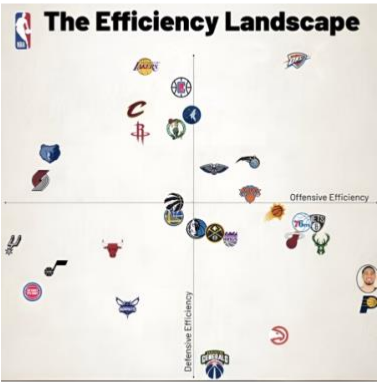
Обзор алгоритма **OPTICS**

Ковыляев Александр БПМИ228-1

Введение

В наше время множеством ресурсов в различных целях собираются огромные объёмы данных. Для извлечения из них полезной информации существует много методов анализа данных и статистической обработки, одним из которых является кластерный анализ. Это процедура разделения множества на группы схожих объектов. Это может быть полезно как для последующего анализа каждой группы в отдельности, так и для общего выявления структуры в данных и нахождения скрытых закономерностей. Определим, что каждый объект задаётся набором параметров, например рубашка – размером, цветом, типом ткани и т.д. Тогда каждый объект можно представить точкой в пространстве параметров, где каждое измерение – одна характеристика. Пример: команды НБА на графике защитной и атакующей эффективности - изображение 1.



Изображение 1
Каждая команда характеризуется двумя числами, соответствующими защитной и атакующей эффективностями.

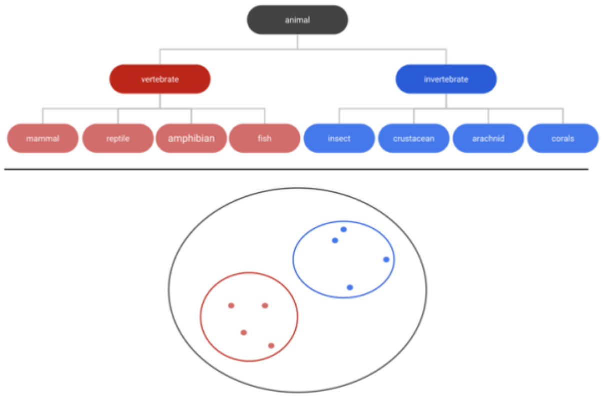
Количество характеристик – это размерность пространства. Схожесть объектов измеряется расстоянием между ними. Чем более похожи два объекта, тем ближе они в пространстве своих параметров, и наоборот, чем они менее похожи, тем дальше. Расстояние может задаваться различными метриками: евклидово расстояние, Манхэттенское расстояние и т.д.

Кластерный анализ используется и как самодостаточный метод анализа, и как один из шагов предобработки данных. Кластером можно назвать область в пространстве, в которой объекты расположены плотно, и которая окружена менее плотными областями.

Аналоги

Кластерный анализ извлекает очень ценную информацию о структуре расположения объектов в данных, из-за чего было придумано много различных алгоритмов кластеризации.

Алгоритмы кластеризации могут быть разделены на 2 типа: иерархические и плотностные. Иерархические определяют весь датасет в 1 кластер, и на каждом уровне разделяют его на меньшие, получая в итоге дендрограмму. Изображение 2.



Изображение 2
Иерархическая структура на примере животных.

Плотностные используют локальный критерий кластера (далее - критерий основной точки). Большинство алгоритмов кластеризации имеют такие недостатки как ухудшение работы при увеличении размерности пространства, большое количество параметров для корректной работы, проблема глобальной плотности. Результаты работы иерархических алгоритмов сложно анализировать при увеличении числа объектов, а метод разделения пространства на отдельные участки и применение разных параметров на разных участках плохи большими затратами памяти и неочевидностью способов дальнейшего анализа.

Рассматриваемый в данной работе метод OPTICS основан на алгоритме DBSCAN и лишён многих недостатков других методов, поэтому сначала рассмотрим предшественника - DBSCAN.

DBSCAN

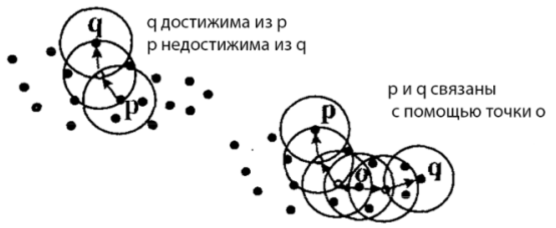
Параметры метода: радиус - eps и минимальное количество соседей - MinPts.

Этот алгоритм относится к плотностным, поэтому его ключевая идея состоит в том, чтобы каждый объект кластера содержал в своей eps-окрестности как минимум MinPts объектов (включая себя) – критерий основной точки.

Далее в работе объект отождествляется с точкой. Также будут использоваться термины и примеры только для двумерного пространства, но всё сказанное можно распространить и на многомерный случай.

Введём несколько определений:

1. Точка **p** – *основная точка*, если на расстоянии не большем eps (в eps-окрестности) находится по меньшей мере MinPts её соседей, включая саму **p**. Это равносильно условию $|N_{eps}(\mathbf{p})|$ >= MinPts, где | | - количество объектов в множестве, а $N_{eps}(\mathbf{p})$ – множество соседей из eps-окрестности точки **p**. Точка **p** может называться ядерной, а условие – ядерным условием.
2. *Соседи* точки **p** – все точки, принадлежащие $N_{eps}(\mathbf{p})$.
3. Точка **q** *напрямую достижима* из **p**, если **p** основная точка, и **q** принадлежит $N_{eps}(\mathbf{p})$.
4. Точка **q** *достижима* из **p**, если существует последовательность точек, начинающаяся с **p** и оканчивающаяся **q**, такая что каждая точка, кроме p, *напрямую достижима* из предыдущей. Изображение 3.
5. Точки **p** и **q** *связаны*, если существует точка **o**, из которой достижимы и **p**, и **q**. Изображение 3.



Изображение 3

6. *Кластер* – это непустое подмножества всех точек, удовлетворяющее условиям:
   a. *Максимальность*: если точка **p** принадлежит кластеру, и **q** *достижима* из **p**, то **q** тоже принадлежит этому *кластеру*.
   b. *Связанность*: две любые точки, принадлежащие *кластеру, связаны*.
7. *Шум* – все точки, не относящиеся ни к какому кластеру.

Только из основных точек другие могут быть достижимы. Кластер состоит из основных точек и граничных – достижимых из основных. Изображение 4.

Изображение 4
Все точки на картинке принадлежат кластеру.
Красные – основные точки, серые – граничные.

В каждый кластер входит как минимум одна основная точка.

Описание алгоритма DBSCAN:

1. Выбирается случайная, ещё не обработанная точка **p.**
2. Если выполняется ядерное условие, то точка присваивается новому кластеру $\mathbf{C_i}$, иначе помечается как шум.
3. В множество **M** записываются все соседи **p**.
4. Перебираются все точки **m** из множества **M**:
   a. **m** присваивается кластеру $\mathbf{C_i}$.
   b. Если **m** – основная точка, то все, ещё необработанные точки из её eps-окрестности, добавляются в множество **M**.

Алгоритм действует изнутри кластера наружу, сначала определяя основную точку как новый кластер, а затем расширяется от неё к границам кластера. GIF-Изображение 5.

GIF-Изображение 5
Работа алгоритма DBSCAN с разными параметрами.

На втором шаге представленного алгоритма точка, помеченная как шум, позднее может быть добавлена в кластер, если она окажется в eps-окрестности какой-то основной точки.
DBSCAN в общем случае имеет среднюю сложность O(n log n) и в худшем случае O(n^2). Также требует O(n) памяти.

Преимущества DBSCAN:

1. Не требует обозначения числа кластеров, на которые нужно разделить данные.
2. Может находить кластеры произвольной формы (в отличие от некоторых других алгоритмов, например k-means).
3. Хорошая скорость работы – O(n log n).

Недостатки:

1. Неоднозначность – точки, находящиеся на границе двух кластеров, могут попасть как в один из них, так и в другой в зависимости от порядка рассмотрения точек.
2. Проблема глобально заданной плотности. Плотность – количество точек на единицу площади.

Главная слабость алгоритма – глобальная плотность. В большинстве датасетов, собранных на основе реальных данных существуют области с различными плотностями объектов. Т.е. в данных может быть кластер, состоящий из большого числа объектов, расположенных в небольшой окрестности, и одновременно с этим кластер, который состоит из меньшего числа точек на бОльшом расстоянии. Изображение 6.

Изображение 6
Разно-плотностные кластеры.

Из-за заданных для всего пространства параметров eps и MinPts DBSCAN может либо выделить обе этих группы в один кластер, либо выделить только более плотный кластер, а второй пометить как шум. Существует расширение DBSCAN, которое способно работать с несколькими плотностями одновременно (для одного значения MinPts несколько eps), для этого необходима очередь – условие, что сначала будет обрабатываться точка достижимая относительно минимального eps, т.е. алгоритм будет искать сначала более плотные кластеры, а затем менее плотные.

OPTICS

Алгоритм OPTICS основан на идее DBSCAN, но решает его главную проблему – избавление от глобальной плотности. OPTICS обрабатывает сразу все возможные плотности eps (для фиксированного параметра MinPts), ограниченные сверху значением max_eps. Ограничение нужно только для ускорения работы алгоритма, его можно задать максимально возможным значением.
Важно отметить, что результат работы OPTICS – это не разделение точек на кластеры. Итогом этого алгоритма является порядок обработки им точек и значение reachability-distance для каждой из них (также сохраняются и ядерные расстояние, но чаще всего они никак не используются). На основе этого можно построить разбиение множества на кластеры без больших затрат по времени и памяти, но также эти данные предоставляют больше возможностей для дальнейшего анализа. Так, например, порядок точек и значения reachability-distance могут быть использованы для:

- Разделения точек на кластеры с различными плотностями.
- Определение характера кластеризации, например, плотность каждого отдельного кластера.
- Нахождения обособленных объектов (это может быть более важной задачей, чем нахождение кластера близких объектов, например, среди всех финансовых

операций банка найти мошеннические) - такая задача и её решение подробно описано в статье «OPTICS-OF: Identifying Local Outliers», в которой по данным о расстояниях достижимости вычисляли коэффициент обособленности.

- Определения иерархической структуры данных.

OPTICS вводит несколько дополнительных понятий:

1. *Ядерное расстояние* (*core-distace или core*) точки **o**, заданное как

   - **Неопределено**, если **o** – не основная для любого eps' меньшего max_eps.

   - **eps'** – минимальное расстояние, такое что $|\mathrm{N}_{\mathrm{eps}'}(\mathbf{o})| >= \mathrm{MinPts}$.

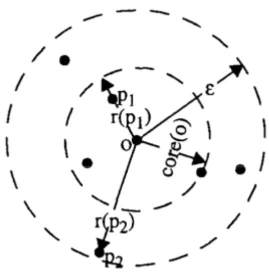
   Ядерное расстояние – минимальное расстояние, при котором точка становится основной. Далее будем считать, что значение “**неопределено**” больше любого другого.

2. *Расстояние достижимости* (*reachability-distance или r*) точки **p** относительно точки **o,** заданное как

   - **Неопределено**, если **o** – не основная при max_eps.

   - **max** {**core-distance**(**o**), **dist**(**o**, **p**)}, где $\mathrm{dist}(\mathbf{o}, \mathbf{p})$ – используемая функция расстояния между точками **o** и **p**.

Если точка **p** находится на расстоянии, меньшем core(**o**), то её **r**(**p**, **o**) = core(**o**). Это происходит из-за того, что если точка o – не основная, то никакая точка не может быть достижимой из неё.
Если же точка p находится на расстоянии, большем core(**o**), но всё ещё в пределах max_eps (верхнего ограничения), то **r**(**p**, **o**) = dist(**o**, **p**). Изображение 7 хорошо иллюстрирует это.



Изображение 7
core(o) – ядерное расстояние, r(p1), r(p2) – расстояния достижимости.

Расстояние достижимости можно также описать как *минимальное расстояние*, при котором точка **p** становится напрямую достижима из **o**.

Алгоритм работы OPTICS:

1. Открывается результирующий файл, строки которого можно представить в виде пар значений id точки и её расстояния достижимости.

2. Всем точкам задаётся значение reachability-distance – “**неопределено**”.

3. Все точки записываются в отсортированную по значению reachability-distance очередь.

4. Достаётся точка **o** из начала очереди:

   a. Точка **o** и её текущее расстояние достижимости записывается в результирующий файл.

   b. Для всех точек **p** из max_eps-окрестности **o** обновляется reachability-distance и соответственно обновляется очередь.

5. Закрывается результирующий файл.

На 4-ом шаге алгоритма достаётся точка **o** из начала очереди, поскольку очередь отсортирована, то это точка с наименьшим значением reachability-distance. Если у всех точек одинаковое значение **r**, то выбирается случайная точка.
На выходите получается файл, содержащий очерёдность точек, для каждой из которых записано минимальное расстояние достижимости относительно всех предыдущих точек. Для дальнейшего анализа полезно изобразить результирующий файл в виде гистограммы. Изображение 8.

Изображение 8
Несколько кластеров и гистограмма расстояний достижимости

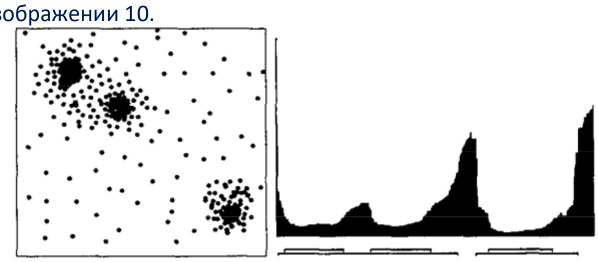
Из-за того, что reachability-distance связано с расстояниями до предшественников и из-за порядка выбора точек из очереди (с минимальным **r**), кластеры на графике достижимости обозначаются низменностями. В сущности, низкий показатель расстояния достижимости говорит о принадлежности точки кластеру, а высокий – о принадлежности к шуму или о прыжке от одного кластера к другому.
На графике хорошо видно, что алгоритм чётко выделяет все кластеры (независимо от их плотности), также видна иерархическая структура, например, в правой области пространства, где один кластер содержит другие.
По построенной гистограмме можно выделить кластеры. Это достигается с помощью либо DBSCAN-подобного алгоритма A, т.е. алгоритма, выделяющего кластеры на основе одного любого eps небольшего, чем max_eps (при фиксированном MinPts)., либо алгоритма B, выделяющего все кластеры основываясь на дополнительном параметре $\xi \in [0,1)$.
Алгоритм A обрезает гистограмму по вертикальной оси по значению eps. Тогда за один проход A выделяет в отдельные кластеры точки, чьи значения ограниченны с каждой стороны и не выходят за границу сверху. Остальные точки помечаются как шум. Этот метод изображён на картинке 9 (обрезанное изображение 8).

Изображение 9
DBSCAN-подобный алгоритм выделяет кластеры A, B, C, D и E.

Результат работы алгоритма A идентичен результату обычного DBSCAN (при использовании одинаковых параметры MinPts и eps. max_eps задано бесконечностью), возможно, за исключением некоторых граничных точек (это может произойти из-за порядка обхода точек), однако количество таковых несущественно и не влияет на последующие шаги анализа.

Алгоритм B действует иначе, он определяет кластеры по ξ-наклонным областям. Это означает, что алгоритм считает отдельным кластером ту последовательность точек, что ограничена с каждой стороны ξ-наклонными областям, и у которой расстояние достижимости каждой точки из последовательности ниже, чем у границ. Низменность по краям ограничивается точками, значения которых в (1 - ξ) раз больше значений ближайших точек низменности. Подробнее о том, как рассчитываются точные границы, можно прочесть в статье [1] в разделе 4.3.1.

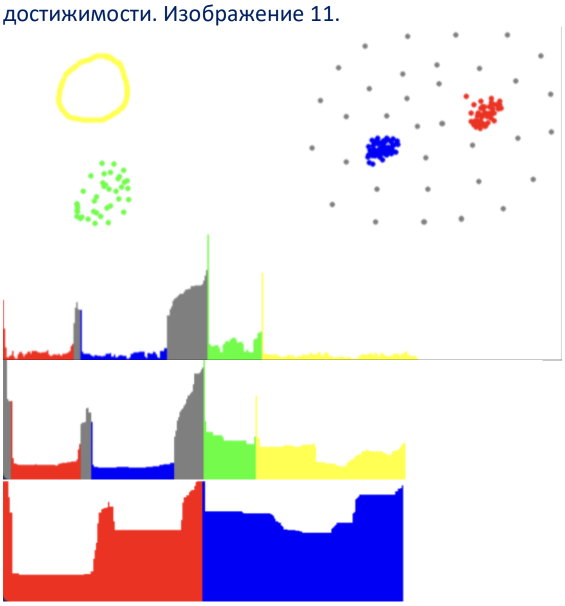
Пример иерархической структуры можно увидеть на изображении 10.



Изображение 10
Алгоритм успешно находит иерархическую структуру в виде двух кластеров и более верхне-уровневого кластера, содержащего их обоих, а также третьего кластера с областью повышенной плотности.

Параметры OPTICS

Параметр MinPts влияет на сглаженность графика достижимости. Изображение 11.



Изображение 11
Графики достижимости с параметром MinPts – 3, 18 и 55 сверху вниз соответственно. Метод кластеризации – ξ со значением 0.3.

Чем меньше значение параметра, тем более шумный график. Чем больше параметр, тем более сглаженный график, но из-за этого могут теряться детали. Также при больших значениях MinPts ослабляется возможный “single-link” эффект, когда несколько кластеров образуют один из-за тонкой линии точек, связывающих их.

Если параметр max_eps задать слишком маленьким, то вырастет скорость работы алгоритма, однако кластеры низкой плотности могут быть невидны.

Существует довольно большое количество оптимальных комбинаций параметров, при которых график отличается незначительно.

Сложность

Т.к. OPTICS структурно аналогичен DBSCAN, он имеет те же затраты по времени - в общем случае при применении ускоряющих индексных структур (например, R*-tree, X-tree или M-tree) и вне вырожденных данных имеет среднюю сложность O(n log n) и в худшем случае O(n^2). При проведении экспериментов видна зависимость времени исполнения OPTICS от времени исполнения DBSCAN с константным замедлением в 1.6 раз.

Сложность обоих алгоритмов главным образом зависит от времени возврата списка соседей из eps-окрестности, т.е. O(n * (время запроса eps-соседей)). Поэтому, если алгоритм имеет прямой доступ к eps-соседям, например, если объекты организованы в виде сетки, то скорость алгоритма достигает O(n), т.к. запрос соседей выполняется за O(1). Алгоритмы, распределяющие точки на кластеры по построенному графику достижимости, работают за линейное время, поэтому общее время работы алгоритмов OPTICS и кластеризации остаётся на уровне – O(n log n).

Преимущества OPTICS:

1. Не зависит от выбора глобальной плотности.
2. Работает за O(n log n).
3. Выделяет кластеры произвольной формы.
4. Подходит для выделения аномальных объектов – выбросов.
5. Подходит для иерархической кластеризации.

OPTICS не обладает какими-либо существенными недостатками.

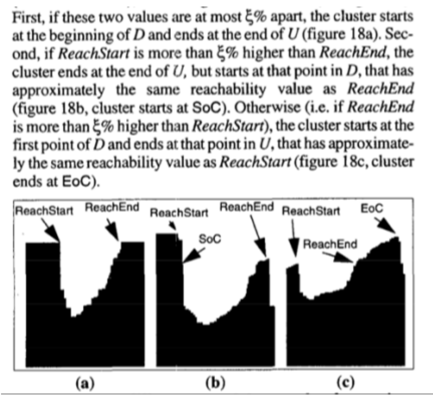
Видео:

Также прикрепляю видео с кратким объяснением OPTICS, красивыми графиками и инструкцией по эксплуатации прилагаемой программы для иллюстрации работы метода. Ссылка на видео.

Ошибки, найденные в статьях

Базовая статья [1]:

- Раздел 4.3.1 - перепутан ReachEnd и EoC в c).



First, if these two values are at most ξ% apart, the cluster starts at the beginning of *D* and ends at the end of *U* (figure 18a). Second, if *ReachStart* is more than ξ% higher than *ReachEnd*, the cluster ends at the end of *U*, but starts at that point in *D*, that has approximately the same reachability value as *ReachEnd* (figure 18b, cluster starts at SoC). Otherwise (i.e. if *ReachEnd* is more than ξ% higher than *ReachStart*), the cluster starts at the first point of *D* and ends at that point in *U*, that has approximately the same reachability value as *ReachStart* (figure 18c, cluster ends at EoC).

- Раздел 4.3.2 - в sc1* и sc2* должно быть так – max{r(x) | s_D < x < e_U} <= … . (пропущено r())

Condition 3b,
$\forall x, s_D < x < e_U : (r(x) \le min(r(s_D), r(e_U)) \times (1 - \xi))$, is equivalent to
(sc1) $\forall x, s_D < x < e_U : (r(x) \le r(s_D) \times (1 - \xi))$ $\wedge$
(sc2) $\forall x, s_D < x < e_U : (r(x) \le r(e_U) \times (1 - \xi))$,
so we can split it and check the sub-conditions (sc1) and (sc2) separately. We can further transform (sc1) and (sc2) into the equivalent condition (sc1*) and (sc2*), respectively:
(sc1*) $max\{x \mid s_D < x < e_U\} \le r(s_D) \times (1 - \xi)$
(sc2*) $max\{x \mid s_D < x < e_U\} \le r(e_U) \times (1 - \xi)$
In order to make use of conditions (sc1*) and (sc2*), we need to introduce the concept of maximum-in-between values, or mib-values, containing the maximum value between a certain point and the current index. We will keep track of one mib-value for each steep down region in SDASet, containing the maximum value between the end of the steep down region and the current index, and one global mib-value containing the maximum between the end of the last steep (up or down) region found and the current index.

Статья “*Fast Parameterless Density-Based Clustering via Random Projections*“ (Fast OPTICS - FOPTICS):

- Раздел 2 - конце должно быть n^(c1-c0-2), а не n^(c-c0-2)

THEOREM 2.2. *For n^{c_0} (dependent) events E_i with $i \in [0, n^{c_0} - 1]$ and constant c_0 s.t. each event E_i occurs with probability $p(E_i) \ge 1 - 1/n^{c_1}$ for $c_1 > c_0 + 2$, the probability that all events occur is at least $1 - 1/n^{c - c_0 - 2}$.*

- Раздел 4 - Корень из с1, а должен быть из с10

THEOREM 4.3. *For every point $A \in \mathbb{R}^d$ holds $\overline{D}(A, dPts$ $c_{10}) < D_{avg}(A) < 2\sqrt{c_1} \cdot \overline{D}(A, dPts)$ for a constant c_{10} wh*

- Раздел 4 - красным: Merging distance: m() – минимум из средних => не превосходит каждое среднее, а в подписи к изображению 2 – наоборот.
- Раздел 4 - синим: написано: … если дистанция между А и В меньше, чем merging distance… . В формулах и коде – наоборот

4. DENSITY MEASURE AND NEIGHBORHOOD SAMPLE

Using the previous data partitioning we compute for each point a probabilistic neighborhood and an estimate of density.

Sampled Neighbors: For each point A we compute a sample of close neighbors using a set of sequences of points $\mathfrak{S}$ for a parameter $dPts$. A sequence is an ordering of points projected onto a random line (see Figure 1). For each sequence $S \in \mathfrak{S}$ and every point $A \in S$ we choose randomly a point B being at most $dPts$ points after point A in the sequence S. All the selected points around A form the sampled neighbors $\mathcal{N}(A)$. See Algorithm 2 and for an example consider Figure 2.

Algorithm 2 SampledNeighbors(set of sequences of points $\mathfrak{S}$, distance in points $dPts$, return for each point A neighbor set $\mathcal{N}(A)$)

```
1: for all P ∈ S ∈ 𝔖 do 𝒩(P) := {} end
2: for all S ∈ 𝔖 with |S| ≥ 2 · dPts do
3:    Sort S according to values of projected points
4:    for i = 1 to |S| − dPts do
5:       j := Random integer in [1, dPts]
6:       𝒩(S(i)) := 𝒩(S(i)) ∪ S(i + j)
7:    end for
8: end for
```

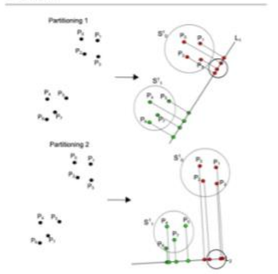


Figure 2: Two partitionings using random projections. For $dPts = 1$ a density estimate for point P_3 is obtained as follows: For the first partitioning either P_2 or P_0 is added to the sampled neighbors $\mathcal{N}(P_3)$. For the second, P_1 or P_2. Say P_0 and P_2 are chosen, i.e. $\mathcal{N}(P_3) = \{P_0, P_2\}$. Then, for $f = 1$ the average distance $Davg(P_3)$ becomes $(D(P_3, P_0) + D(P_3, P_2))/2$. The candidate(s) for P_3 can only be P_2, since $D(P_3, P_0) > m(P_3, P_0) \ge Davg(P_3)$.

Density Estimate: The density of a point A is the average

points from $N_f(A) \subset \mathcal{N}(A)$ that are among the fraction of closest points in $\mathcal{N}(A)$ for some constant f. Mathematically speaking, let C be the $|\mathcal{N}(A)| \cdot f$-th closest point in $\mathcal{N}(A)$ to A then $N_f(A) := \{B \in \mathcal{N}(A) | D(A, B) \le D(A, C)\}$ and $D_{avg}(A) := \sum_{B \in N_f(A)} D(A, B)/|N_f(A)|$. Thus, for $f = 1$ $D_{avg}(A)$ is just the average of all sampled neighbors. Note the smaller the average distance, the larger the density.

Assume that there are just $dPts+1$ points. Then $D_{avg}(A)$ is an approximation of the average distance to the $f \cdot dPts$-th (closest) neighbors. If we add a point to the dataset that is closer than the $f \cdot dPts$-th nearest neighbor N' then the average distance will decrease (in expectation). If we add a point that is farther away than the $f \cdot dPts$-th nearest neighbor B then the average distance increases. So, if we add many points that are somewhat further away than B then the added points may cause the average distance to increase significantly beyond the average distance of the $f \cdot dPts$-th closest neighbors B. However, as we shall see due to our partition process (Algorithm MultiPartition) points distant from A only appear in a set $S \in \mathfrak{S}$ with low probability. However, to ensure that the average is not significantly distorted with high probability, we do not compute only the average of all sampled neighbors but restrict ourselves to a subset dependent on the parameter f.

Candidate Mergers: Two points A, B are candidates to be merged if their distance is within merging distance. The merging distance $m(A, B)$ is just the minimum of the average distances of the two points, i.e. $m(A, B) := min(D_{avg}(A), D_{avg}(B))$. A point A may merge with any sampled neighbor $B \in \mathcal{N}(A)$, if and only if their distance is less than the merging distance $m(A, B) < D(A, B)$. Thus we restrict the pairs of points that can be merged by any clustering algorithm according to some criterion to the pairs of points A, B with $m(A, B) < D(A, B)$ with $B \in \mathcal{N}(A)$ (or $A \in \mathcal{N}(A)$), i.e. we define the candidates for point A as $\mathcal{N}_C(A) := \{B \in \mathcal{N}(A) | m(A, B) < D(A, B)\}$. This process is captured in Algorithm 3. For an example, the reader is directed to Figure 2.

Algorithm 3 CandidateMergers(points $\mathcal{P}$, distance in points $dPts$, return for each point A candidates $\mathcal{N}_C(A)$ for potential mergers)

```
1: 𝔖 := MultiPartition(𝒫, 2 · dPts)
2: 𝔖' := {S|S ∈ 𝔖, |S| < 6 · dPts}
3: Compute SampledNeighbors(𝔖', dPts)
4: for all P ∈ S ∈ 𝔖 do 𝒩_C(P) := {} end
5: for all A ∈ 𝒫 do
6:    for all B ∈ 𝒩(A) do
7:       m(A, B) := min(D_avg(A), D_avg(B))
8:       if m(A, B) < D(A, B) then 𝒩_C(A) := 𝒩_C(A) ∪ B
         end if
9:    end for
10: end for
```

Assume that to estimate the density of a point A we measure a volume $V(A)$ containing $dPts$ points. If the volumes $V(A)$ and $V(B)$ of two points intersect significantly then the density at a point contained in the intersection is likely to be of similar density of either A or B (or both). However, in case the two volumes do not intersect this does not hold. For density-based clustering we want to form clusters of points of similar density - at least all nearby points must have sim

Список источников

[1] - Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, J&g Sander OPTICS: Ordering Points To Identify the Clustering Structure // International conference on Management of data. - Munich, Germany: ACM SIGMOD, 1999. - С. 49-60.

[2] - Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander OPTICS-OF: Identifying Local Outliers // Principles of Data Mining and Knowledge Discovery. - Prague, Czech Republic: Third European Conference, PKDD'99, 1999. - С. 262-270.

[3] - Johannes Schneider, Michail Vlachos Fast Parameterless Density-Based Clustering via Random Projections: дис. IBM Research Clustering наук: F.2.0, I.5.3. - Zurich, 2013. - 6 с.

[4] - Сравниваем популярные алгоритмы кластеризации DBSCAN и OPTICS // Habr URL: https://habr.com/ru/articles/818889/ (дата обращения: 05.06.2024).

[5] - 2.3. Clustering // scikit-learn URL: https://scikit-learn.org/stable/modules/clustering.html#optics (дата обращения: 05.06.2024).

[6] - ML | OPTICS Clustering Explanation // GeeksforGeeks URL: https://www.geeksforgeeks.org/ml-optics-clustering-explanation/ (дата обращения: 05.06.2024).