ЦИФРОВАЯ КУЛЬТУРА

УНИВЕРСИТЕТ ИТМО

Лекция **«Кластеризация»**

Санкт-Петербург
2019

Содержание

1 Что будет рассмотрено в этом разделе

Обучение без учителя имеет принципиальное отличие от обучения с учителем и призвано решать другой класс задач. В случае обучения с учителем мы имели набор предикторов $X = (X_1, X_2, \ldots, X_p)$ и некоторый отклик Y. Основные задачи заключались либо в предсказании отклика Y на новых значениях предикторов, либо в «восстановлении» зависимости, то есть самой функции f, для качественного определения зависимости отклика от конкретного или конкретных предикторов, а именно $Y = f(X) + \varepsilon$, где ε – случайная ошибка. А как быть, если у нас есть некоторый набор объектов, характеризующихся какими-либо атрибутами, те самые предикторы $X = (X_1, X_2, \ldots, X_p)$, но при этом мы не заинтересованы ни в прогнозировании отклика Y, ни в построении функции, моделирующей какой-либо процесс, а хотим проанализировать закономерности (возможно скрытые) в самом наборе рассматриваемых объектов. В этом разделе мы рассмотрим одно из основных направлений обучения без учителя, а именно кластеризацию.

Суть задачи кластеризации заключается не в разделении объектов по заранее известным критериями или заранее определенным классам (как, например, в задаче классификации), а в необходимости выявить такие классы (или кластеры), к которым объекты будут отнесены на основе имеющихся данных.

Приведем следующий пример. Допустим, у нас есть некоторое количество животных, обладающих определенным набором как числовых атрибутов (размеры, скорость бега, продолжительность жизни), так и нечисловых (рацион питания, умение летать, умение плавать). Кто конкретно эти животные мы не знаем, но зная их характеристики можно выявить некоторые группы (кластеры), в которых объекты будут обладать схожими характеристиками. Причем кластеризацию в таком случае можно производить разными методами и получать разные результаты. Например, животные могут разделиться по принципу хищные и травоядные, а могут быть сгруппированы по ареолам обитания, типу размножения, средней продолжительности жизни и тому подобное. Различные методы кластеризации могут давать разные результаты в зависимости от исходных данных, поэтому выбор метода – одна из важнейших задач, стоящих перед исследователем.

2 Кластеризация

Под кластеризацией понимается большой набор различных методов по выделению подгрупп или кластеров из исходного набора данных. Суть заключается в поиске такого разделения, при котором объекты, принадлежащие одной группе, имеют максимальное сходство, при этом объекты, принадлежащие разным группам, обладают существенными отличиями.

Одну и ту же задачу кластеризации можно решить, используя различные алгоритмы, каждый из которых обладает как своими преимуществами, так и недостатками. Дело в том, что в зависимости от того, какими являются входные данные, тот или иной метод может быть более точным и/или более эффективным с точки зрения производительности. На рисунке 1 представлены результаты работы нескольких алгоритмов с одними и теми же начальными наборами данных. Цветами изображены кластеры, полученные в результате работы алгоритмов. В каждом случае рассматривалось 2000 объектов.

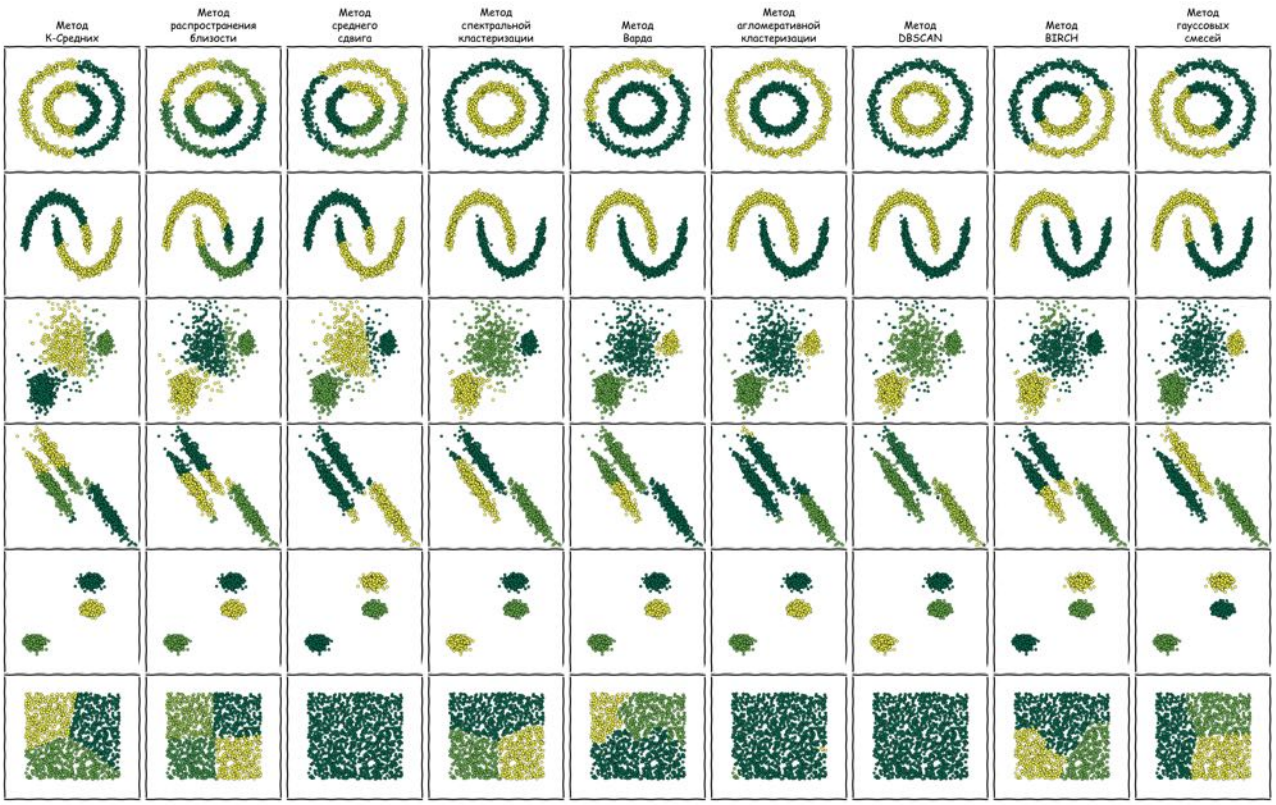

Рис. 1: Сравнение алгоритмов кластеризации

Как можно заметить, алгоритмы по-разному справляются с одной и той же задачей. В первом случае (первая строка), наверное, самым правильным было бы разделение на внешнее и внутреннее кольца (справились спектральная кластеризация, агломеративная кластеризация и DBSCAN). В случае полумесяцев (вторая строка), фавориты те же. Причем обратите внимание на

алгоритм распространения близости. Явно не лучший результат. В третьем случае (третья строка) при интуитивном разбиении возможны различные варианты, что и демонстрируют все алгоритмы. В четвертой строке представлены так называемые полосовые кластеры, где наиболее эффективно сработали спектральная кластеризация, DBSCAN и гауссова смесь. С предпоследним набором данных справились все.Все дело в форме кластеров и в удалении, на котором они находятся друг от друга. В последнем случае использовалось классическое равномерное распределение, и правильный ответ предполагает наличие всего одного кластера. Здесь лучше всего себя показали средний сдвиг, агломеративная кластеризация и DBSCAN. Как видно из приведенного примера, задача кластеризации весьма трудна не в смысле сложности отдельно взятого алгоритма (хотя некоторые могут быть крайне сложными), а в выборе того или иного метода, так как в разных ситуациях можно получать абсолютно разные результаты.

3 Метод K-средних

3.1 Общее описание метода

На первый взгляд задача распределения n объектов по K кластерам выглядит не очень простой, особенно если K и n достаточно велики, так как существует K^n способов распределить n объектов по K кластерам. Перебор всех возможных вариантов в поисках оптимального, конечно, не самое разумное решение, при этом нужно еще определиться, что называть «оптимальным».

Исходя из рассмотренного примера сравнения различных алгоритмов кластеризации, метод K-средних является далеко не самым точным, однако у него есть неоспоримые преимущества. Этот метод по праву считается наиболее интуитивно понятным (поэтому во многих источниках обзор начинается именно с него). Как следствие, из-за своей простоты обеспечивает хорошую скорость работы. Метод K-средних – это простой и элегантный способ разбиения данных на K различных, непересекающихся множеств. В качестве первого шага необходимо определить само число K, эта функция отводится исследователю. Конечно, можно произвести расчеты при различных значениях K и выбрать наиболее подходящий вариант. Однако встает вопрос, как определить, что одно разбиение лучше, чем другое? Оценка эффективности разбиения – тема, заслуживающая отдельного разговора, в рамках же этой лекции мы коснемся ее весьма поверхностно. После определения параметра K, необходимо отнести каждый объект к какому-либо кластеру. Дадим формальное определение понятию кластер.

Определение 3.1.1 *Пусть C – множество всех рассматриваемых объектов. Непустые множества $C_1, C_2, \ldots, C_K \subset C$ называются кластерами, если их объединение совпадает с C, а пересечение любых двух из них есть пустое множество:*

$$C_1 \cup C_2 \cup \ldots \cup C_K = C,$$
$$\forall (k \neq k') \Longrightarrow C_k \cap C_{k'} = \varnothing.$$

В математике семейство множеств $C_1, C_2, \ldots, C_K$ называют разбиением множества C. Если объект x_i находится в k-ом кластере, то пишут: $x_i \in C_k$.

Основная идея кластеризация заключается в том, что разбиение тем лучше, чем более «похожи» объекты в рамках одного кластера. Отсюда возникает вполне резонный вопрос: что можно использовать для оценки «похожести» объектов между собой в рамках одного кластера и «непохожести» в рамках двух различных кластеров. Введем некоторые определения:

Определение 3.1.2 *Пусть $d(x_i, x_{i'})$ – функция расстояния между объектами $x_i, x_{i'}$. Внутрикластерным расстоянием называют сумму расстояний между всеми объектами одного кластера:*

$$W(C_k) = \sum_{x_i, x_{i'} \in C_k} d(x_i, x_{i'}).$$

Определение 3.1.3 *Средним внутрикластерным расстоянием называется величина:*

$$\frac{W(C_k)}{|C_k|},$$

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

Ответом на ранее поставленный вопрос может служить следующее: для оценки «похожести» объектов в рамках одного кластера можно использовать среднее внутрикластерное расстояние. Тогда кластеризацию будем производить так, чтобы сумма средних внутрикластерных расстояний была минимальной. Это будет гарантировать корректное, с нашей точки зрения, разделение на кластеры, да и вообще это кажется логичным. Иными словами, будем минимизировать следующее выражение:

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

Для перехода к рассмотрению алгоритма и вычислениям, необходимо определиться с функцией расстояния. В предыдущей лекции в качестве расстояния d рассматривались евклидово и манхэттенское расстояния. Мы будем использовать первое. Если объект x_i имеет p координат (атрибутов):

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}),$$

то евклидово расстояние между объектами x_i и $x_{i\prime}$ определяется следующим образом:

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

Обратим внимание на один существенный момент. Отслеживать изменение значения заявленной минимизируемой функции

$$\sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{x_i, x_{i'} \in C_k} d\left(x_i, x_{i'}\right) \to \min$$

достаточно сложно, так как при изменении принадлежности объекта кластеру во внешней сумме меняется не только числитель, но и знаменатель, причем сразу у двух слагаемых в сумме. Удобнее в таком случае привязаться к какому-то инварианту кластера на данной итерации. Для этого можно выполнить 2 перехода. Сначала договоримся вместо минимизации суммы средних внутрикластерных расстояний минимизировать суммы средних внутрикластерных квадратов расстояний:

$$\sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{x_i, x_{i'} \in C_k} d_E^2\left(x_i, x_{i'}\right) = \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{x_i, x_{i'} \in C_k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2 \to \min.$$

С точки зрения логики построения ничего не нарушено: минимизация суммы расстояний или суммы квадратов расстояний в рамках нашей задачи почти одно и то же. Однако, теперь близкие объекты имеют меньший вес, а далекие – больший. Для второго перехода нам потребуется ввести еще одно определение:

Центроидом $\overline{x}_k = (\overline{x}_{k1}, \ldots, \overline{x}_{kp})$ кластера C_k будем называть объект, координаты которого имеют вид:

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

При этом алгебраически можно показать, что справедливо следующее тождество:

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

Кстати, попробуйте осознать его самостоятельно: откуда, на интуитивном уровне, появляется двойка? Почему это и правда верно?

Объединяя оба перехода и выкидывая двойку, которая не влияет на оптимизацию, получим выражение, которое и будем минимизировать:

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

Резюмируя получаем, что задача минимизации суммы средних внутрикластерных расстояний сводится к задаче минимизации суммы квадратов расстояний до центроидов, а центроиды и есть инварианты на каждой конкретной итерации.

3.2 Алгоритм

Пусть всего имеется n объектов: $x_1, x_2, \ldots, x_n$. Рассмотрим необходимую последовательность действий для выполнения кластеризации методом K-средних.

1. **Инициализация.** Каждый объект случайным образом относят к какому-либо одному кластеру (от C_1 до C_K). На рисунке 2 представлены $K = 5$ кластеров.

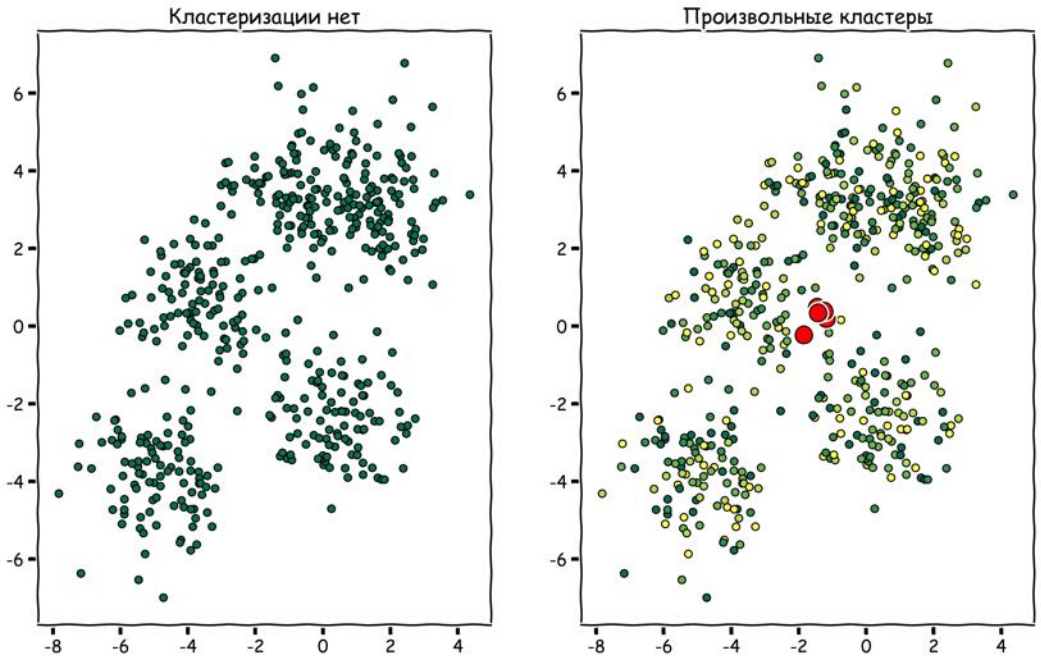

Рис. 2: Инициализация

2. **Нахождение центроидов.** Для каждого кластера C_k находим координаты центроида:

$$\overline{x}_{kj} = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_{ij},$$

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

На рисунке 2 центроиды изображены красными кружками.

3. **Вычисление квадратов расстояний до центроидов.** Находим квадрат евклидова расстояния от i-го объекта до центроида каждого кластера:

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

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

4. **Перераспределение.** Объект x_i относят к кластеру с наиболее близким к нему центроидом, то есть $x_i \in C_{k^*}$, где

$$k^* = \arg\min_{k \in K}(d_E^2(x_i, \overline{x}_k)).$$

В качестве замечания к этому пункту можно добавить, что если квадрат расстояния от объекта до центроида какого-либо кластера равен квадрату расстоянию до центроида текущего кластера, принадлежность не изменяется.

Далее шаги 2 — 4 повторяются, пока объекты не перестанут перераспределяться по кластерам (рисунок 3).

Можно заметить, что во время работы алгоритма кластеризация на каждой итерации будет улучшаться до тех пор, пока объекты изменяют свою принадлежность. На этапе перераспределения объектов по новым кластерам сумма:

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

будет уменьшаться, так как объекты будут переходить в тот кластер, центроид которого ближе. Тогда сумма квадратов расстояний от объектов x_i, до соответствующего центроида $\overline{x}_k$

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

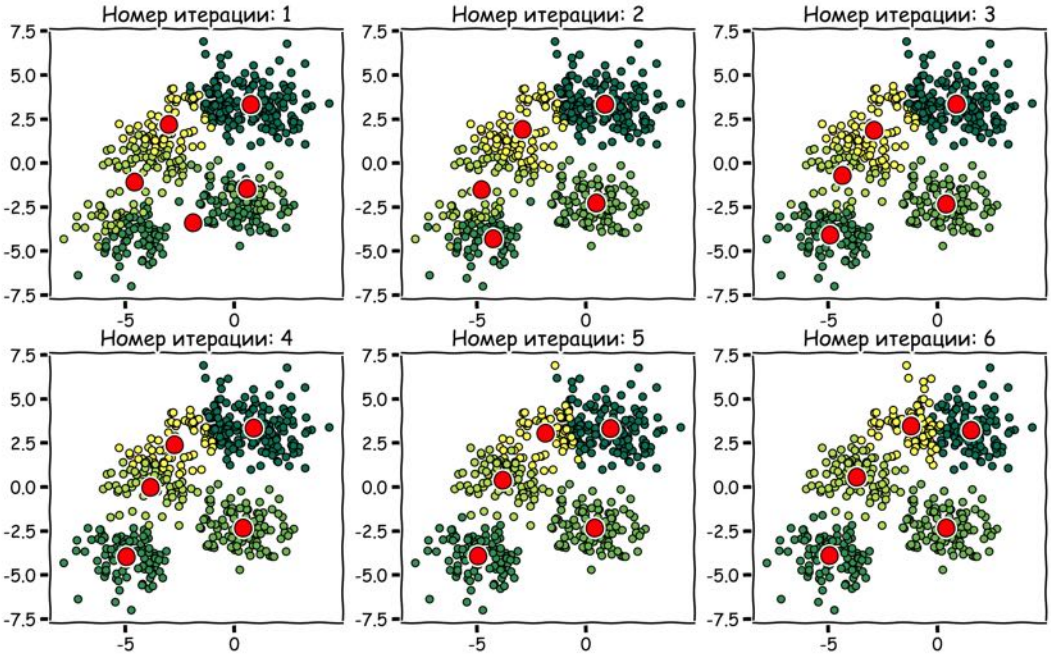

Рис. 3: Итерации кластеризации

кластера k, из которого ушел объект уменьшится сильнее, чем увеличится аналогичная сумма для кластера, в который этот объект пришел. Нахождение новых центроидов также, как минимум не увеличивает эту сумму внутри каждого кластера.

Алгоритм уменьшает минимизируемое выражение на каждом шаге, и если результат более не изменяется, а объекты не изменяют своей принадлежности, то алгоритм останавливается, и говорят, что достигнут локальный оптимум. При этом, так как всего возможно K^n распределений объектов по кластерам, то алгоритм конечен. Важно помнить, что на шаге 1 алгоритма объекты распределялись по кластерам случайным образом, поэтому оптимум и называется локальным, и при другом начальном распределении по кластерам может меняться.

Для демонстрации зависимости итогового результата от первоначального распределения объектов по кластерам, обратимся к следующему примеру. Рассмотрим один и тот же набор данных. Каждый объект обладает двумя атрибутами, и может быть изображен на координатной плоскости. Выполним кластеризацию согласно нашему алгоритму, отличаться в каждом из 4 случаев будет лишь начальное распределение объектов по кластерам (шаг 1). Распределять в этом случае будем на $K = 5$ кластеров. Для оценки полученных результатов в каждом случае найдем сумму средних внутрикластерных расстояний. Результаты представлены на рисунке 4.

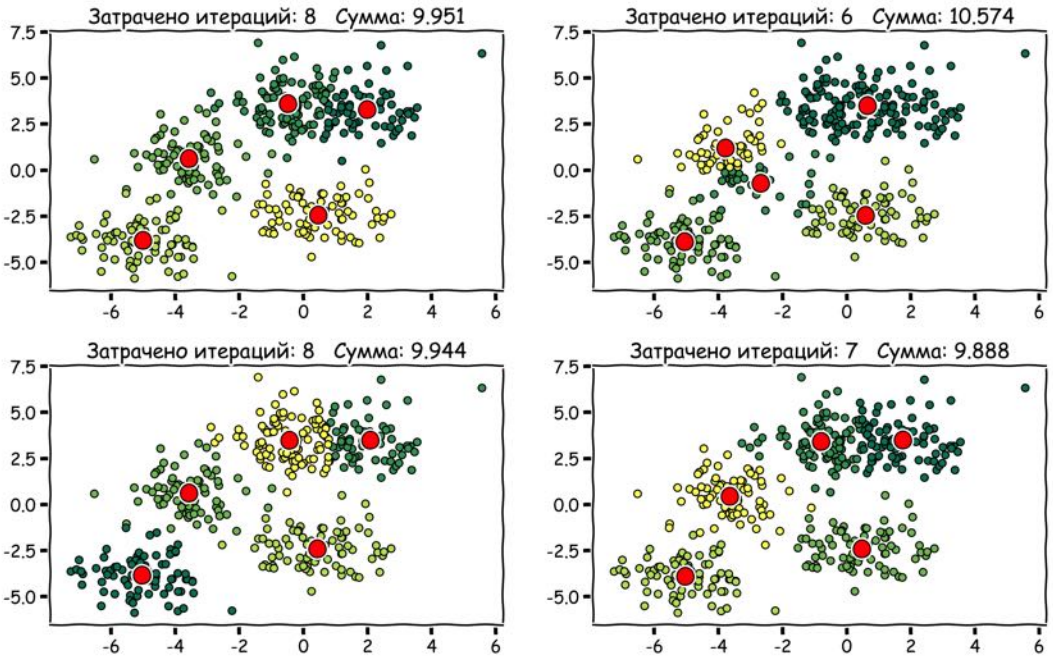

Рис. 4: Различные варианты инициализации

Как можно заметить, кластеры распределились по-разному, хотя исходные объекты одни и те же. Отличаются также и суммы средних внутрикластерных расстояний. Напомним, что по нашему предположению, чем ближе точки кластера друг к другу, тем лучше. В рассматриваемом случае из 4 вариантов начальной «раскраски» точек самой оптимальной оказалась кластеризация 4 с суммой 9.888. При этом каждый из этих 4 вариантов является локальным оптимумом в соответствии со своим случаем начальной «раскраски» точек.

3.3 Пример: хруст и сладость продуктов

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.

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: а сколько же кластеров мы будем брать? 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 (овощи, фрукты и протеины) и посмотрим, что у нас получится.

На первом шаге необходимо отнести каждый объект к одному из трех кластеров. Заполним произвольным образом столбец «Кластер» числами от 1 до 3.

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:

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

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

Аналогично найдем координаты $\overline{x}_{k1},\overline{x}_{k2}$ 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 5:

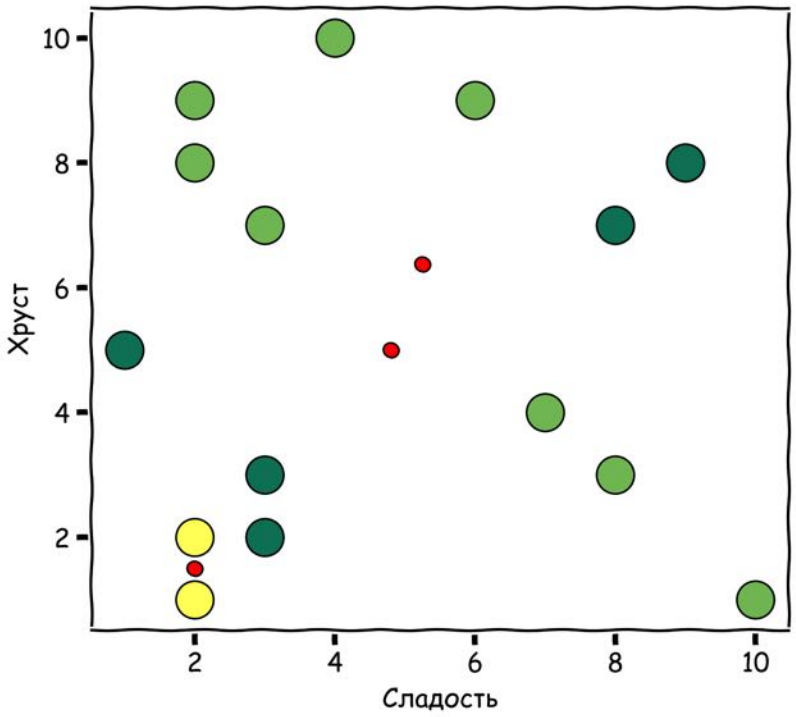

Рис. 5: Инициализация и нахождение центроидов

Переходим к шагу 3. Теперь необходимо вычислить квадраты расстояний от каждой точки до всех трех центроидов, выбрать наименьшее, тем самым переопределив принадлежность объектов кластерам. 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	2
морковь	4	10	2
сельдерей	2	9	2
салат	3	7	2
груша	8	7	2
перец	6	9	2

Далее повторяем шаг 2 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 3, находим расстояние от каждого объекта до новых центроидов и выбираем наименьшее. Выполняя расчеты, можно заметить, что ни один объект не изменяет своей принадлежности кластеру, а значит алгоритм закончен. 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 6).

В качестве проверки качества модели, можно посчитать суммы средних внутрикластерных расстояний. Рассмотрим 3 варианта начальной «раскраски» объектов на шаге 1 (Рисунок 7).

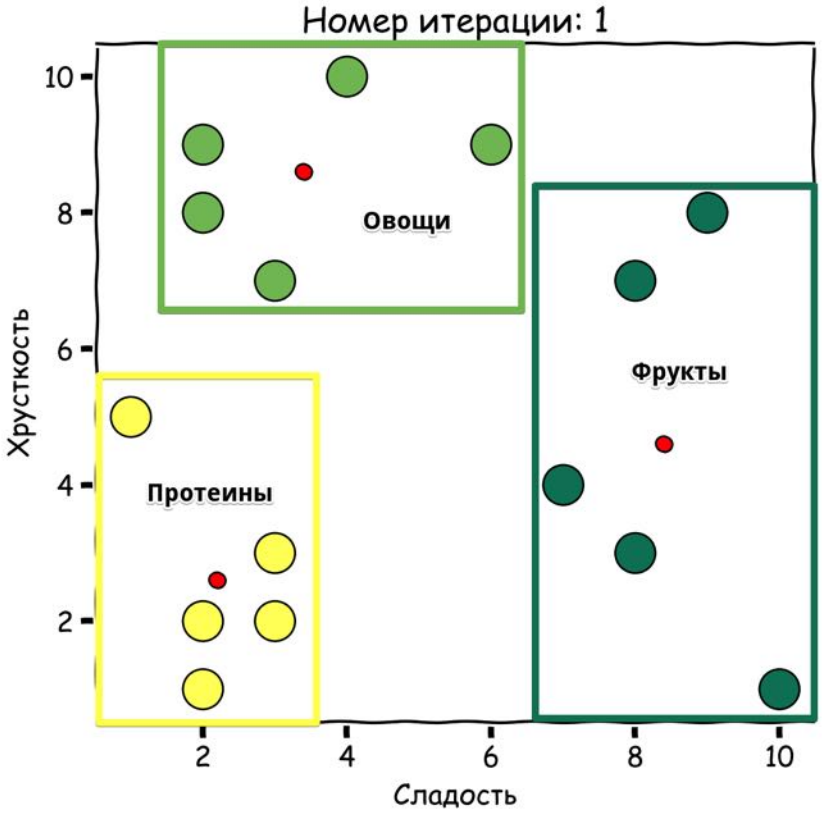

Рис. 6: Результаты кластеризации

Как видно из полученных результатов, ранее произведенная кластеризация (результат 13.36), претендует на то, чтобы быть глобальным оптимумом, при этом существуют и другие варианты разбиения с более плохими значениями сумм средних внутрикластерных расстояний.

4 Агломеративная кластеризация

4.1 Общее описание и алгоритм метода

Рассмотренный метод K-средних имеет ряд достоинств, такие как простота и скорость работы, но обладает существенным недостатком, а именно необходимостью заранее определять число кластеров. Агломеративная или, как еще ее называют, иерархическая кластеризация, позволяет исследователю определять число кластеров уже после самой процедуры кластеризации. Еще одним достоинством иерархической кластеризации можно считать то, что результаты представляются в весьма наглядной форме, которую называют дендрограммой. Дендрограммы получили свое название из-за схожести с перевернутыми деревьями (дендрос с древнегреческого переводится как де-

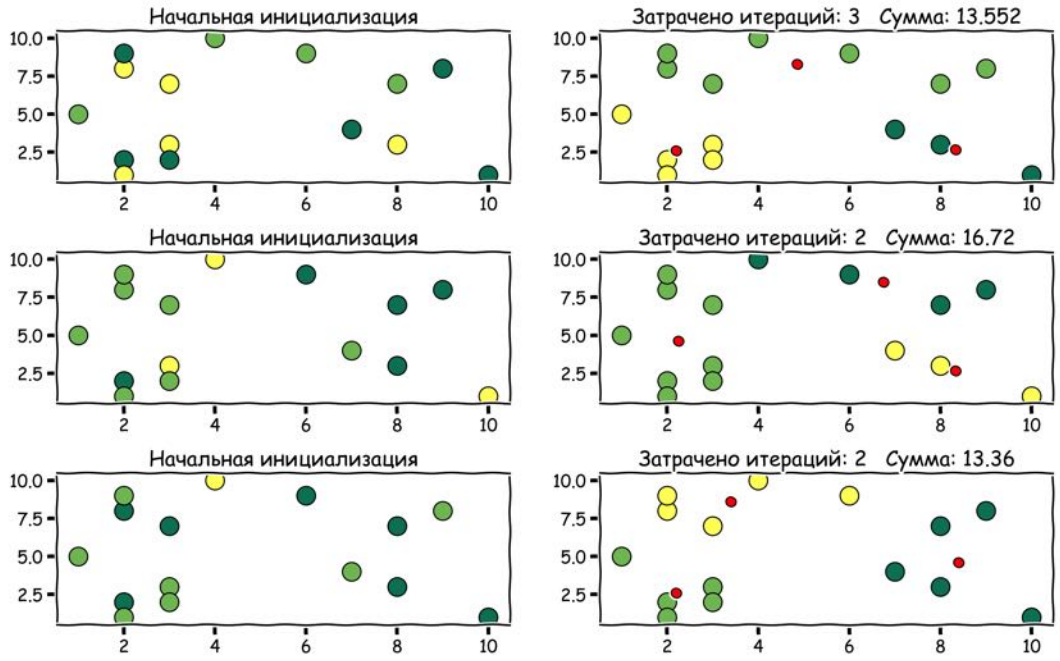

Рис. 7: Различные варианты инициализации

рево). Построение начинается снизу, где каждый объект образует свой собственный кластер. Затем группы схожих объектов/кластеров объединяются в более крупные кластеры, которые, в свою очередь, потом объединяются в еще более крупные кластеры до тех пор, пока не получится один мега-кластер, который будет являться условным стволом построенного дерева.

Рассмотрим алгоритм чуть более подробно. На начальном этапе каждый объект представляет собой отдельный независимый кластер, то есть n объектов составляют n кластеров. На следующем шаге все кластеры сравниваются между собой, выбираются 2 наиболее «похожих» (функция похожести выбирается исследователем) и объединяются в один. После этого остается $n-1$ кластер, далее опять сравниваются все кластеры и объединяются 2 ближайших так, что остается $n-2$ кластера. Алгоритм продолжается до тех пор, пока все кластеры не будут объединены в единый. Находить расстояние между двумя объектами, зная их координаты, мы уже умеем, а как сравнивать «похожесть» кластеров, состоящих из нескольких объектов?

Пусть кластер X состоит из элементов $\{x_1, x_2, \ldots, x_s\}$, а кластер X' – из элементов $\{x'_1, x'_2, \ldots, x'_r\}$. Пусть также $d(x, x')$ – некоторая функция расстояния(метрика) между объектами x и x'. Обозначим через $\rho(X, X')$ – функцию «похожести» между кластерами. Говоря о похожести кластеров, мы тоже будем использовать термин расстояние, хотя функция $\rho(X, X')$ может и не являться метрикой. Рассмотрим основные подходы к определению расстояния

между кластерами:

1. **Метод полной связи.** (Рисунок 8). Все элементы $x \in X$ попарно сравниваются с элементами $x' \in X'$. За расстояние между кластерами принимается наибольшее из всех полученных значений:

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

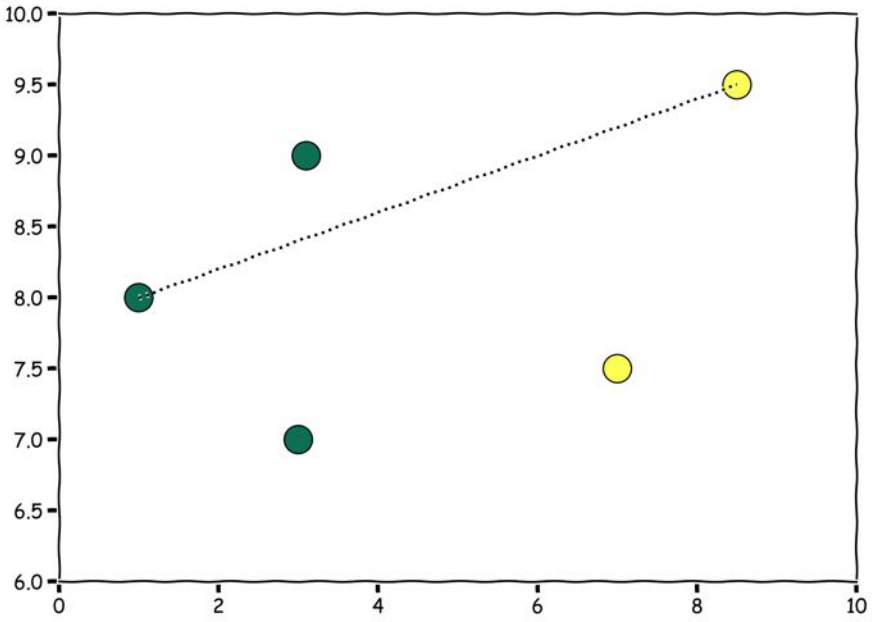

Рис. 8: Метод полной связи

2. **Метод одиночной связи.** (Рисунок 9). В качестве расстояния между кластерами принимается минимальное из расстояний между парами элементов:

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

3. **Метод средней связи.** (Рисунок 10). Расстояние между кластерами вычисляется как среднее значение расстояний между парами элементов:

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

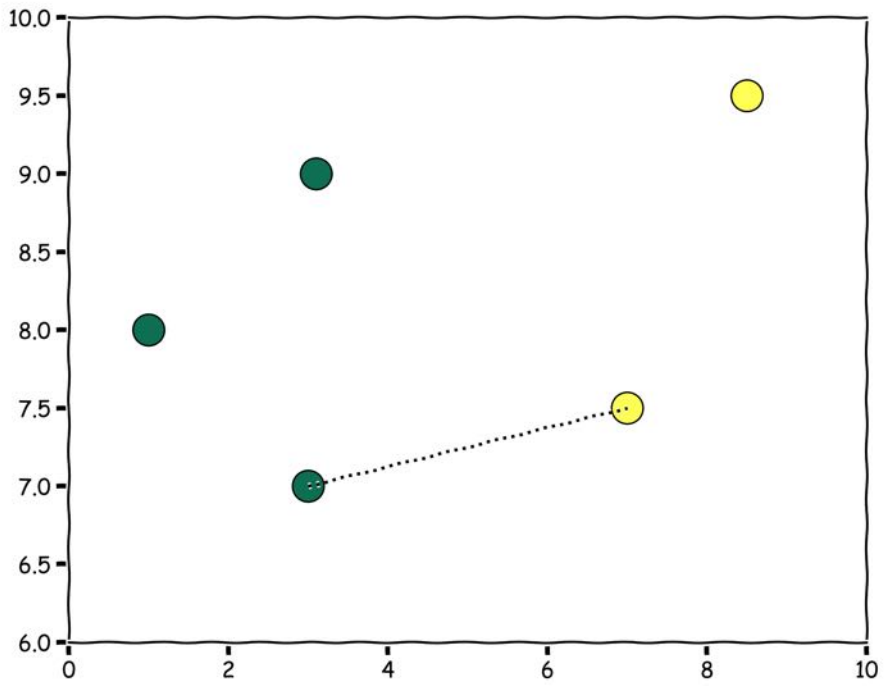

Рис. 9: Метод одиночной связи

4. **Центроидный метод.** (Рисунок 11). За расстояние между кластерами принимается расстояние между их центроидами.

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

где $\overline{x}$ и $\overline{x}'$ – центроиды соответственно кластеров X и X'.

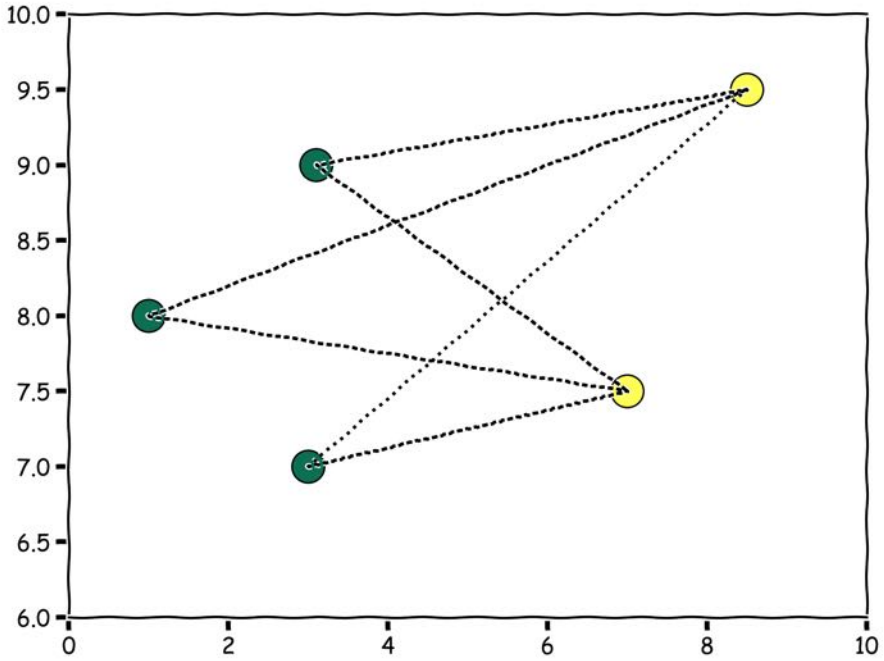

Рис. 10: Метод средней связи

4.2 Пример сладость и хруст продуктов

Рассмотрим процедуру кластеризиции и построения дендрограммы на уже знакомом нам примере с хрустом и сладостью пищи. Напомним исходные данные:

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

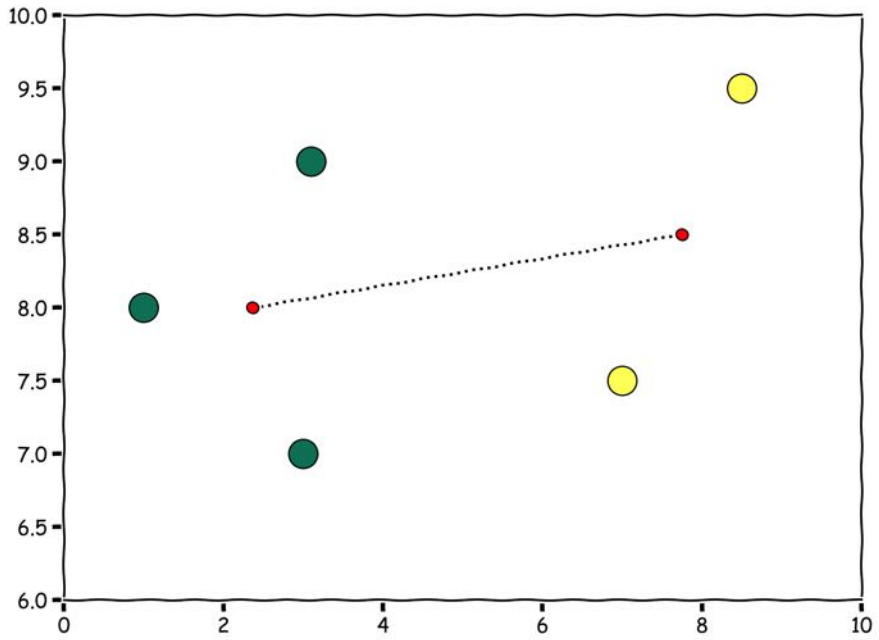

Рис. 11: Центроидный метод

Дендрограмма для этого набора данных представлена на рисунке 12. Давайте разберемся в том, как ее строить и как интерпретировать результаты.

Для удобства, вместо точек на плоскости будем использовать соответствующие номера объектов из таблицы исходных данных, как показано на рисунке 13. Получим, что объект «банан» имеет номер 1 и координаты $(10, 1)$, объект «апельсин» – номер 2 и координаты $(7, 4)$ и так далее.

Нужно определиться с методом нахождения расстояния как между кластерами, так и между непосредственно объектами. В рассматриваемом примере в качестве первого мы будем использовать метод полной связи, а второго – евклидово расстояние.

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

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

$$x = (x_1, x_2, \ldots, x_p),$$

$$x' = (x'_1, x'_2, \ldots, x'_p).$$

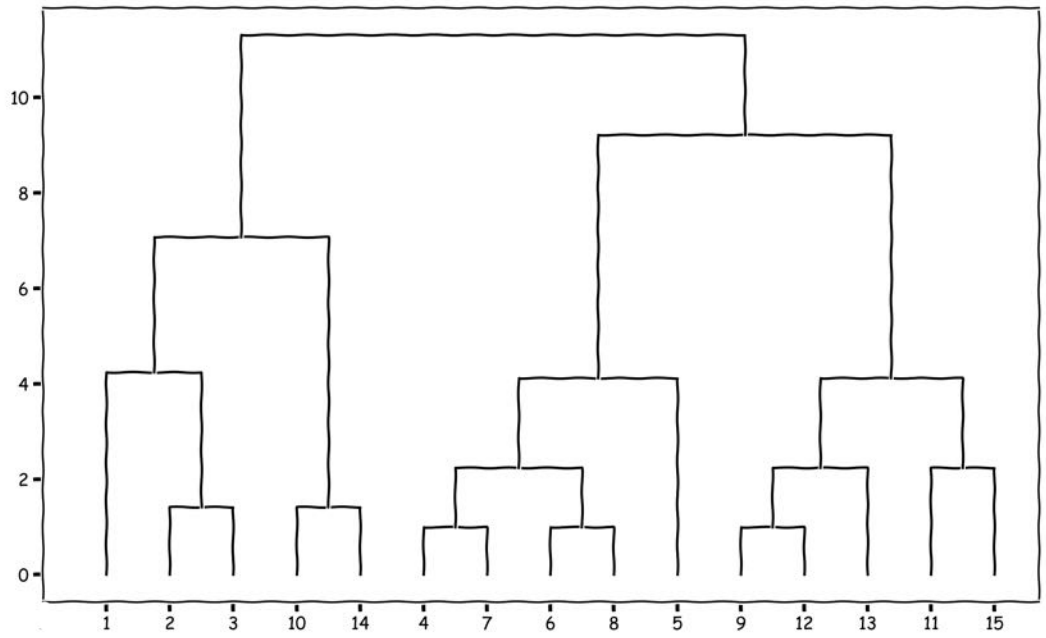

Рис. 12: Агломеративная кластеризация

В самом начале мы имеем $n = 15$ кластеров, так как каждый объект представляет собой отдельный кластер. Необходимо сравнить каждый кластер с каждым и выбрать 2 максимально близких. Для краткости записи, и удобства поиска, составим матрицу расстояний. В качестве элементов матрицы выступают расстояния между рассматриваемыми объектами. На рисунке 14 приведены округленные результаты с точностью до 1 знака после целой части. При настоящих расчетах округления лучше не использовать.

Матрица расстояний симметрична относительно главной диагонали, так как евклидово расстояние является метрикой. Можно заметить, что у нас получилось несколько пар, расстояния между которыми равны единице:

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

но так как по порядку пара $(4,7)$ идет первой, сначала объединим «креветку» и «сыр». Договоримся обозначать кластер фигурными скобками. Например, кластер, содержащий элементы 6 – орехи и 8 – рыба, будем обозначать: $\{6,8\}$.

Теперь у нас осталось 14 кластеров, причем один из них состоит из 2-ух элементов. По аналогии с предыдущей итерацией ищем расстояния между кластерами, учитывая выбранный метод. Для примера, найдем расстояние от кластера $\{9,12\}$ до кластера $\{13\}$ (забегая вперед, признаемся, что они объединятся в один кластер на 7 итерации). Можно воспользоваться уже

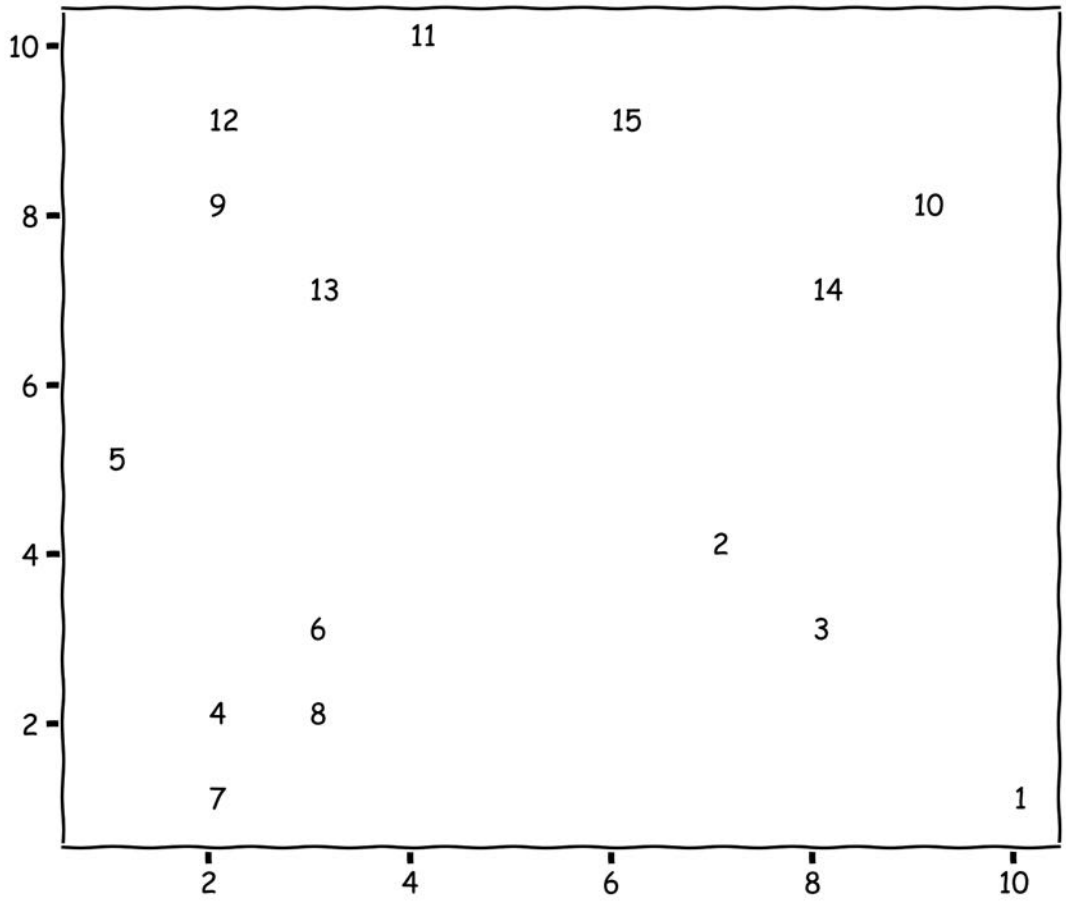

Рис. 13: Исходные данные по номерам объектов

найденной матрицей расстояний. Тогда получим:

$$d(9, 13) \approx 1.4,$$

$$d(12, 13) \approx 2.2.$$

За расстояние между кластерами $\{9, 12\}$ и $\{13\}$ берется максимальное:

$$\rho(\{9, 12\}, \{13\}) = \max(d(9, 13), d(12, 13)) \approx 2.2.$$

Аналогично вычисляются расстояния между всеми кластерами на текущей итерации и объединяются два ближайших.

Приведем список всех объединений в рамках рассматриваемой кластеризации:

1. $\{4\} \cup \{7\}$.

	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	2.2	4	3.6	2.8	0

Рис. 14: Матрица расстояний

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 того, как кластеризация выполнена, можно приступать к построению дендрограммы. 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 необходимо соединить объекты (листья) 4 и 7 «веткой», высота которой равна расстоянию между кластерами $\{4\}$ и $\{7\}$. Затем аналогично соединяются «ветками» пары листьев $(6, 8), (9, 12), (2, 3), (10, 14)$. Причем высоты веток соответствуют расстояниям между этими объектами. Дальше необходимо соединить кластеры $\{4, 7\}$ и $\{6, 8\}$. Высота ветки ищется следующим образом: сначала необходимо найти расстояние между кластерами $\{4, 7\}, \{6, 8\}$.

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

Далее из полученного значения вычитается наибольшая из высот веток, соответствующих кластерам $\{4, 7\}$ и $\{6, 8\}$. Для нашего случая:

$$\rho(\{4, 7\}, \{6, 8\}) - \max(\rho(\{4\}, \{7\}), \rho(\{6\}, \{8\})) = 2.2 - 1 = 1.2$$

Попробуйте самостоятельно найти, например, $\rho(\{9, 12, 13\}, \{11, 15\})$, используя матрицу расстояний (правильный ответ: $d_E(9, 15) \approx 4.1$).

При дальнейшем продвижении вверх, образованные ветки объединяются с другими листьями или даже ветками. Чем более похожи объекты, тем на более ранней стадии они будут объединены в один кластер. С другой стороны, чем ветки ближе к стволу, тем объекты, входящие в соответствующие кластеры, будут сильнее различаться. Степень различия между двумя ветками и отражается по оси ординат, то есть чем объекты сильнее отличаются, тем более высокой будет объединяющая их ветвь. Вся последовательность построения показана на рисунке 15.

В начале этого фрагмента было анонсировано, что иерархическая кластеризация учитывает недостаток метода K-средних в плане определения количества желаемых кластеров до самой кластеризации. При помощи дендрограммы, исследователь также должен определять количество кластеров самостоятельно, однако он это может сделать уже постфактум, подбирая наиболее оптимальное и правдоподобное число кластеров (рисунок 16). Нужно лишь провести горизонтальную черту, и тогда количество пересечений с вертикальными соединениям веток даст желаемое число кластеров.

Если в соответствии с нашими представлениями о еде выделять 3 кластера, то результат можно наблюдать на рисунке 17. Римскими цифрами отмечены итерации, на которых произошло объединение.

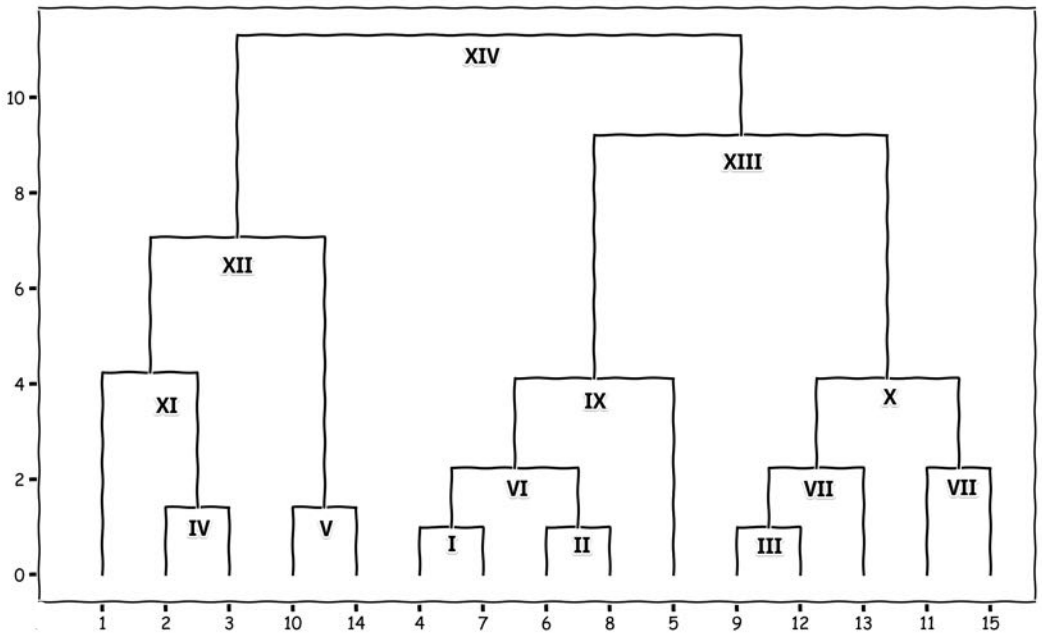

Рис. 15: Последовательность построения дендрограммы

Немного потренируемся «на глаз» определять количество кластеров по дендрограмме. Сколько кластеров вы бы выделили, посмотрев на картинку 18?

Четыре? Пять? Или может больше? Конечно, все зависит от того, где провести горизонтальную линию отсечения. Правильного ответа нет, потому что эти результаты можно интерпретировать в зависимости от остального контекста, но синтезировались данные все-таки из расчета наличия 5-ти различных кластеров.

5 Оценка качества кластеризации

Оценка качества кластеризации – достаточно неоднозначный процесс. Различные алгоритмы могут давать не одинаковые результаты, а так как кластеризация – это обучение без учителя, то и истинных значений не предполагается. Помимо этого, почти всегда возникает вопрос о количестве кластеров, поэтому без участия человека в том или ином виде, кластеризация на нынешнем этапе своего развития обойтись не может.

Вопрос существования универсального алгоритма кластеризации беспокоит человечество еще со времен появления первых алгоритмов, решающих эту задачу. Джон Клейнберг сформулировал набор требований, которым мо-

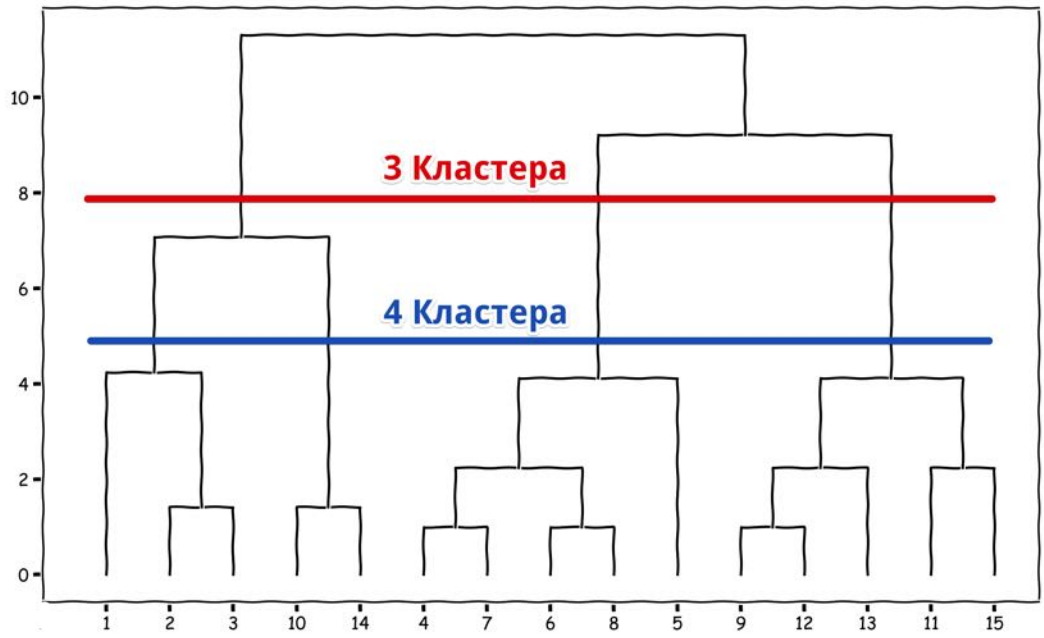

Рис. 16: Определение количества кластеров

гут удовлетворять или не удовлетворять алгоритмы кластеризации:

1. **Масштабно-инвариантность.** Для произвольной функции расстояния d и для любого числа $\alpha > 0$ результаты кластеризации с использованием расстояний d и $d \cdot \alpha$ совпадают.

   Требование заключается в том, что алгоритм кластеризации не должен быть чувствителен к изменениям в единицах измерения расстояния – он не должен иметь встроенной «шкалы длин».

   Графически эта идея проиллюстрирована на рисунке 19. Слева представлен исходный набор объектов, справа он же, но «сжатый» вдвое по оси абсцисс.

2. **Полнота.** Множество результатов кластеризации алгоритма A в зависимости от изменения функции расстояния d должно совпадать со множеством всех возможных разбиений множества объектов X.

   Иными словами, выбрав любое разбиение исходного множества, найдется такая функция расстояния, используя которую, алгоритм произведет желаемое разбиение.

3. **Согласованность.** Результат кластеризации не изменяется после допустимого преобразования функции расстояния. Пусть d и d' – две

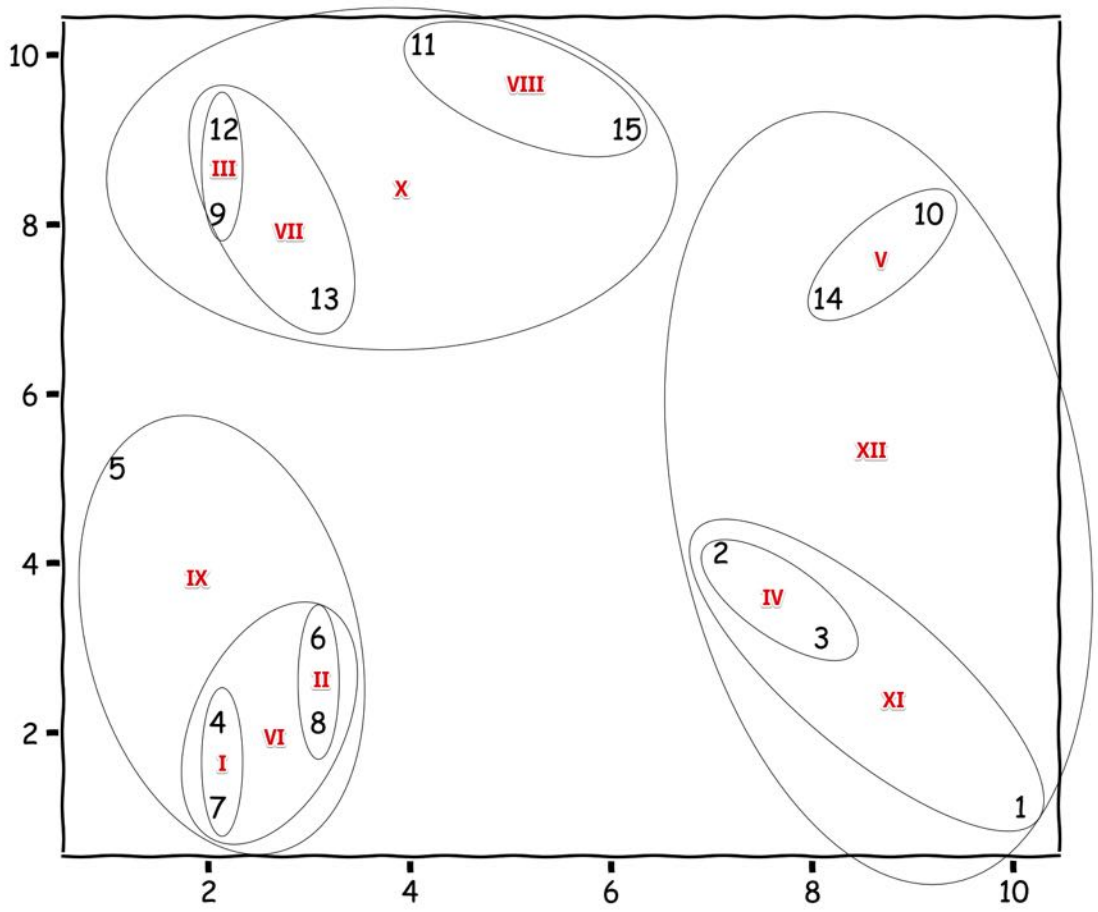

Рис. 17: Этапы объединения кластеров

функции расстояния, тогда d' называется допустимым преобразованием d, если:

- $d'(x_i, x_j) \leqslant d(x_i, x_j)$, если x_i и x_j лежат в одном кластере;
- $d'(x_i, x_j) \geqslant d(x_i, x_j)$, если x_i и x_j лежат в разных кластерах.

Иными словами, допустимое преобразование – это такая замена функции расстояния, при которой расстояние между двумя объектами, принадлежащими одному кластеру не увеличивается, а расстояние между двумя объектами, принадлежащими разным кластерам, не уменьшается. Иллюстрация этой идеи представлена на рисунке 20. Слева представлен исходный набор объектов, а справа объекты располагаются в два раза ближе к своим центроидам.

Клейнберг сформулировал и доказал следующую теорему:

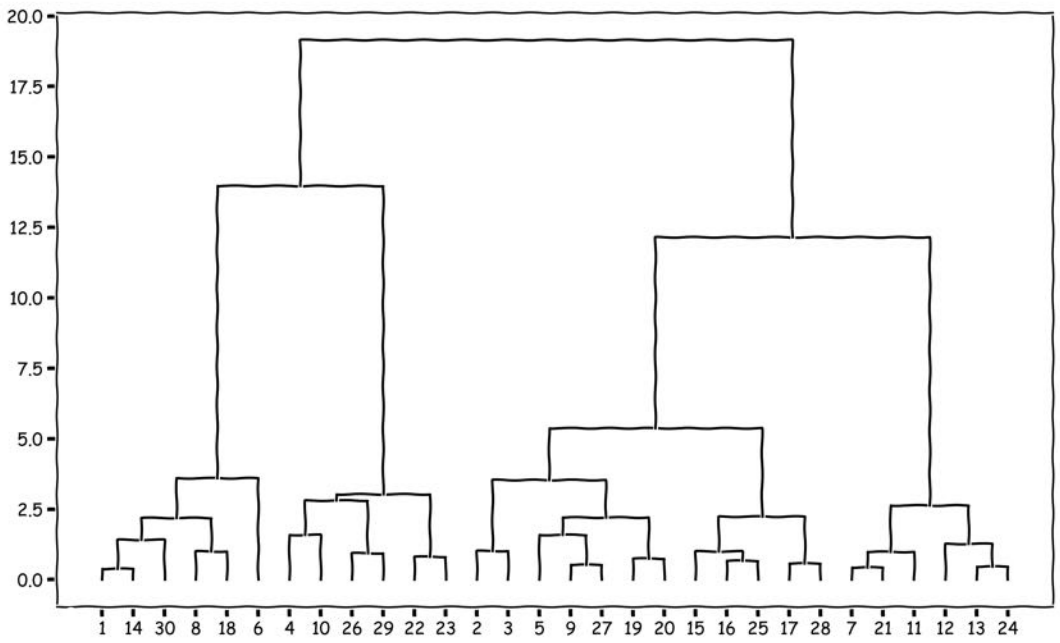

Рис. 18: Выбор подходящего числа кластеров

Теорема 5.0.1 (Клейнберга) *Для любого множества, состоящего из* $n \geq 2$ *объектов, не существует такого алгоритма кластеризации, обеспечивающего одновременно полноту, согласованность и масштабно-инвариантность.*

Доказательство в рамках этого курса мы опустим. Эта теорема ограничивает возможности кластеризации, так как говорит о том, что оптимального алгоритма кластеризации не существует. С другой стороны, раз совершенство недостижимо, нужно работать над улучшением существующих решений и изобретать новые, более эффективные алгоритмы.

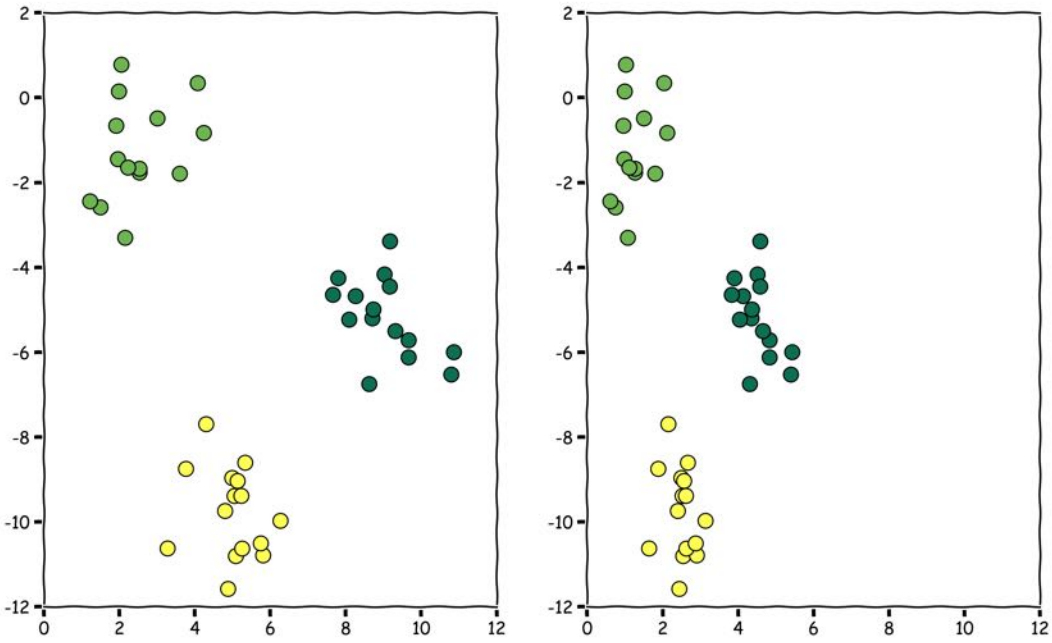

Рис. 19: Масштабно инвариантность

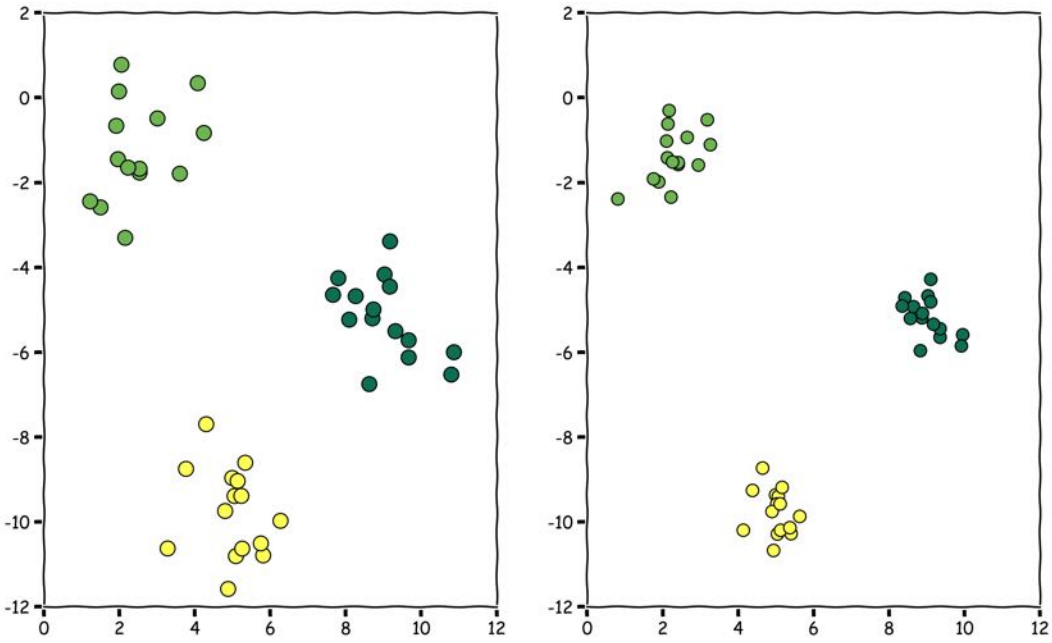

Рис. 20: Согласованность