация, конечно, крайне нежелательна. Обратите внимание, не помогло даже то, что мы изначально правильно определили количество кластеров!

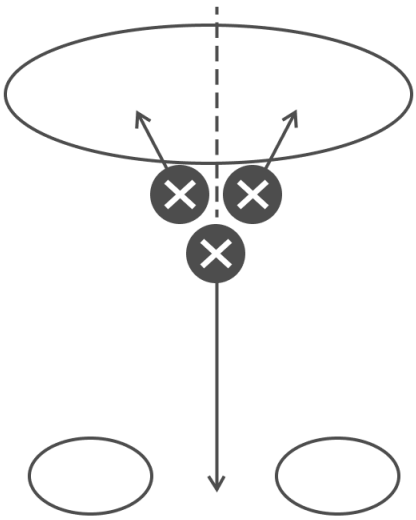

Рис. 13: Изначальный вариант инициализации – плохой вариант

Для исправления описанной ситуации, мы предложим два других (не единственных, но наиболее популярных) эмпирических подхода к начальной инициализации. Оба подхода реализованы в большинстве современных инструментов. При использовании описываемых способов, шаги 2 и 3 приведенного ранее алгоритма меняются на один из двух алгоритмов, описанных ниже.

1. **Инициализация наибольшим удалением**. Данный подход предполагает первоначальный выбор K центроидов среди набора исходных данных по следующему алгоритму:

- В качестве первых двух центроидов берутся те два объекта исходных данных, которые расположены на наибольшем расстоянии друг от друга. Если таких пар несколько, выбирается любая.
- Каждый следующий центроид (до K-ого включительно) выбирается таким образом, что расстояние от него до ближайшего из выбранных ранее центроидов максимально. Если претендентов несколько, выбирается любой из них.

Понятно, что описанный метод пытается расположить центроиды как можно дальше друг от друга и «разбросать» их по пространству признаков $\mathbb{R}^p$ так, чтобы избежать описанных ранее склеек, ненужных объединений или, наоборот, разделений кластеров.

2. **K-means++**. Данный подход очень похож на предыдущий и нацелен на внесение некоторой вероятностной подоплеки в описанный детерминизм. Алгоритм первоначального выбора K центроидов таков:

   - Случайно выбранный объект исходного набора данных назначается первым центроидом.
   - Пусть выбрано $1 \leq m \leq K-1$ центроидов среди n исходных данных. Перенумеруем оставшиеся данные, обозначив их $x_1, ..., x_{n-m}$, и вычислим расстояние d_i от каждого из объектов до ближайшего центроида, $i \in \{1, 2, ..., n-m\}$. Следующий центроид выбирается вероятностным образом, причем вероятность выбрать в качестве центроида объект x_i равна

$$\mathsf{P}((m+1)\text{-ый центроид – это } x_i) = \frac{d_i}{d_1 + d_2 + ... + d_{n-m}}.$$

   Отметим, что в качестве расстояния может быть выбрано произвольное расстояние или произвольная функция сходства, которая интересна исследователю. Например, часто в качестве функции расстояния выбирают квадрат евклидова расстояния.

Сходимость метода K-средних

Ранее мы уже говорили о том, что метод K-средних сходится, то есть существует шаг, начиная с которого объекты перестают менять свою кластерную принадлежность. В этом пункте мы обоснуем это формально.

Замечание 2.4.1 *Напомним, что если $x = (x_1, x_2, ..., x_p)$, то норма, согла-*

сованная с евклидовым расстоянием, вводится следующим образом:

$$\|x\| = \sqrt{\sum_{i=1}^{p} x_i^2}.$$

Если $x' = (x'_1, x'_2, ..., x'_p)$, *то*

$$d_E(x, x') = \sqrt{\sum_{i=1}^{p} (x_i - x'_i)^2} = \|x - x'\|$$

и, соответсвенно,

$$d_E^2(x, x') = \sum_{i=1}^{p} (x_i - x'_i)^2 = \|x - x'\|^2.$$

Сначала докажем следующую лемму.

Лемма 2.4.1 *Пусть* $x_1, ..., x_n \in \mathbb{R}^p$. *Тогда*

$$\operatorname*{Arg\,min}_{z \in \mathbb{R}^p} \sum_{i=1}^{n} \|x_i - z\|^2 = \overline{x}, \quad \text{где} \quad \overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

Иными словами, лемма утверждает, что объект, до которого сумма квадратов расстояний от всех объектов кластера минимальна – это центроид соответсвующего кластера.

Доказательство. Рассмотрим цепочку преобразований

$$\operatorname*{Arg\,min}_{z \in \mathbb{R}^p} \sum_{i=1}^{n} \|x_i - z\|^2 = \operatorname*{Arg\,min}_{z \in \mathbb{R}^p} \sum_{i=1}^{n} (x_i - z, x_i - z) =$$

$$= \operatorname*{Arg\,min}_{z \in \mathbb{R}^p} \sum_{i=1}^{n} \left(\|x_i\|^2 - 2(x_i, z) + \|z\|^2 \right) = \operatorname*{Arg\,min}_{z \in \mathbb{R}^p} \sum_{i=1}^{n} \left(\|z\|^2 - 2(x_i, z) \right) =$$

$$= \operatorname*{Arg\,min}_{z \in \mathbb{R}^p} \left(n\|z\|^2 - 2 \left(\sum_{i=1}^{n} x_i, z \right) \right) = \operatorname*{Arg\,min}_{z \in \mathbb{R}^p} n \left(\|z\|^2 - 2 \left(\frac{1}{n} \sum_{i=1}^{n} x_i, z \right) \right) =$$

$$= \operatorname*{Arg\,min}_{z \in \mathbb{R}^p} \left(\|z\|^2 - 2 \left(\overline{x}, z \right) \right) = \operatorname*{Arg\,min}_{z \in \mathbb{R}^p} \left(\|z\|^2 - 2 \left(\overline{x}, z \right) + \|\overline{x}\|^2 \right) =$$

$$= \operatorname*{Arg\,min}_{z \in \mathbb{R}^p} \|z - \overline{x}\|^2 \quad \Rightarrow \quad z = \overline{x}.$$

Более того, выкладки показывают, что решение задачи единственно. □

Теперь обоснуем сходимость описанного ранее алгоритма. Используя введенные обозначения, вспомним, что мы минимизируем функцию

$$Q = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \overline{x}_k\|^2.$$

Пусть по окончании итерации p мы имеем набор кластеров $C_1^p, ..., C_K^p$ и соответствующих им центроидов $\overline{x}_1^p, ..., \overline{x}_K^p$. Тогда, на итерации $p+1$, если существуют объекты, для которых можно найти центроид, квадрат расстояния до которого будет меньше, чем квадрат расстояния до центроида текущего кластера, то, изменив принадлежность этих объектов и сформировав новые кластеры, имеем

$$Q_p = \sum_{k=1}^{K} \sum_{x_i \in C_k^p} \|x_i - \overline{x}_k^p\|^2 > \sum_{k=1}^{K} \sum_{x_i \in C_k^{p+1}} \|x_i - \overline{x}_k^p\|^2,$$

иначе алгоритм завершен. Пересчитав центроиды после перераспределения, используя доказанную лемму, получим следующее неравенство

$$Q_p > \sum_{k=1}^{K} \sum_{x_i \in C_k^{p+1}} \|x_i - \overline{x}_k^p\|^2 \geq \sum_{k=1}^{K} \sum_{x_i \in C_k^{p+1}} \|x_i - \overline{x}_k^{p+1}\|^2 = Q_{p+1},$$

откуда $Q_p > Q_{p+1}$. Итого, алгоритм на каждом шаге уменьшает рассматриваемую функцию Q. Так как всего возможно не более K^n различных разбиений на кластеры, а центроид по кластеру строится единственным образом, то алгоритм конечен.

2.5 Выбор числа K. Каменистая осыпь

Изучив как алгоритм, так и его применение достаточно подробно, вопрос изначального определения числа кластеров пока что так и остается открытым. Ясно, что гарантированно верного ответа о правильном количестве кластеров дать нельзя, но можно попробовать отследить «степень качества» кластеризации, используя так называемую «каменистую осыпь» или «локоть».

При кластеризации исходного набора данных на K кластеров, мы старались уменьшать значение выражения

$$\sum_{k=1}^{K} \sum_{x_i \in C_k} d_E^2(x_i, \overline{x}_k),$$

численно равного сумме квадратов расстояний от объектов кластера до центроида этого кластера по всем кластерам. В результате кластеризации (после

окончания работы алгоритма), мы можем вычислить значение $Q_{final}(K)$ – сумму квадратов расстояний от объектов кластера до центроида этого кластера по всем кластерам после того, как объекты перестают менять свою кластерную принадлежность (то есть после того, как алгоритм сошелся). Понятно, что при увеличении числа кластеров, значение $Q_{final}(K)$ будет уменьшаться.

Определение 2.5.1 *Каменистой осыпью или локтем называется график зависимости значения выражения $Q_{final}(K)$ от количества кластеров K.*

На рисунке 14 представлен пример такого графика. По оси абсцисс отложено количество кластеров, по оси ординат – значение функции $Q_{final}(K)$ в зависимости от K. Легко видеть, что значения $Q_{final}(K)$ перестают резко изменяться, начиная с $K = 3$, или $K = 4$, или $K = 5$. Скорее всего, именно эти значения и имеет смысл проверить на «истинность»: осуществить кластеризацию, проинтерпретировать кластеры и посмотреть, какая из интерпретаций оказывается наиболее адекватной с точки зрения как выводов, так и предметной области.

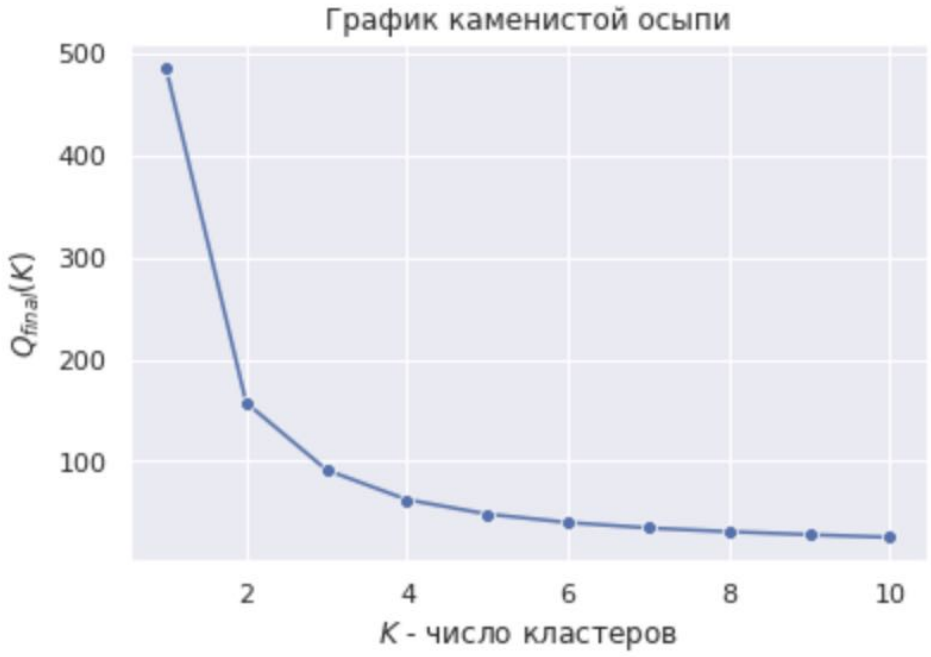

Рис. 14: График каменистой осыпи

Почему так, спросите вы? Потому что при $K > 5$, значение $Q_{final}(K)$ перестает меняться резко, а значит, скорее всего, основные кластеры выбраны, точки разделились на сгустки, а дальнейшее деление (то есть увеличение K) начинает разделять на части уже обозначенные сгустки, что, скорее всего, излишне и неправильно.

Описанный график называют каменистой осыпью из достаточно житейских соображений. При малых K мы видим резкое изменение значения $Q_{final}(K)$ – так называемый обрыв. Дальнейшее же изменение очень медленно и плавно и практически вырождается в «легкое затухание» – так называемый пляж. Вот вам и каменистая осыпь. Наверное, каждый из вас может провести похожую аналогию и с названием «локоть».

3 Агломеративная кластеризация

3.1 Небольшая мотивировка

Итак, рассмотренный ранее метод K-средних, несомненно, имеет свои преимущества – это и наглядность, и интерпретируемость, и простота реализации. В то же время, он обладает и существенными недостатками. Во-первых, метод K-средних предполагает, что кластеры – это плотные шаровые сгустки, а во-вторых он требует задания количества кластеров еще до начала исследования, что, конечно, сравнимо с подходом «ткнуть пальцем в небо». Конечно, у нас есть способ более интеллектуальной настройки алгоритма – так называемая каменистая осыпь, но и она не всегда является панацеей. Кроме того, из-за вычислительной сложности алгоритма, кластеризация при разных значениях K может длиться очень и очень долго.

В этом разделе мы рассмотрим еще один метод кластеризации – так называемую агломеративную или, как еще ее называют, иерархическую кластеризацию. Данный метод, во-первых, вычислительно более прост, во-вторых, не требует первоначального задания числа кластеров, в-третьих, позволяет удобным образом визуализировать результаты кластеризации и в-четвертых (но далеко не в последних), умет находить ленточные кластеры.

3.2 Описание алгоритма. Способы измерения расстояния между кластерами

Сразу приведем формальное описание алгоритма, делая ставку, скорее, на интуицию; технические детали поясним несколько позже. Пусть имеется набор данных $X = (x_1, x_2, ..., x_n)$ объема n.

1. Выбирается функция расстояния $\rho(X, Y)$, отражающая «похожесть» (или «близость») кластеров X и Y.

2. Каждый объект x_i, $i \in \{1, 2, ..., n\}$, помещается в отдельный кластер:

$$C_1 = \{x_1\}, \quad C_2 = \{x_2\}, \quad ...., \quad C_n = \{x_n\}.$$

3. Пусть имеется $K > 1$ кластеров. Ищутся два наиболее «похожих» относительно выбранной функции расстояния ρ кластера и объединяются между собой, остальные кластеры остаются неизменными. В случае, если кандидатов (пар) на объединение несколько, объединяется любая пара кандидатов. В итоге остается $K - 1$ кластер.

4. Шаг 3 повторяется, пока количество кластеров не станет равным 1.

Все шаги алгоритма должны быть интуитивно понятны, их легко изобразить на рисунке, см. рисунок 15. У нас есть 5 объектов, каждый из которых на начальном этапе является отдельным кластером, итого на старте (после шага 2) мы имеем 5 кластеров. Ясно, что ближайшими кластерами, с точки зрения человеческого восприятия, являются кластеры, содержащие крестик и треугольник, значит они и будут объединены на первом шаге (кластеры, получившиеся после первой итерации объединения, обведены оранжевым). Итак, осталось 4 кластера. Двигаемся дальше. Теперь ближе всего расположены кластеры, содержащие квадрат и ромб, объединяем их. Кластеры по окончании этой итерации обведены красным. Осталось три кластера. Теперь, вроде как, ближе всего находятся кластеры, один из которых содержит крестик и треугольник, а второй – круг, объединим их. Результат обведен синим. Осталось два кластера – объединяем их и... алгоритм завершен. Не останавливаясь на данный момент на вопросе интерпретации полученных кластеров, обсудим немаловажный технический вопрос: а что значит «ближайший» кластер или, что то же самое, а откуда берется упомянутая в алгоритме (и, в некотором смысле, ключевая) функция ρ?

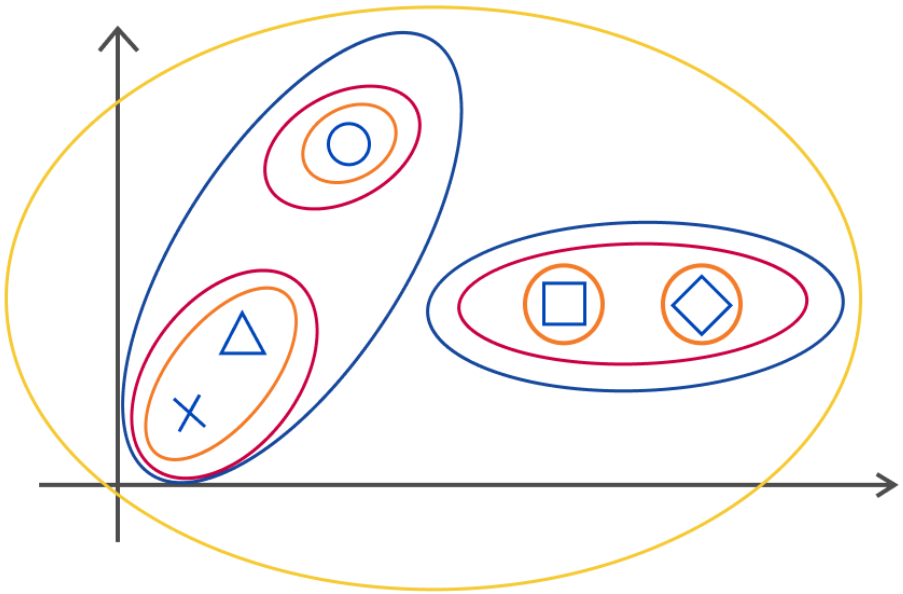

Рис. 15: Визуализация алгоритма

Пусть кластер X состоит из элементов $x_1, x_2, \ldots, x_s$, а кластер X' – из элементов $x'_1, x'_2, \ldots, x'_r$, причем все элементы кластеров принадлежат пространству $\mathbb{R}^p$, то есть описываются p числовыми признаками-предикторами. Пусть также d – некоторая функция расстояния, показывающая схожесть объектов из $\mathbb{R}^p$. Покажем несколько основных приемов определения расстояния между кластерами X и X'.

1. **Метод полной связи (метод дальнего соседа).** В качестве расстояния $\rho(X, X')$ между кластерами X и X' принимается максимальное расстояние между элементами соответствующих кластеров, то есть:

$$\rho(X, X') = \max_{x \in X,\ x' \in X'} d(x, x'),$$

см. рисунок 16. Данное расстояние достаточно хорошо выделяет плотные шаровые сгустки.

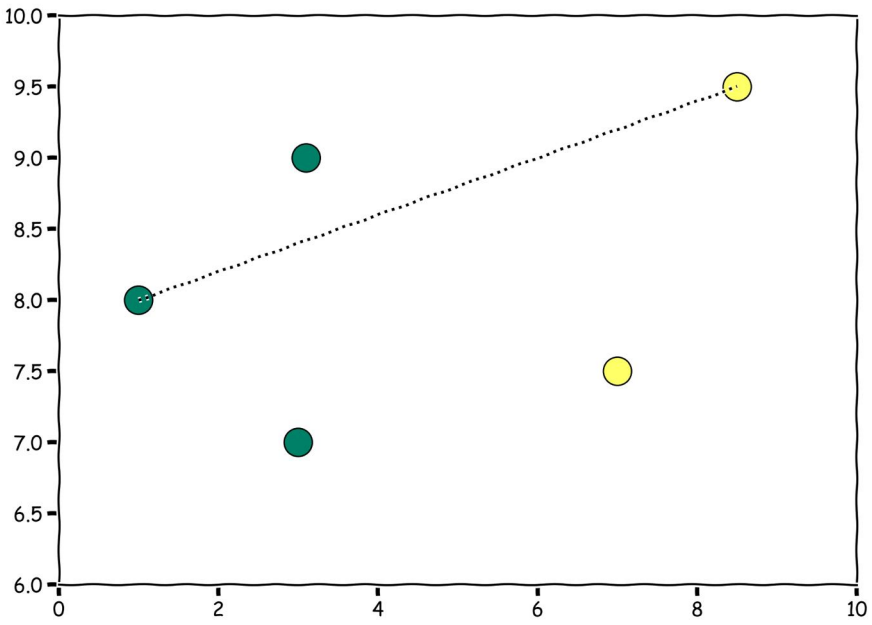

Рис. 16: Метод полной связи

2. **Метод одиночной связи (метод ближайшего соседа).** В качестве расстояния $\rho(X, X')$ между кластерами X и X' принимается минимальное расстояние между элементами соответствующих кластеров, то есть:

$$\rho(X, X') = \min_{x \in X,\ x' \in X'} d(x, x'),$$

см. рисунок 17. Данное расстояние отлично выделяет ленточные кластеры.

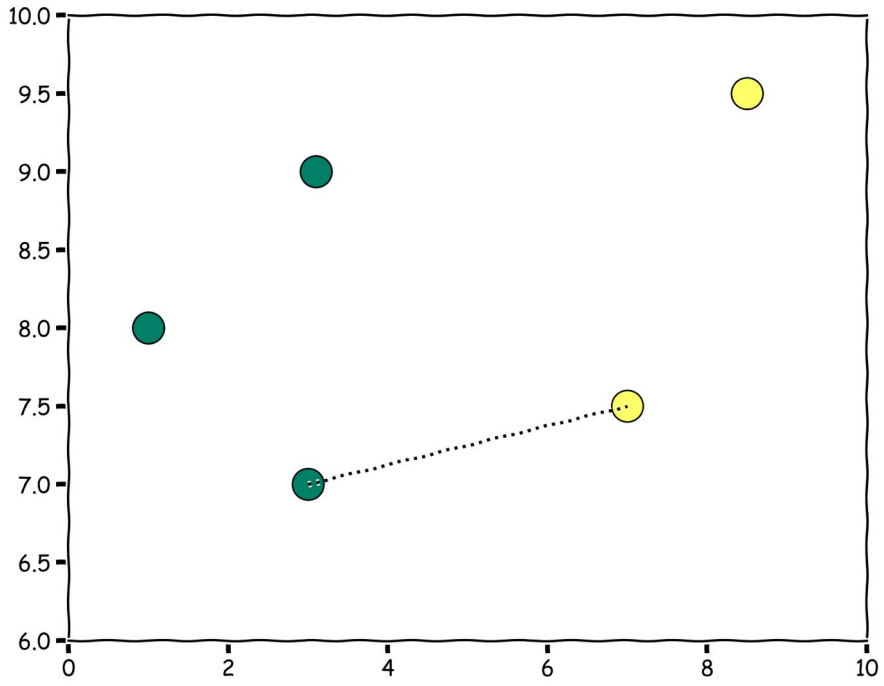

Рис. 17: Метод одиночной связи

3. **Метод средней связи (среднее невзвешенное расстояние).** В качестве расстояния $\rho(X, X')$ между кластерами X и X' принимается среднее арифметическое всех попарных расстояний между элементами соответствующих кластеров, то есть:

$$\rho(X, X') = \frac{1}{|X| \cdot |X'|} \sum_{x \in X} \sum_{x' \in X'} d(x, x'),$$

где $|X|$ и $|X'|$ – количество элементов в кластерах X и X', соответственно, см. рисунок 18. Данное расстояние хорошо справляется с выделением плотных шаровых сгустков.

4. **Центроидный метод (центроидальный метод).** В качестве расстояния $\rho(X, X')$ между кластерами X и X' принимается расстояние между их центроидами, то есть

$$\rho(X, X') = d(\overline{x}, \overline{x}'),$$

где $\overline{x}$ и $\overline{x}'$ – центроиды кластеров X и X', соответственно, см. рисунок 19. Данный метод тоже справляется с плотными шаровыми сгустками, однако применяется достаточно редко из-за возможных самопересечений в так называемой дендрограмме при визуализации результатов кластеризации; мы обратим на это внимание чуть позже.

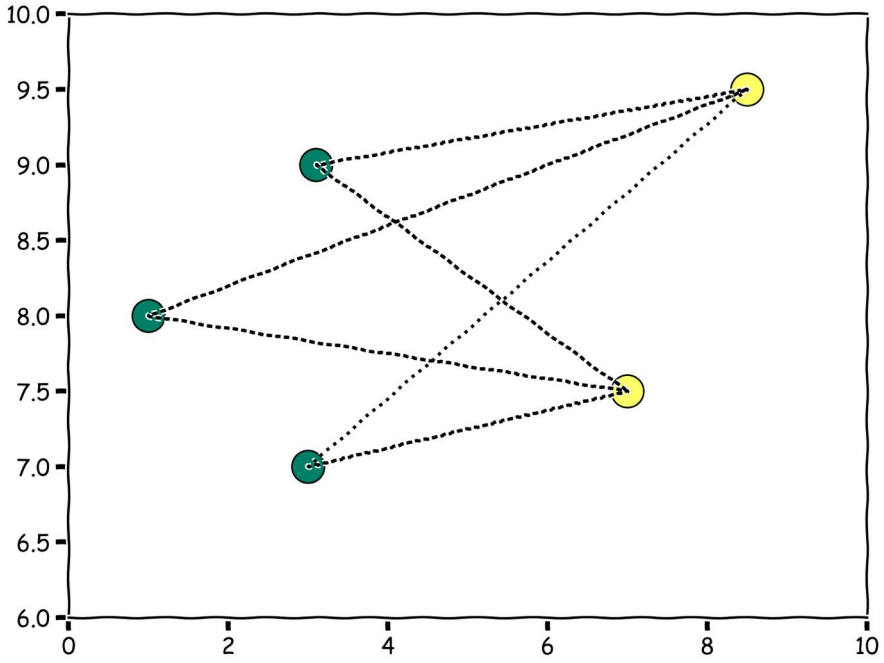

Рис. 18: Метод средней связи

5. **Метод Варда.** В качестве расстояния $\rho(X, X')$ между кластерами X и X' принимается величина, показывающая изменение суммы квадратов расстояний от элементов кластера до центроида до и после объединения, то есть:

$$\rho(X, X') = \sum_{x \in X \cup X'} d_E^2(x, \overline{x}_\cup) - \sum_{x \in X} d_E^2(x, \overline{x}) - \sum_{x \in X'} d_E^2(x', \overline{x}') =$$

$$= \frac{|X||X'|}{|X| + |X'|} d_E^2(\overline{x}, \overline{x}'),$$

где $\overline{x}$, $\overline{x}'$ и $\overline{x}_\cup$ – центроиды кластеров X, X' и $X \cup X'$, соответственно, а $|X|$ и $|X'|$ – количество элементов в кластерах X и X', соответственно, d_E – евклидово расстояние.

Видно, что первое слагаемое в последней формуле отвечает за сумму квадратов расстояний от объектов объединенного кластера до центроида этого кластера, а второе и третье – за сумму квадратов расстояний от объектов кластера X и X' до центроидов кластеров X и X', соответсвенно. В итоге, в методе Варда расстояние между кластерами показывает что-то вроде изменения «плотности» кластера при объединении двух кластеров; нацелено оно, скорее, на выделение мелких кластеров.

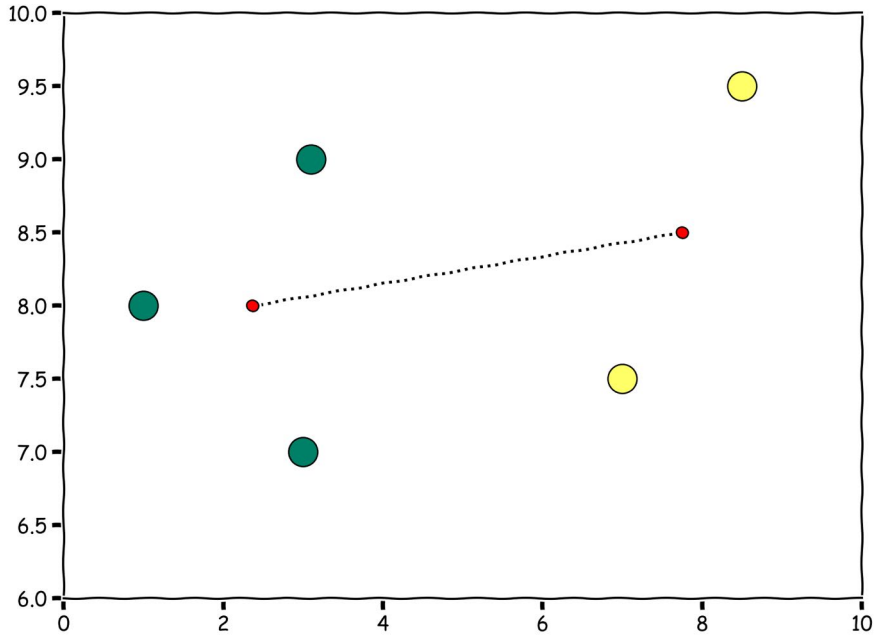

Рис. 19: Центроидный метод

Заметим и следующее. Расстояние, используемое в методе Варда, имеет отношение к статистике, к дисперсионному анализу. Оно оценивает изменение разброса в данных после объединения кластеров; в частности поэтому оно достаточно популярно в приложениях.

Замечание 3.2.1 *Отметим, что в приведенном нами аналитическом выражении для расстояния $\rho(X, X')$ в методе Варда используется именно евклидово расстояние d_E. В инструментах часто бывает возможным использовать и какую-то другую функцию d; в этом случае последнее равенство, вообще говоря, неверно.*

Также отметим, что при использовании метода Варда, некоторые инструменты вычисляют расстояние между кластерами следующим образом:

$$\sqrt{\sum_{x\in X\cup X'} d_E^2(x,\overline{x}_\cup) - \sum_{x\in X} d_E^2(x,\overline{x}) - \sum_{x\in X'} d_E^2(x',\overline{x}')}.$$

Это – не что иное, как корень из введенного нами расстояния. Более того, в некоторых инструментах написанное выражение умножается на положительную константу. Не вдаваясь в детали озвученных отличий скажем, что подобные изменения, по большому счету, не сказываются на результате кластеризации, ведь по сути к введенной ранее функции $\rho(X, X')$ просто-напросто применяется некоторое монотонное преобразование, мас-

штабирующее расстояния.

Замечание 3.2.2 *Понятно, что так как понятие близости кластеров в иерархической кластеризации основано на понятии расстояния, то признаки кластеризуемых объектов перед началом кластеризации имеет смысл либо стандартизировать, либо нормировать.*

В заключение данного пункта, приведем формальное обоснование написанной в пункте 5 формулы.

Лемма 3.2.1 *Пусть X и X' – два кластера объектов из $\mathbb{R}^p$, d_E – евклидово расстояние в $\mathbb{R}^p$, тогда*

$$\sum_{x\in X\cup X'} d_E^2(x,\overline{x}_\cup) - \sum_{x\in X} d_E^2(x,\overline{x}) - \sum_{x\in X'} d_E^2(x',\overline{x}') = \frac{|X||X'|}{|X|+|X'|} d_E^2(\overline{x},\overline{x}'),$$

где $\overline{x}$, $\overline{x}'$ и $\overline{x}_\cup$ – центроиды кластеров X, X' и $X\cup X'$, соответственно, а $|X|$ и $|X'|$ – количество элементов в кластерах X и X', соответственно.

Доказательство. Во-первых заметим, что в силу того, что кластеры не пересекаются, справедливо соотношение

$$\overline{x}_\cup = \frac{1}{|X|+|X'|}\sum_{x\in X\cup X'} x = \frac{|X|}{|X|+|X'|}\overline{x} + \frac{|X'|}{|X|+|X'|}\overline{x}'.$$

Так как

$$d_E^2(x,x') = \|x-x'\|^2,$$

то

$$\sum_{x\in X} d_E^2(x,\overline{x}_\cup) = \sum_{x\in X}\|x-x_\cup\|^2 = \sum_{x\in X}\left\|x-\overline{x}-\frac{|X'|}{|X|+|X'|}(\overline{x}'-\overline{x})\right\|^2 =$$

$$= \sum_{x\in X}\|x-\overline{x}\|^2 - 2\frac{|X'|}{|X|+|X'|}\sum_{x\in X}(x-\overline{x},\overline{x}'-\overline{x}) + |X|\left(\frac{|X'|}{|X|+|X'|}\right)^2\|\overline{x}-\overline{x}'\|^2.$$

Ясно, что среднее слагаемое равно нулю. Симметрично,

$$\sum_{x\in X'} d_E^2(x,\overline{x}_\cup) = \sum_{x\in X'}\|x-\overline{x}'\|^2 + |X'|\left(\frac{|X|}{|X|+|X'|}\right)^2\|\overline{x}-\overline{x}'\|^2.$$

Складывая равенства, получим

$$\sum_{x\in X\cup X'} d_E^2(x,\overline{x}_\cup) = \sum_{x\in X} d_E^2(x,\overline{x}) + \sum_{x\in X'} d_E^2(x,\overline{x}') + \frac{|X||X'|}{|X|+|X'|}\|\overline{x}-\overline{x}'\|^2.$$

□

Теперь рассмотрим пример применения написанного алгоритма, а также научимся визуализировать кластеризацию – строить и интерпретировать так называемую дендрограмму.

3.3 Пример: хруст и сладость продуктов. Дендрограмма

Пример кластеризации

Рассмотрим процедуру кластеризиции и построения дендрограммы на уже знакомом нам примере с хрустом и сладостью пищи. Исходные данные приведены в следующей таблице:

Номер	**Продукт**	**Сладость**	**Хруст**
1	банан	10	1
2	апельсин	7	4
3	виноград	8	3
4	креветка	2	2
5	бекон	1	5
6	орехи	3	3
7	сыр	2	1
8	рыба	3	2
9	огурец	2	8
10	яблоко	9	8
11	морковь	4	10
12	сельдерей	2	9
13	салат	3	7
14	груша	8	7
15	перец	6	9

Визуализируем наши данные. Для удобства восприятия, вместо изображения точек на плоскости будем прописывать соответствующие номера объектов из таблицы исходных данных, рисунок 20. Например, объект «банан» имеет номер 1 и координаты $(10, 1)$, объект «апельсин» – номер 2 и координаты $(7, 4)$, и так далее.

Сначала определимся с методами нахождения расстояния как между кластерами, так и между объектами. В рассматриваемом примере в качестве первого мы будем использовать метод полной связи, а второго – евклидово расстояние:

$$\rho(X, X') = \max_{x \in X,\ x' \in X'} d_E(x, x'), \quad d_E(x, x') = \sqrt{\sum_{j=1}^{p} (x_j - x'_j)^2},$$

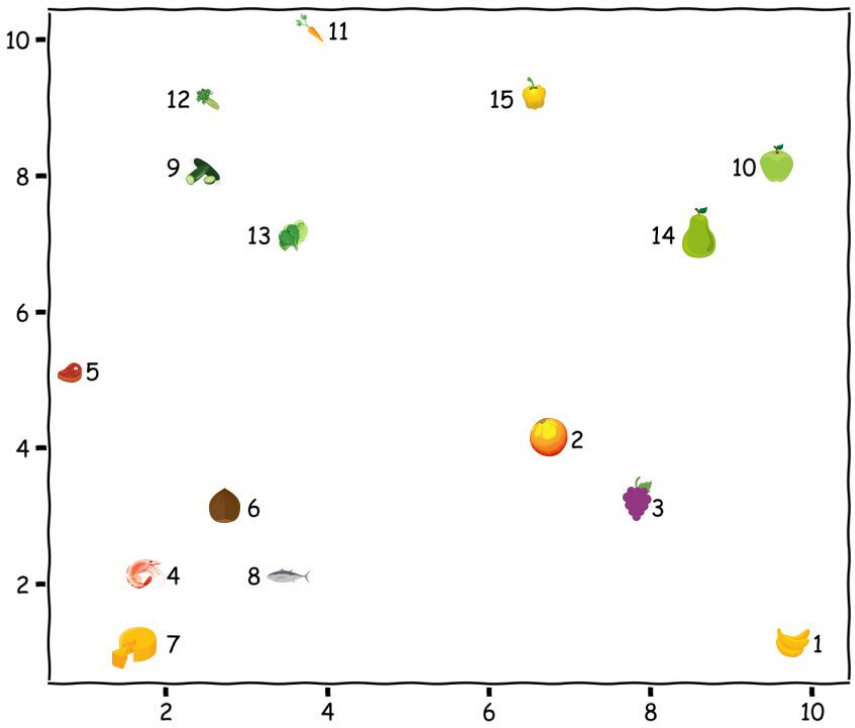

Рис. 20: Исходные данные по номерам объектов

$$x = (x_1, x_2, \ldots, x_p), \quad x' = (x'_1, x'_2, \ldots, x'_p).$$

В самом начале, согласно алгоритму, каждый объект представляет собой отдельный кластер, а значит у нас в распоряжении 15 кластеров. Вычислим все попарные расстояния между кластерами и выберем два наиболее близких. Для удобства поиска расстояния между кластерами составим матрицу расстояний. В качестве элементов матрицы выступают расстояния между рассматриваемыми объектами. На рисунке 21 приведены округленные результаты с точностью до 1 знака после запятой; при настоящих расчетах округления лучше не использовать.

Можно заметить, что у нас получилось несколько пар (объектов, или, что на первой итерации то же самое, кластеров), расстояния между которыми равны единице:

$$d_E(4, 7) = d_E(4, 8) = d_E(6, 8) = d_E(9, 12) = 1.$$

Согласно описанному алгоритму, мы можем объединить в новый кластер любую из представленных пар. Объединим, например, «креветку» и «сыр» (объекты с номерами 4 и 7).

Теперь у нас осталось 14 кластеров, причем один из них состоит из 2-ух элементов:

$$\{4, 7\}, \{1\}, \{2\}, \{3\}, \{5\}, \{6\}, \{8\}, \{9\}, \{10\}, \{11\}, \{12\}, \{13\}, \{14\}, \{15\}.$$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	4.2	2.8	8.1	9.8	7.3	8	7.1	10.6	7.1	10.8	11.3	9.2	6.3	8.9
2	4.2	0	1.4	5.4	6.1	4.1	5.8	4.5	6.4	4.5	6.7	7.1	5	3.2	5.1
3	2.8	1.4	0	6.1	7.3	5	6.3	5.1	7.8	5.1	8.1	8.5	6.4	4	6.3
4	8.1	5.4	6.1	0	3.2	1.4	1	1	6	9.2	8.2	7	5.1	7.8	8.1
5	9.8	6.1	7.3	3.2	0	2.8	4.1	3.6	3.2	8.5	5.8	4.1	2.8	7.3	6.4
6	7.3	4.1	5	1.4	2.8	0	2.2	1	5.1	7.8	7.1	6.1	4	6.4	6.7
7	8	5.8	6.3	1	4.1	2.2	0	1.4	7	9.9	9.2	8	6.1	8.5	8.9
8	7.1	4.5	5.1	1	3.6	1	1.4	0	6.1	8.5	8.1	7.1	5	7.1	7.6
9	10.6	6.4	7.8	6	3.2	5.1	7	6.1	0	7	2.8	1	1.4	6.1	4.1
10	7.1	4.5	5.1	9.2	8.5	7.8	9.9	8.5	7	0	5.4	7.1	6.1	1.4	3.2
11	10.8	6.7	8.1	8.2	5.8	7.1	9.2	8.1	2.8	5.4	0	2.2	3.2	5	2.2
12	11.3	7.1	8.5	7	4.1	6.1	8	7.1	1	7.1	2.2	0	2.2	6.3	4
13	9.2	5	6.4	5.1	2.8	4	6.1	5	1.4	6.1	3.2	2.2	0	5	3.6
14	6.3	3.2	4	7.8	7.3	6.4	8.5	7.1	6.1	1.4	5	6.3	5	0	2.8
15	8.9	5.1	6.3	8.1	6.4	6.7	8.9	7.6	4.1	3.2	22	4	3.6	2.8	0

Рис. 21: Матрица расстояний

Снова ищем попарные расстояния между всеми кластерами и выбираем наименьшее. Для примера, найдем расстояние от кластера $\{4, 7\}$ до кластера $\{13\}$. Можно воспользоваться уже найденной матрицей расстояний. Так как

$$d_E(4, 13) \approx 5.1, \quad d_E(7, 13) \approx 6.1,$$

а в качестве расстояния между кластерами берется метод полной связи (или дальнего соседа), то

$$\rho(\{4, 13\}, \{7\}) = \max(d_E(4, 13), d_E(7, 13)) \approx 6.1.$$

Приведем список всех объединений (по шагам) в рамках рассматриваемой кластеризации:

1. $\{4\} \cup \{7\}$.
2. $\{6\} \cup \{8\}$.
3. $\{9\} \cup \{12\}$.
4. $\{2\} \cup \{3\}$.
5. $\{10\} \cup \{14\}$.

6. $\{4, 7\} \cup \{6, 8\}$.
7. $\{9, 12\} \cup \{13\}$.
8. $\{11\} \cup \{15\}$.
9. $\{4, 7, 6, 8\} \cup \{5\}$.
10. $\{9, 12, 13\} \cup \{11, 15\}$.
11. $\{1\} \cup \{2, 3\}$.
12. $\{1, 2, 3\} \cup \{10, 14\}$.
13. $\{4, 7, 6, 8, 5\} \cup \{9, 12, 13, 11, 15\}$.
14. $\{1, 2, 3, 10, 14\} \cup \{4, 7, 6, 8, 5, 9, 12, 13, 11, 15\}$.

После окончания алгоритма кластеризации, можно приступать к построению дендрограммы.

Построение дендрограммы

Дендрограмма, отвечающая проведенной кластеризации, представлена на рисунке 22. Давайте разберемся в том, как ее строить и как интерпретировать результаты. Для этого отложим по оси абсцисс номера наших объектов в соответствии с порядком, полученным на последней итерации:

$$\{1, 2, 3, 10, 14, 4, 7, 6, 8, 5, 9, 12, 13, 11, 15\}.$$

На первой итерации соединяются кластеры (проводя аналогию с деревьями – листья) $\{4\}$ и $\{7\}$ «веткой» (горизонтальным отрезком), расположенной на высоте, равной расстоянию между кластерами $\{4\}$ и $\{7\}$; так как расстояние равно 1, то горизонтальный отрезок имеет игрековую координату, равную 1, вертикальные отрезки от листьев до «ветки» проводятся лишь для удобства и наглядности.

Затем аналогичным образом соединяются «ветками» пары листьев $(6, 8), (9, 12), (2, 3), (10, 14)$ (расстояние между элементами двух последних пар равно 1.4, поэтому ветки, соединяющие элементы этих пар, выше и находятся на отметке 1.4 по оси игрек).

Дальше необходимо соединить кластеры $\{4, 7\}$ и $\{6, 8\}$. Высота «ветки» опять же равна расстоянию между кластерами $\{4, 7\}, \{6, 8\}$, то есть

$$\rho(\{4, 7\}, \{6, 8\}) = \max(d_E(4, 6), d_E(4, 8), d_E(7, 6), d_E(7, 8)) = d_E(7, 6) \approx 2.2.$$

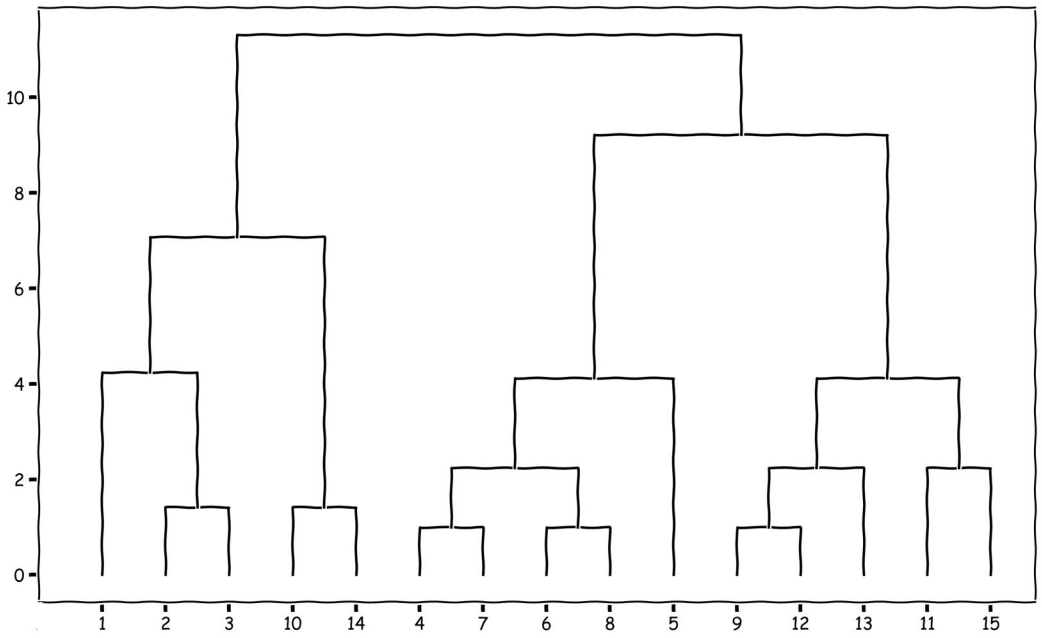

Рис. 22: Агломеративная кластеризация

Попробуйте самостоятельно найти, например, расстояние между кластерами $\{9, 12, 13\}$ и $\{11, 15\}$), используя матрицу расстояний. Должно получиться так:

$$\rho(\{9, 12, 13\}, \{11, 15\}) = d_E(9, 15) \approx 4.1.$$

Все дальнейшие действия аналогичны описанным ранее. Вся последовательность построений показана на рисунке 23.

Итак, кластеризация проведена, дендрограмма построена, но все еще остаются незатронутыми два ключевых вопроса. Первый – это вопрос о количестве кластеров, а второй – о составе получившихся кластеров. Оказывается, имея перед глазами дендрограмму, варианты ответов на первый и второй вопросы становятся очень наглядными.

Сразу скажем, что исследователь должен самостоятельно определить количество кластеров. В прочем, есть и некоторый бонус, он это может сделать уже постфактум, подбирая наиболее оптимальное и правдоподобное (об этом подробнее мы скажем в следующем пункте) число кластеров, рисунок 24. Для получения итоговых кластеров, достаточно провести горизонтальную черту на выбранной «высоте». Тогда количество пересечений с вертикальными соединениями «веток» и будет равно количеству отсеченных кластеров. Листья отсеченных кластеров – это и есть кластеризованные объекты. Если в соответствии с нашими представлениями о еде выделять 3 кластера, то результат можно наблюдать на рисунке 25. Римскими цифрами отмечены итерации, на

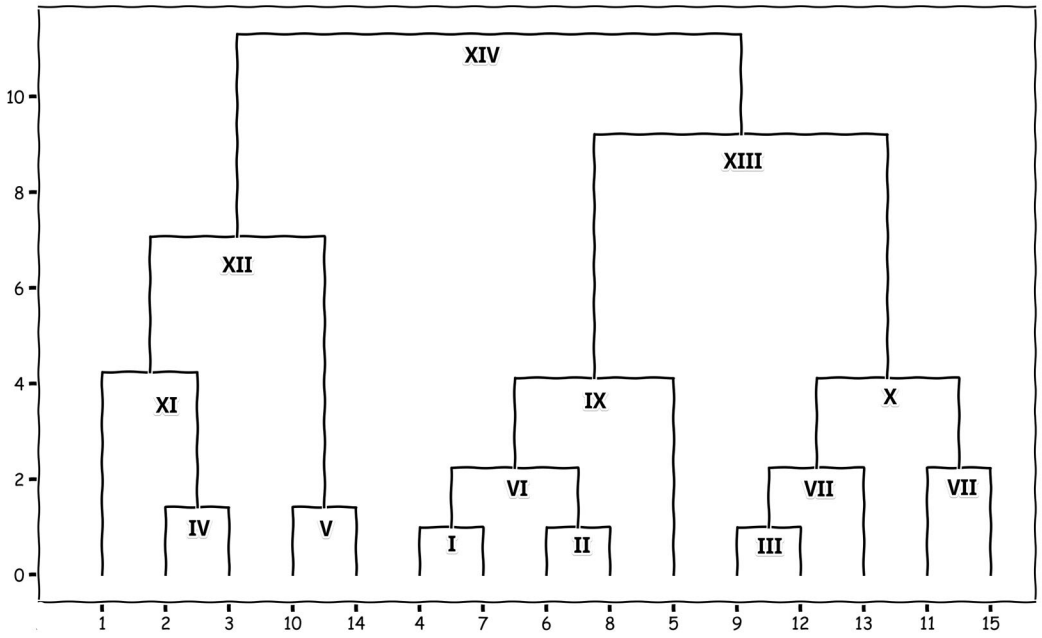

Рис. 23: Последовательность построения дендрограммы

которых произошло объединение.

3.4 Каменистая осыпь и определение числа кластеров

Как вы уже, наверное, поняли, озвученный ранее бонус иерархической кластеризации – отсутствие надобности изначального выбора количества кластеров – вовсе не бонус как таковой. Конечно, имея n кластеризуемых объектов, в результате кластеризации мы получаем n возможных вариантов разделения объектов на группы, но как понять какую из кластеризаций выбрать, в какой момент остановиться? Наверное, по опыту метода K-средних, вы и сами можете предложить практическое решение: посмотреть на каменистую осыпь. Чем же она является в случае иерархической кластеризации? А вот чем.

Определение 3.4.1 *Каменистой осыпью или локтем называется график зависимости значения расстояния $\rho(X, X')$ между объединяемыми кластерами X и X' от количества оставшихся после объединения кластеров.*

Посмотрите на рисунок 26, на нем изображена каменистая осыпь для иерархической кластеризации. Как интерпретировать данный график? Давайте разбираться. В случае, когда все объекты объединены в один кластер, согласно графику расстояние между последними объединяемыми кластерами

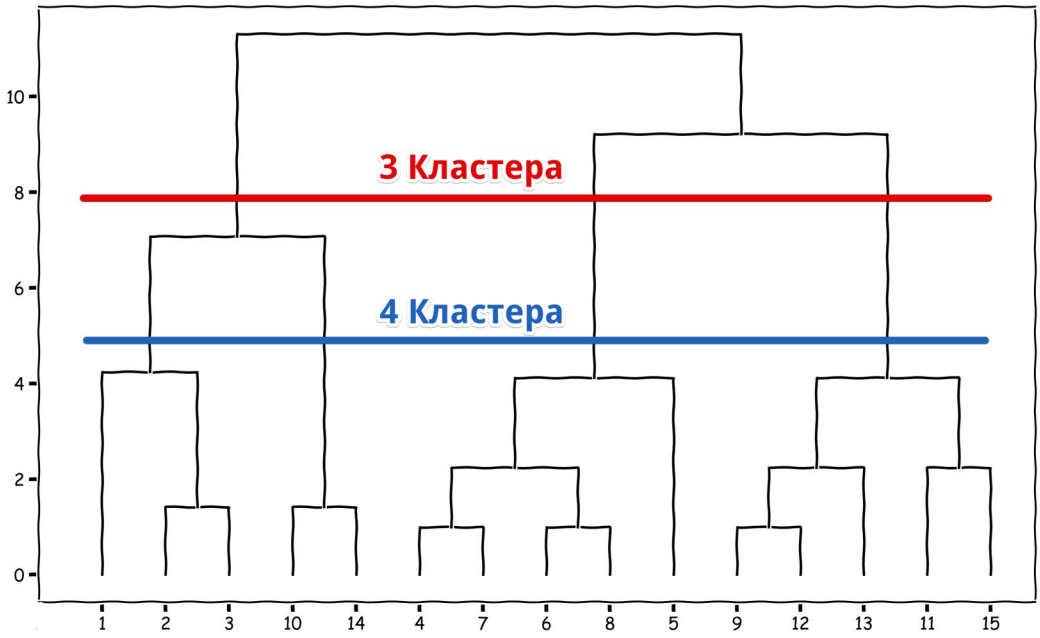

Рис. 24: Определение количества кластеров

чуть больше, чем 25. На предыдущем этапе, когда после объединения осталось два кластера, наименьшее расстояние между объединяемыми кластерами было равно примерно 10. На этапе до этого – около 9, и так далее. Как и в случае метода K-средних, по каменистой осыпи имеет смысл найти такое число кластеров, после которого изменение расстояния становится значительно меньше, чем до которого. Судя по приведенному графику, имеет смысл попробовать оставить либо 2, либо 4, либо 6 кластеров. Визуальная интерпретация с обрывом и пляжем остается неизменной – такой же, как и при обсуждении метода K-средних; более подробно на этом мы останавливаться не будем.

Отметим отдельно, что график каменистой осыпи вовсе не всегда оказывается монотонным. Рассмотрим все тот же пример с хрустом и сладостью продуктов, только теперь расстояние между кластерами будем вычислять, используя центроидальный метод; расстояние между объектами оставим прежним – евклидовым. Дендрограмма приведена на рисунке 27, а график каменистой осыпи – на рисунке 28. Видно, что на последней итерации, то есть при объединении двух кластеров в один, расстояние между объединяемыми кластерами оказалось меньше, чем на предыдущей итерации: когда три кластера обьединялись в два. Это – особенность центроидального метода измерения расстояний между кластерами; в частности из-за этого данный метод все реже используют в иерархическом кластерном анализе. Сколько же

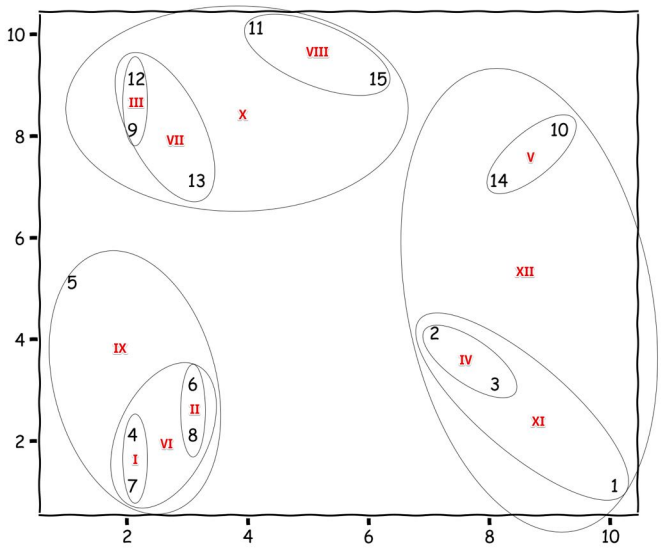

Рис. 25: Этапы объединения кластеров

кластеров выбрать в такой ситуации? Похоже, хороший вариант – это либо 4, либо 9, либо 12. Последние варианты отметем сразу – у нас все 15 объектов, кластеры будут очень малочисленными, это не представляет интерес. А что будет, если выбрать 4 кластера? Чтобы не запутаться, достаточно выбрать ρ равным, например, 4 и провести на соответствующем уровне дендрограммы прямую (в данном случае – вертикальную). Мы видим, что груша и яблоко отделились от винограда, апельсина и банана. Что же, скорее всего они просто более хрустящие, разделение достаточно разумно.

Напоследок отметим, что, выбрав в качестве расстояния между кластерами метод полной связи (как было в примере ранее), и отсекая 4 кластера, мы получили бы точно такую же кластеризацию, как и сейчас. Совпадение кластеризаций – хороший знак, указывающий скорее всего на то, что кластеры действительно есть и результаты кластеризации состоятельны.

4 DBSCAN

4.1 Небольшая мотивировка и описание

Последним методом, который мы будем рассматривать в данной лекции, является набравший заслуженную популярность метод DBSCAN (Density-based spatial clustering of applications with noise) – плотностный алгоритм пространственной кластеризации с присутствием шума. Если вдуматься в само название алгоритма, то становится понятной и идея: объекты составля-

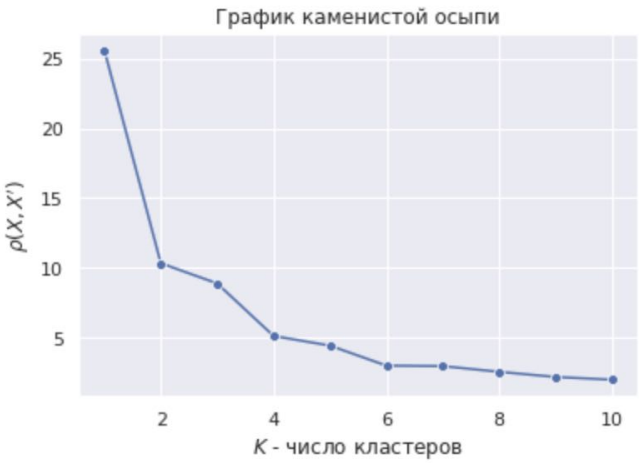

Рис. 26: Каменистая осыпь в иерархической кластеризации

ют один и тот же кластер, если все они плотно расположены друг к другу. Что значит плотно? Это значит, что рядом с каждым объектом кластера есть достаточно много соседей – элементов этого же кластера.

Оказывается, что DBSCAN с успехом справляется с нахождением как плотных шаровых сгустков, так и ленточных кластеров. Кроме того, при фиксированных параметрах модели количество кластеров определяется однозначным образом, а значит наконец-то исследователю и правда не нужно определять последнее каким-то хитрым образом. Но.. Все, конечно, далеко не так безоблачно. Давайте для начала разберемся с алгоритмом.

4.2 Основные определения и описание алгоритма

Пусть имеется набор данных $X = (x_1, x_2, ..., x_n)$ объема n, причем $x_i \in \mathbb{R}^p$, $i \in \{1, 2, ..., n\}$, то есть каждый объект описывается p числовыми признаками. Пусть d – выбранная функция расстояния в $\mathbb{R}^p$. Раз в преамбуле к этому методу речь шла о соседях, то давайте сначала разберемся с тем, каким образом отвечать на вопрос: является ли объект x' соседом для x, или нет. Для этого введем следующее определение.

Определение 4.2.1 *Пусть $\varepsilon > 0$. Множество*

$$\overline{B}(x_0, \varepsilon) = \{y \in \mathbb{R}^p : \ d(x_0, y) \leq \varepsilon\}$$

называется замкнутым (в $\mathbb{R}^p$) шаром радиуса ε с центром в точке x_0.

Так как мы кластеризуем исходный набор данных X, то нам совершенно не интересны точки из $\mathbb{R}^p$, в X не входящие, поэтому разумно ввести и следующее определение (и обозначение) для замкнутого шара в X.

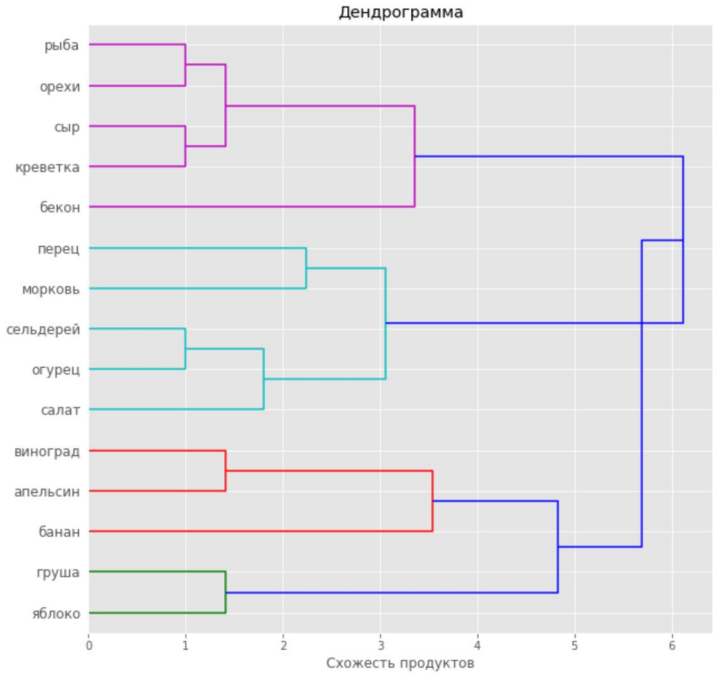

Рис. 27: Дендрограмма для примера

Определение 4.2.2 *Пусть* $\varepsilon > 0$, $x_0 \in X$. *Множество*

$$\overline{B}_X(x_0, \varepsilon) = \{y \in X : \ d(x_0, y) \leq \varepsilon\}$$

называется замкнутым шаром в X *с центром в* x_0 *и радиуса* ε.

Нас с вами будут интересовать только шары в смысле только что данного определения, поэтому мы, для краткости, будем писать $\overline{B}(x_0, \varepsilon)$, подразумевая $\overline{B}_X(x_0, \varepsilon)$. Учитывая это соглашение, а так же тот факт, что множество X конечно, оказывается разумным (и полезным) ввести следующее обозначение:

$$|\overline{B}(x_0, \varepsilon)| \text{ – количество элементов в множестве } \overline{B}_X(x_0, \varepsilon).$$

Определение 4.2.3 *Пусть* $\varepsilon > 0$. *Элементы множества* $\overline{B}(x_0, \varepsilon)$ *называются соседями элемента* x_0.

Итак, соседи элемента $x_0 \in X$ – это точки из X, лежащие в замкнутой ε-окрестности точки x_0; интуитивно понятно, не так ли?

Замечание 4.2.1 *Отметим несколько полезных замечаний. Во-первых, если выбранная функция расстояния* d *обладает свойством, что* $d(x, x) = 0$, *то множество* $\overline{B}(x_0, \varepsilon)$ *никогда не пусто, а каждый элемент является соседом сам для себя. Кроме того, если* $d(x, x') = d(x', x)$, *то если* x *– сосед для*

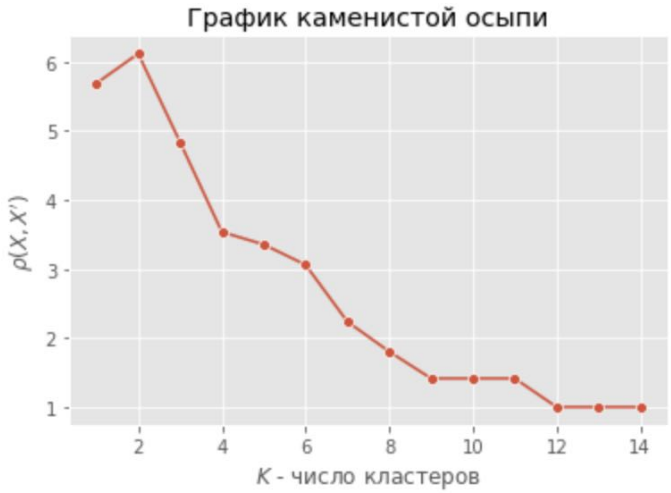

Рис. 28: Каменистая осыпь для примера

x', то и x' – сосед для x. В дальнейшем мы будем предполагать, что функция расстояния обладает озвученными свойствами. В частности, любая метрика обладает написанными свойствами.

Теперь определим класс «хороших объектов» – объектов, на которые будут опираться, или за счет которых будут строиться наши кластеры – объекты, собирающие вокруг себя достаточное количество соседей.

Определение 4.2.4 *Пусть $m \in \mathbb{N}$ – некоторое наперед заданное число. Если*

$$|\overline{B}(x_0, \varepsilon)| \geq m,$$

то объект x_0 называется корневым объектом.

Итак, корневой объект – это объект, у которого есть «достаточное» число (как минимум m штук) соседей.

Ну что, мы все подготовили для того, чтобы перейти к формальному описанию алгоритма DBSCAN. В рамках обозначений и терминологии, принятых в этом разделе, алгоритм таков:

1. Выбираются $\varepsilon > 0$, $m \in \mathbb{N}$. Пусть $i = 1$.

2. Находится какой-нибудь корневой объект x в X. Если таковых нет, то производится переход к пункту 7.

3. Все соседи объекта x помещаются в кластер C_i.

4. Для каждого элемента из C_i проверяется, является ли он корневым. Если является, то все соседи найденного корневого элемента добавляются к C_i.

5. Пункт 4 повторяется до тех пор, пока состав кластера C_i не перестанет изменяться.

6. Полагается $X = X \setminus C_i$, $i = i + 1$. Производится переход к пункту 2.

7. Все объекты множества X, если они есть, помещаются в отдельный кластер – выбросы. Кластеризация завершена.

Проиллюстрируем описанный алгоритм. Пусть имеется набор данных, изображенный на рисунке 29, $m = 3$, функция расстояния d – обычное евклидово расстояние, $\varepsilon > 0$. Согласно алгоритму, сначала находим какой-нибудь корневой объект; например, объект, обведенный в синий квадрат, является корневым, так как имеет трех соседей, ведь круг радиуса ε вокруг него содержит ровно 3 объекта. Следуя алгоритму, относим объекты, находящиеся внутри круга, к кластеру C_1.

Теперь нужно проверить: будут ли элементы кластера C_1 корневыми, и если да – добавят ли они в него элементов? Из геометрических соображений понятно, что каждый (а всего их 3) объект образовавшегося кластера является корневым, поэтому помещаем в кластер C_1 и соседей только что найденных объектов. За второй проход к кластеру C_1 примкнут еще три элемента (найдите их!), а на третьем проходе к кластеру C_1 примкнет лишь один, причем на этот раз некорневой объект, и, тем самым, первый кластер будет выделен, рисунок 30. Легко понять, что аналогичным образом будет выделен и второй кластер, см. рисунок 31. На этом этапе хочется обратить внимание на два

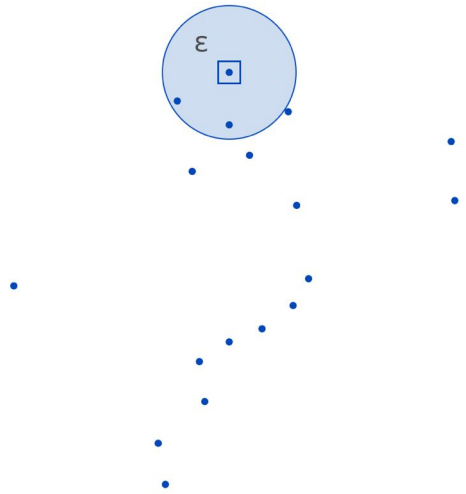

Рис. 29: Пример исходных данных

момента. Сначала взгляните на объект с желтым восклицательным знаком. Совершенно ясно, что он попал к синим квадратам только за счет того, что первым был выбран корневой элемент, сформировавший синий кластер. Если бы первым был выбран какой-либо корневой элемент, сформировавший желтый кластер, то этот объект был бы зачислен к желтым.

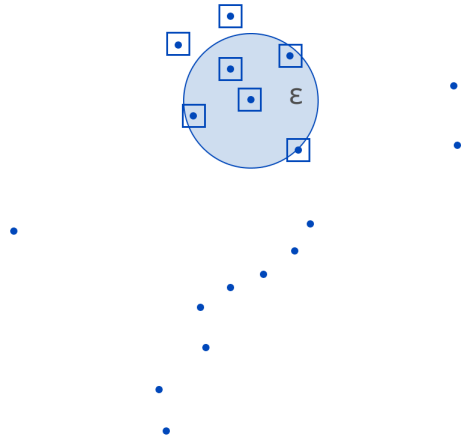

Рис. 30: Выделение первого кластера

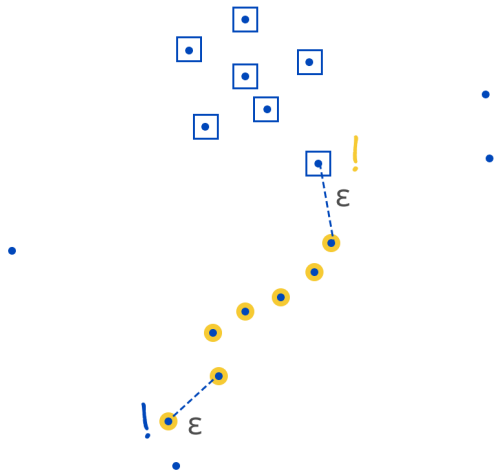

Рис. 31: Два выделенных кластера

Теперь взгляните на объект с синим восклицательным знаком. Он не является корневым, в окружающем его шаре радиуса ε лишь две точки, принадлежащие к желтому кластеру, поэтому точка, близкая к нему, в желтый кластер не попадает. Так как больше не осталось ранее нерассмотренных корневых точек, то все объекты, нарисованные синими точками – это выбросы или шум (их 4 штуки). Кластеризация окончена.

Замечание 4.2.2 *Не вдаваясь в детали отметим, что при фиксированных параметрах алгоритма m и ε, корневые элементы, если они есть, распределяются по кластерам однозначно. То же самое касается и выбросов. В то же время, некорневые объекты, не являющиеся выбросами, могут менять кластерную принадлежность в зависимости от порядка выбора корневых элементов, мы это обсудили в приведенном примере.*

Данное замечание показывает, что как только выбраны параметры алгоритма, все кластеризуемые объекты делятся на один из трех типов: корневые

объекты, выбросы и так называемые граничные объекты.

Определение 4.2.5 *Некорневой объект, не являющийся выбросом, называется граничным объектом или граничной точкой.*

Еще раз отметим, что граничные объекты могут менять свою кластерную принадлежность при перезапуске алогоритма даже с неизменными параметрами.

Из приведенных иллюстраций также ясно, что при малых значениях m DBSCAN хорошо справляется с поиском и выделением ленточных кластеров. Однако, здесь есть и подводный камень: если кластеры соединены не очень разреженными перемычками, то, скорее всего, DBSCAN объединит эти кластеры в один единственный, и отработает куда хуже, чем ранее рассмотренные методы. Главная проблема заключается в том, что мы зачастую не можем понять: перед нами ленточный кластер или кластеры с перемычками, а значит не можем и «подсказать» алгоритму, поднастроив его параметры. Все потому, что зачастую мы не можем визуализировать кластеризуемые данные из-за большой размерности пространства признаков.

Замечание 4.2.3 *Отметим также, что так как понятия корневого объекта, соседей завязаны на понятии расстояния, то признаки кластеризуемых объектов перед началом кластеризации имеет смысл либо стандартизировать, либо нормировать.*

Естественно, возникает и следующий вопрос: а как подобрать параметры алгоритма DBSCAN оптимальным образом? Конечно, можно устраивать перебор и каким-то образом оценивать качество кластеризации, но и здесь бывают проблемы: многие методы оценки качества кластеризации считают, что кластеры – это плотные шаровые сгустки, и тогда ленточных кластеров нам просто не найти. В принципе, на этот счет существует достаточно много эвристик, мы не будем говорить о них в нашей лекции, а заинтересованных слушателей отправляем к дополнительным материалам.

5 Заключение

Итак, в этой лекции мы познакомились с некоторыми алгоритмами кластеризации, которые успешно применяются в повседневных задачах. Выбор алгоритма, его параметров, – задача, которая до сих пор под силу лишь только аналитику-исследователю. Пробуйте, набирайтесь опыта, и со временем результат превзойдет любые ваши ожидания. Удачи!