Настройка параметров кластеризации DBSCAN по значениям метрики силуэта

В. Г. Мосин

Аннотация

В настоящей статье рассматривается метод автоматической настройки параметров кластеризации DBSCAN, основанный на авторском алгоритме определения локтевой точки метрики силуэта. Излагается алгоритм настройки и его приложения к наборам синтетических данных.

Содержание

1 Введение **2**
- 1.1 Теоретическая часть . . . 2
  - 1.1.1 Кластеризация DBSCAN . . . 2
  - 1.1.2 Метрика силуэта . . . 3
- 1.2 Постановки задачи . . . 3
  - 1.2.1 Предмет исследования . . . 3
  - 1.2.2 Методика исследования . . . 3
  - 1.2.3 Цель исследования . . . 4
- 1.3 Библиотеки . . . 4

2 Функция `fold_point` **4**

3 Алгоритм **6**
- 3.1 Генерация синтетических данных . . . 6
- 3.2 Двумерная сетка метрики силуэта . . . 6
- 3.3 Таблица метрик . . . 7
- 3.4 Вложенный цикл по параметрам `eps` и `min_samples` . . . 7
  - 3.4.1 Стартовая итерация . . . 7
  - 3.4.2 Внутренний цикл по `min_samples` . . . 8
  - 3.4.3 Внешний цикл по eps . . . 8
- 3.5 Редукция таблицы результатов . . . 9
- 3.6 Лучшая строка в таблице результатов . . . 9

3.7 Локтевая точка на лучшей кривой 10
3.8 Оптимальные параметры кластеризации 11
3.9 Оптимальная кластеризация 11

4 Результаты **11**
4.1 Четкое разбиение 11
4.2 Разбиение с высокой долей шума 12

5 Выводы **12**

6 Литература **13**

1 Введение

Кластеризация является одним из основных методов анализа данных, который позволяет разделить множество объектов на группы, называемые кластерами, таким образом, чтобы объекты внутри одного кластера были более похожи друг на друга, чем на объекты из других кластеров. Этот подход играет важную роль в различных областях, таких как машинное обучение, статистика и многих других (см. [4]).

Верный подбор параметров кластеризации является критически важным этапом процесса анализа данных. Параметры кластеризации определяют, какие именно свойства объектов будут учитываться при формировании кластеров. Неправильный выбор параметров может привести к некорректному образованию кластеров, а значит, к неверной интерпретации данных и неправильным выводам.

1.1 Теоретическая часть

В нашей работе мы проанализируем один из методов автоматической настройки параметров кластеризации DBSCAN, используя в качестве метрики ее эффективности метрику силуэта.

1.1.1 Кластеризация DBSCAN

Кластеризация DBSCAN (Density-Based Spatial Clustering of Applications with Noise) — это алгоритм машинного обучения, который используется для группировки данных на основе их плотности. В отличие от других методов кластеризации, таких как KMeans или агломеративная кластеризация, алгоритм DBSCAN способен обнаруживать кластеры произвольной и сложной формы, а также находить шумовые точки, которые не принадлежат ни одному кластеру. DBSCAN является мощным инструментом для кластеризации данных и находит применение в различных областях, таких как анализ социальных сетей, геоинформационные

системы, анализ изображений и т. д. Его способность обнаруживать кластеры произвольной формы и игнорировать шумовые данные делает его очень полезным инструментом для извлечения информации из больших объемов данных.

1.1.2 Метрика силуэта

Метрика силуэта является одной из ключевых метрик, которая используется для оценки качества кластеризации. Она позволяет измерить, насколько хорошо объекты распределены по кластерам, и определить, насколько каждый объект подходит своему кластеру. Принцип работы метрики силуэта основывается на оценке плотности и компактности кластеров. На высокий показатель силуэта влияют два фактора: среднее расстояние между объектом и остальными объектами в его кластере (компактность), а также среднее расстояние до объектов, принадлежащих другим кластерам (разделенность). Чем больше величина силуэта, тем лучше кластеризация.

Одним из основных преимуществ метрики силуэта является ее универсальность. В отличие от других метрик, она может быть применена к различным типам данных и алгоритмам кластеризации. Эта метрика не требует априорного знания количества кластеров, что является ее значительным преимуществом, а в применении к кластеризации DBSCAN, для которой число кластеров заранее неизвестно, метрика силуэта является незаменимой.

1.2 Постановки задачи

Когда исследователь применяет метод локтя, его действия состоят в визуальной оценке графика изучаемой метрики. За этими красивыми словами кроется тот факт, что исследователь определяют локтевую точку на глаз. И хотя при этом используются более корректные слова — визуально или интуитивно, сути это не меняет.

1.2.1 Предмет исследования

Ключевым фрагментом нашего алгоритма настройки параметров кластеризации DBSCAN является функция автоматического определения локтевой точки (см. п. 2). Мы будем исследовать работу этой функции.

1.2.2 Методика исследования

Мы реализуем алгоритм настройки и протестируем его на достаточном количестве синтетических данных.

1.2.3 Цель исследования

Наша цель — выяснить, является ли предложенный нами метод автоматической настройки параметров кластеризации DBSCAN самостоятельным рабочим инструментом, не требующим вмешательства человека, и, по возможности, установить границы применения этого метода.

1.3 Библиотеки

Для выполнения вычислений и анализа данных мы пользуемся средой `Jupyter Notebook`, которая предоставляет удобные средства для работы с языком программирования Python и его главными библиотеками: `NumPy`, `Pandas`, `sklearn` и `matplotlib`. Благодаря этим инструментам, мы можем эффективно работать с данными, выполнять исследования и визуализировать результаты (см. [1], [2]).

2 Функция `fold_point`

В теле алгоритма (см. п. 3) мы осуществим прогон кластеризации DBCSAN по двумерной сетке ее двух основных параметров:

1. `eps` — радиус, внутри которого точки облака данных учитываются, как принадлежащие одному кластеру;
2. `min_samples` — число точек, начиная с которого фрагмент облака может считаться самостоятельным кластером.

При этом для оценки качества кластеризации мы будем использовать метрику силуэта `silhouette_score`, которая принимает значения из промежутка [–1, 1], и качество кластеризации оценивается тем выше, чем больше значение `silhouette_score`-метрики.

Однако при таком вполне очевидном подходе возникает одна из существенных проблем подбора параметров кластеризации DBCSAN. Она состоит в том, что лучшая кластеризация далеко не всегда соответствует лучшему значению метрики.

Максимальное значение метрики силуэта (см. p. 1(b)) может привести к тому, что все объекты будут объединены в один кластер (плюс, возможно, некоторое количество выбросов), а это не самая лучшая кластеризация, так как она вообще не улавливает каких-либо различий между объектами. При этом разумное разделение на кластеры происходит при так называемом локтевом значении метрики (см. p. 1(a)), которое не определяется математически. Это не точка экстремума и не точка перегиба (хотя иногда в литературе ее называют именно так, что является грубейшей ошибкой), это не точка максимальной кривизны и т. д. Локтевое значение метрики требует интуиции и опыта исследователя.

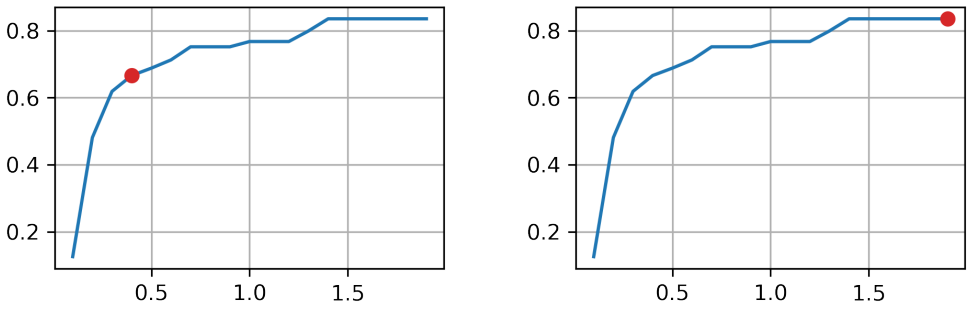

Рис. 1: Локтевое значение метрики и ее максимальное значение

Обычно говорят, что локтевая точка — это точка, после прохождения которой уже не происходит существенного изменения метрики. Не претендуя на строгое определение локтевой точки (которое, видимо, невозможно), мы предлагаем собственный метод ее нахождения. Мы хотим избавиться от человеческого фактора в процессе локализации локтевой точки и исключить те самые «интуицию и опыт исследователя», а наоборот: дать формальный автоматический алгоритм, возвращающий локтевую точку.

Наша идея состоит в том, чтобы осуществить ансаблирование одномерной линейной регрессионной модели путем разбиения промежутка изменения предиктора метрики силуэта на два интервала (см. [5], [6]). Напомним, что оценка линейной регрессионной модели проводится при помощи коэффициента детерминации $R^2 = 1 - S^*/S'$, где:

1. S^* — сумма квадратов отклонений прогнозируемых значений целевой функции от ее истинных значений,
2. S' — сумма квадратов отклонений истинных значений целевой функции от ее среднего значения.

Если разбить промежуток изменения предиктора на два участка, построить на каждом из них линейную модель и вычислить совокупный коэффициент детерминации для ансамбля из двух моделей, то суммарное отклонение в знаменателе останется точно таким же, но суммарное отклонение в числителе станет меньше (см. рис. 2).

Это приводит к повышению коэффициента детерминации (см. [6]).

Теперь, если выполнить цикл по одномерной сетке, разбивающей промежуток изменения предиктора, получим список коэффициентов детерминации возникающих при этом ансамблей, среди которых окажется какой-то максимальный элемент. Индекс этого элемента списка мы и примем в качестве абсциссы локтевой точки (еще раз повторим, что это не определение локтевой точки, а рабочий метод ее нахождения).

Подробное описание алгоритма функции, определяющей локтевую точку, выходит за рамки настоящей статьи, которая посвящена не локте-

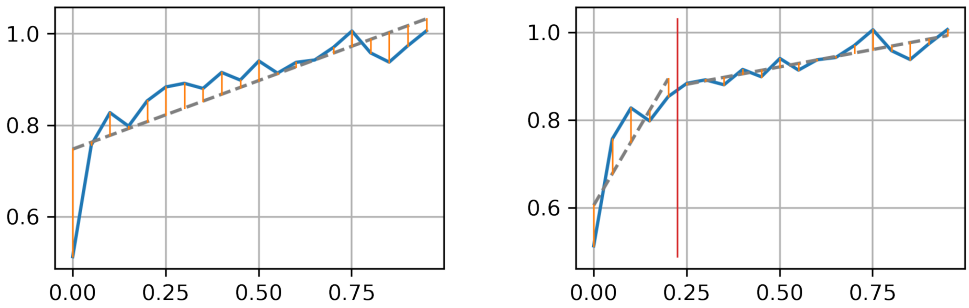

Рис. 2: Линейная модель (a) и ансамбль из двух линейных моделей (b)

вым точкам, а кластеризации DBSCAN, поэтому мы ограничимся лишь указанием ее основных характеристик:

1. имя функции — `fold_point`;
2. аргументы функции — пара одномерных массивов `A`, `B`, обладающих равной длиной, причем, первый массив содержит значения горизонтальной переменной, а второй — вертикальной;
3. в теле функции реализован алгоритм, основанный на изложенной выше идее ансаблирования линейной регрессионной модели для метрики силуэта;
4. функция возвращает номер элемента массива `A`, в котором фиксируется локтевое значение массива `B`.

3 Алгоритм

3.1 Генерация синтетических данных

Для реализации нашего алгоритма настройки кластеризации DBCSAN мы используем синтетические данные. Мы импортируем специальный модуль `datasets` библиотеки `sklearn`, после чего нам становится доступен метод `make_blobs`, при помощи которого мы генерируем двумерные данные. На старте алгоритма мы создаем облако из 500 точек и записываем его в двумерный массив `X`.

3.2 Двумерная сетка метрики силуэта

Сетка для вложенного цикла должна учитывать специфику параметров `eps` и `min_samples`. Параметр `eps` задает радиус, внутри которого рассчитывается плотность облака, то есть по своей природе это непрерывный параметр, мы будем варьировать его в пределах от 0.1 (так как нулевой радиус не имеет смысла) до 2 с шагом 0.1. Для задания сетки по параметру `eps` мы применяем метод `arange` библиотеки `numpy`:

```
eps_grid = np.arange(0.1, 2, 0.1)
```

Второй параметр `min_samples` задает количество объектов, то есть по своей природе является натуральным числом, и мы варьируем его в пределах от 1 (но не от 0) до 15 с шагом 1. Для задания сетки по параметру `min_samples` мы также применяем метод `arange` библиотеки `numpy`:

```
min_samples_grid = np.arange(1, 15, 1)
```

3.3 Таблица метрик

Мы будем обучать модель кластеризации DBCSAN для каждой пары значений параметров `eps` и `min_samples` и вычислять соответствующую этим значениям величину метрики `silhouette_score`. Таким образом, все возможные значения метрики силуэта, которые возникнут по результатам вложенного цикла по сетке, естественным образом заполнят таблицу с двумя входами: столбцы таблицы будут нумероваться значениями `eps`, а строки — значениями `min_samples`.

В начале работы алгоритма, еще до исполнения цикла по сетке мы вызываем метод `DataFrame` библиотеки `pandas`, передаем ему в качестве значения параметра columns значение переменной `eps_grid`, а в качестве значения параметра `index` — значение переменной `min_samples_grid`. Результат записываем в переменную `result`:

```
result = pd.DataFrame(columns=eps_grid, index=min_samples_grid)
```

Таким образом, в начале алгоритма мы имеем датафрейм `result`, который полностью заполнен значениями `NaN`. Далее, для сокращения кода, а также для того, чтобы подчеркнуть табличную природу элементов сетки, вводим две переменные:

```
cols = result.columns
ind = result.index
```

Это два массива, по элементам которых и будет осуществляться вложенный цикл.

3.4 Вложенный цикл по параметрам `eps` и `min_samples`

3.4.1 Стартовая итерация

Присваиваем параметрам `eps` и `min_samples` стартовые значения массивов `cols` и `ind` и, при помощи метода `DBCSAN` из модуля `clusters` библиотеки `sklearn` получаем модель кластеризации `clust`:

```
clust = DBSCAN(eps=cols[0], min_samples=ind[0])
```

Обучаем модель на массиве `X` при помощи метода `fit` и выводим метки кластеров в список `labels` при помощи метода `labels_`:

```
clust.fit(X)
labels = clust.labels_
```

Далее нам нужно учесть ограничения метрики силуэта. Дело в том, что ее невозможно вычислить, если по результатам кластеризации с данными значениями параметров возникли следующие ситуации:

1. все объекты оказались объединены в единственный кластер (в этом случае отсутствует межкластерное различие),
2. каждый объект оказался выделен в самостоятельный кластер (в этом случае отсутствует внутрикластерное сходство).

Поэтому для бесперебойной работы цикла при заполнении таблицы `result` должно выполняться условие: число получившихся кластеров больше 1 и меньше числа объектов в массиве `X`. Так как число кластеров — это число уникальных элементов в списке `labels`, мы применяем условный оператор `if` к следующему логическому выражению:

```
if (len(set(labels)) > 1) and (len(set(labels)) < len(X))
```

Если это условие выполняется, то в датафрейм `result` на место, расположенное в столбце с именем `cols[0]` и строке с индексом `ind[0]` записываем метрику силуэта, вычисленную при данных значениях параметров `eps` и `min_smples`. Для этого применяем метод `silhouette_score` из модуля `metrics` библиотеки sklearn к паре `(X, labels)` и выполняем присваивание:

```
result[cols[0]][ind[0]] = silhouette_score(X, labels)
```

Напомним, что изначально датафрейм `result` заполнен значениями `NaN`. Поэтому, если условие присваивания не выполняется, то на соответствующей позиции остается `NaN`, если же оно выполняется, позиция заполняется значением метрики силуэта.

3.4.2 Внутренний цикл по `min_samples`

Переходим к следующему значению массива `ind` и повторяем действия до исчерпания всех элементов массива. В результате в датафрейме `result` получаем заполненный первый столбец (за исключением тех позиций, для которых условие вычисления метрики силуэта оказалось невыполненным).

3.4.3 Внешний цикл по `eps`

Переходим к следующему значению массива `cols` и повторяем действия до исчерпания всех элементов массива. В результате получаем запол-

ненный датафрейм `result` (за исключением тех позиций, для которых условие вычисления метрики силуэта оказалось невыполненным).

3.5 Редукция таблицы результатов

Итак, на этом шаге мы имеет таблицу `result`, заполненную метриками силуэта, вычисленными для различных значений параметров `eps` и `min_samples`. Но некоторые позиции этой таблицы могут быть не заполнены в силу указанных выше причин. Мы удаляем из таблицы `result` все столбцы, в которых встречается хотя бы одно значение `NaN`. Для этого мы транспонируем датафрейм `result`, применяем метод `dropna` из библиотеки pandas и снова транспонируем то, что получилось:

```
result = result.T.dropna().T
```

Теперь в таблице `result` заполнены все поля, хотя количество столбцов в ней, скорее всего, уменьшилось.

3.6 Лучшая строка в таблице результатов

Визуально строки таблицы `result` представляют собой семейство кривых, где по горизонтали откладывается параметр `eps` (при фиксированном значении параметра `min_smples`), а по вертикали — значение метрики силуэта (см. рис. 3(a)).

Если усреднить значения каждой такой кривой по всем возможным значениям горизонтальной переменной, то получится семейство горизонтальных линий, среди которых какая-то одна линия окажется самой верхней. Это будет означать, что соответствующая кривая по совокупности всех возможных значений горизонтальной переменной демонстрирует наилучший результат (см. рис. 3(b)), в качестве которого в данном исследовании мы используем метрику силуэта.

На этом шаге мы составляем список средних значений по всем кривым, описывающим поведение метрики силуэта в зависимости от параметра `eps`. Для этого транспонируем датафрейм `result`, применяем к транспонированному датафрейму метод `mean` билиотеки `pandas`, а полученную серию переводим в список при помощи метода `list`. Получаем список усредненных значений метрики силуэта, отвечающий всем фиксированным значениям параметра `min_samples`:

```
min_samples_means_list = list(result.T.mean())
```

Затем к списку `min_samples_means_list` применяем метод `index` и получаем индекс максимального элемента этого списка, который записываем в переменную `min_samples_best`:

```
min_samples_best =
min_samples_means_list.index(max(min_samples_means_list))
```

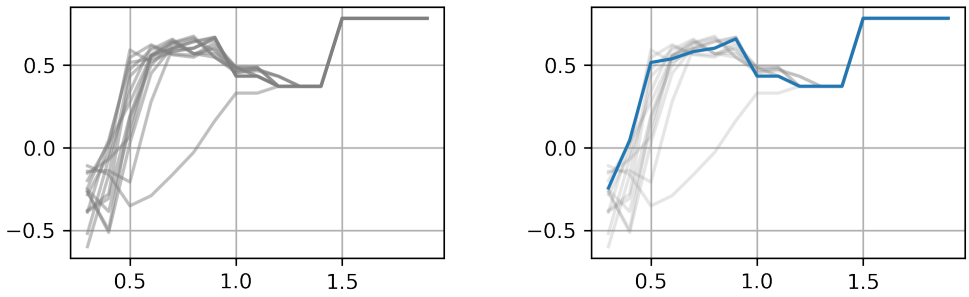

Рис. 3: Семейство кривых и лучшая кривая

На рисунке 3(b) изображена кривая, которая получается извлечением из таблицы `result` строки с индексом `min_samples_best`.

3.7 Локтевая точка на лучшей кривой

Мы уже отмечали что наилучшая кластеризация не всегда соответствует лучшей метрике силуэта, и что для определения оптимальных параметров кластеризации нужно использовать локтевое значение метрики (см. рис. 4(b)).

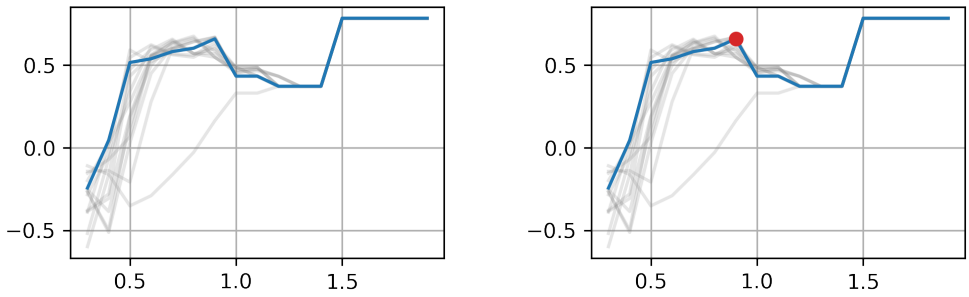

Рис. 4: Лучшая кривая и локтевая точка на лучшей кривой

На этом шаге мы применяем функцию `fold_point`. Создаем массив предиктора:

```
A = np.array(result.columns)
```

Напомним, что в качестве горизонтальной переменной выступает параметр `eps`, то есть, массив `A` — это значения горизонтальной переменной для лучшей кривой метрики силуэта (см. рис. 4(b)). Далее, создаем массив функции, анализируемой на предмет локтевой точки:

```
B = result.iloc[min_samples_best].to_numpy()
```

Массив `B` — это лучшая строка в таблице результатов, переведенная в формат массива `numpy` при помощи метода `to_numpy`. Визуализация

массивов A и B дает изображение лучшей кривой из семейства кривых метрики силуэта (см. рис. 4(b)).

После этого применяем функцию fold_point к паре массивов A и B. Получаем номер предиктора, при котором наблюдается локтевое значение метрики, и записываем его в переменную fold_ind:

```
fold_ind = fold_point(A, B)
```

3.8 Оптимальные параметры кластеризации

Выше мы получили лучшее значение для параметра min_samples и записали его в переменную min_sample_best. Теперь, пользуясь найденным номером локтевой точки, мы находим оптимальное значение параметра eps и записываем его в переменную eps_best:

```
eps_best = result.columns[fold_ind]
```

3.9 Оптимальная кластеризация

Выяснив, при каких значениях параметров eps и min_samples кластеризация DBSCAN дает приемлемые в смысле метода локтя результаты, проводим кластеризацию исходного множества X с найденными значениями параметров:

```
clust = DBSCAN(eps=eps_best, min_samples=min_samples_best)
clust.fit(X)
```

4 Результаты

Одной из интересных особенностей кластеризации DBSCAN является то, что точки, которые с ее точки зрения не могут принадлежать никакому кластеру, выделяются в отдельный набор точек: в набор выбросов. Мы провели множественные эксперименты на синтетических данных и обнаружили, что доля выбросов в кластеризации DBSCAN связана с формой кривой метрики силуэта, соотнесенной с радиусом кластеризации (то есть, с величиной параметра eps). Заметим, что такой же связи с кривой метрики силуэта, соотнесенной с параметром min_samples, мы не обнаружили.

4.1 Четкое разбиение

Если кривая метрики силуэта, соотнесенной с радиусом кластеризации, обладает ярко выраженной локтевой точкой, то, как правило, определенные нашим алгоритмом оптимальные значения параметров eps=eps_best

и `min_samples=min_samples_best` дают четкое разбиение на кластеры с незначительной долей выбросов (см. рис. 5).

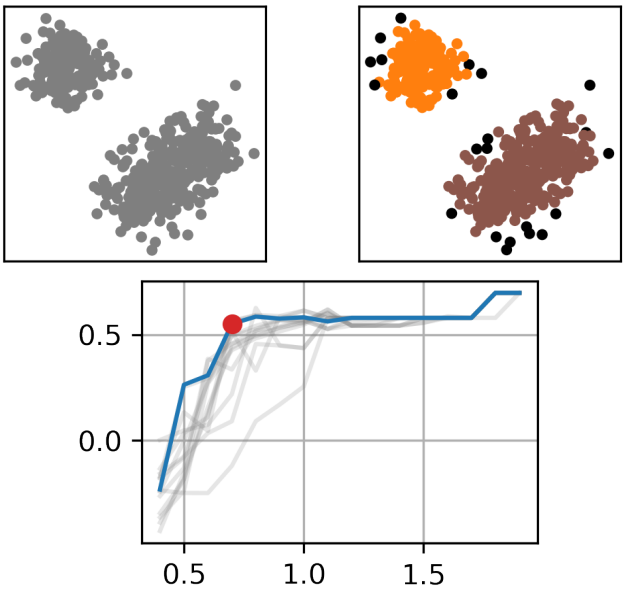

Рис. 5: Четкое разбиение

Это явление наблюдается не всегда, то есть, локтевая точка может быть выраженной ярко, но, тем не менее, разбиение на кластеры будет нечетким. Однако такая ситуация встречается не часто, и в случае ярко выраженной локтевой точки характерным является именно четкое разбиение, проиллюстрированное рисунком 5.

4.2 Разбиение с высокой долей шума

Ели же локтевая точка на кривой метрики силуэта, соотнесенной с радиусом кластеризации, выражена неярко, то, как правило, после проведения кластеризации с найденными значениями параметров `eps=eps_best` и `min_samples=min_samples_best` получается нечеткое разбиение с большой долей выбросов (см. рис. 6).

5 Выводы

Итак, есть ручная настройка параметров кластеризации DBCSAN, основанная на визуальном анализе кривой метрики силуэта, и есть автоматическая настройка, основанная на модельном подходе к определению локтевой точки.

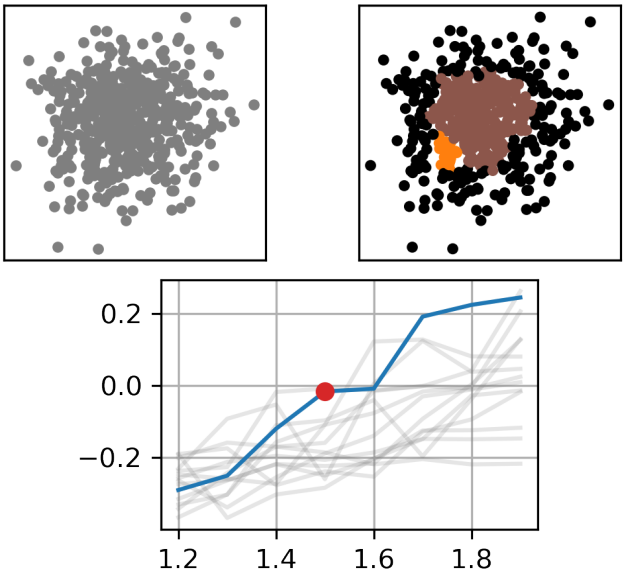

Рис. 6: Разбиение с высокой долей шума

Вне всяких сомнений, ручная настройка точнее отражает специфику данных и дает исследователю возможность выполнить кластеризацию, отталкиваясь от природы данных и особенностей задач, решаемых за счет кластеризации. Однако предложенная нами автоматическая настройка обладает как минимум двумя преимуществами:

1. она выполняется без участия аналитика, что, безусловно, играет решающую роль в ситуациях, когда кластеризация должна осуществляться многократно и/или в режиме реального времени
2. она не допускает грубых ошибок, то есть, предлагая, возможно, не самый лучший вариант кластеризации, она никогда не предлагает неприемлемых вариантов.

В целом, изложенный в настоящей статье алгоритм автоматической настройки параметров кластеризации DBSCAN продемонстрировал хорошие результаты и может быть рекомендован как один из рабочих инструментов анализа данных.

6 Литература

1. Хейдт М. Изучаем Pandas / М. Хейдт; — Москва: ДМК Пресс, 2018. — 438 с.
2. Бурков А. Машинное обучение без лишних слов / А. Бурков; — СПб: Питер, 2020. — 192 с.

3. Вьюгин, В. В. Математические основы теории машинного обучения и прогнозирования / В. В. Вьюгин; — М.: МЦИМО. — 2013. — 387 с.
4. Бринк Х. Машинное обучение / Х. Бринк, Дж. Ричардс, М. Феверолф — СПб.: Питер, 2017. — 336 с.
5. В. Г. Мосин, А. В. Караваев. О некоторых проблемах моделирования измеряемых социально-психологических переменных. Математическое образование в современном мире: теория и практика : сборник статей // Самарский государственный технический университет. Всероссийская научно-методическая конференция с международным участием (28-–30 ноября 2022 г; Самара); ред. О. В. Юсупова. — Самара, 2022. — 175 с.
6. Мосин В.Г. Линеаризация целевой функции в регрессионных задачах методом сингулярных разложений // В книге: Математическое моделирование. Тезисы II Международной конференции. Москва, 2021. С. 66–67.