q i ∈ 1 … n − 1 : p i + 1 q , p n = p

DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)

версия 2

Появление метода

Ester, Kriegel, Sander, Xu
“A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”
Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, pp. 226-231. 1996

О популярности метода

В 2010 DBSCAN стоял на 24 позиции в Microsoft Academic Search (Наиболее цитируемые статьи по интеллектуальному анализу данных согласно Microsoft academic search)

В 2014 алгоритм получил премию «проверено временем» SIGKDD Test of Time Award. Премия даётся алгоритмам, которые получили существенное внимание в теории и практике на ведущей конференции по интеллектуальному анализу данных KDD

IMHO метод опирается на теорию графов, поэтому о нем относительно легко писать статьи.

С какого текста начать изучение

Лучшее объяснение DBSCAN

https://www.quora.com/What-is-an-intuitive-explanation-of-DBSCAN

Есть перевод
https://habr.com/ru/post/322034/

Напоминание.

Синонимы — наблюдение == объект == строка == точка.

Если точки расположены близко, считаем объекты похожими.

Если в строке содержится нечисловая информация, перекодируем, чтобы все элементы строки стали числами

Идея метода

Ищем области
- с высокой плотностью точек
- связные

Как измерять плотность точек

ε - окрестность точки A — шар радиуса **ε** с центром в точке A

Плотность в окрестности точки A равна числу точек в окрестности, деленному на объем окрестности.

Объем окрестности всегда один и тот же
Чтобы сравнивать плотности в разных точках, не обязательно делить на объем шара, радиус шара каждый раз один и тот же.

Считаем, что в шаре высокая плотность точек, если в него попало не менее **m** точек

Основные параметры процедуры DBSCAN

Надо настраивать 2 параметра и еще один

ε Если расстояние между точками меньше **ε** , точки считаются близкими
m Если в окрестности точки **m** или больше точек, то эта точка находится в области, где точки расположены плотно, а плотно упакованная точками область это ядро кластера.
В scikit-learn саму точку включают в подсчет.

Кроме этих двух параметров надо определить **способ подсчета расстояний между точками**, способ подсчета расстояний формализует, что значит «похожи»,

Если эти три параметра заданы, то число кластеров определяется однозначно.
Но состав кластеров может несущественно меняться.

Вместо определения числа кластеров надо определять **ε** и **m**

Повторим то же самое более строго

Определения и обозначения.

точки ***x***, ***y*** и ***z*** – наблюдения

dist(x, y) – расстояние между точками ***x*** и ***y***

Константы **ε** и **m** нужны, чтобы определить соседей, близкие объекты.

ε окрестность объекта **x** это шар радиуса **ε** $E(x)=\{y: dist(x,y)\leqslant\varepsilon\}$

Точки из **ε**- окрестности объекта **x** будем называть соседями точки x

Будем считать их близкими, похожими на **x** точками

Другими словами,

если $z\in E(x)$, то точки **x** и **z** соседи,

если $x\in E(z)$, то точки **x** и **z** соседи

Точки бывают трех классов: **корневые, граничные и выбросы.**

Класс точки зависит от значений трех параметров.

Определение

Корневым объектом или ядерным объектом степени **m** называется объект **x**, **ε-**окрестность которого содержит не менее **m** объектов: $|E(x)|\geq m$.

Комментарий

Вокруг корневого объекта **x** большая плотность точек. Корневые точки **x** в ядре кластера

Алгоритм кластеризации

Составим список объектов из первого кластера

Для этого выберем какой-нибудь корневой объект **p** из набора данных,

Начнем с него список объектов кластера.

Добавим в список всех его соседей.

Продолжим пополнять список объектов кластера.

Для этого начнем перебирать объекты из списка.

Если в ходе перебора встречаем корневую точку, добавляем всех её соседей в список обхода.

Каждая точка входит в список один раз, повторений не допускаем.

Когда перебрали все объекты из списка, пополнение невозможно, все точки из списка составляют кластер.

Вопрос

Верно ли, что кластер состоит только из корневых объектов?

Если точка не является корневой, но среди ее соседей есть корневая точка, то точка называется **граничной**

Если точка не является корневой, и среди ее соседей нет корневых точек, то точка называется **выбросом**

Если в список попали не все точки из набора данных, можно начать составлять новый список, начиная с какого-нибудь другого корневого объекта.

Объекты из нового списка составят следующий кластер.

Новый кластер не будет пересекаться с предыдущим.

Вопрос

Что делать, если точка не попала ни в один список, ни в один кластер.

Кодом **-1** обозначаются наблюдения вне кластеров — шум, выбросы

Снова вспомним три класса точек.

Ядро кластера (корневые точки)

Граница кластера (граничные точки)

Выбросы

Вопрос

Кластеризация зависит от порядка перебора точек?

Ответ

Результат однозначен для корневых точек и для выбросов.

Граничные точки могут оказаться в разных кластерах.

Расстояния между объектами. Какие варианты уже есть в Python.

Список уже реализованных расстояний можно посмотреть в
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise_distances.html#sklearn.metrics.pairwise_distances

- scikit-learn:

  [‘cityblock’, ‘cosine’, ‘euclidean’, ‘l1’, ‘l2’, ‘manhattan’].

- scipy.spatial.distance:

  [‘braycurtis’, ‘canberra’, ‘chebyshev’, ‘correlation’, ‘dice’, ‘hamming’, ‘jaccard’, ‘kulsinski’, ‘mahalanobis’, ‘minkowski’, ‘rogerstanimoto’, ‘russellrao’, ‘seuclidean’, ‘sokalmichener’, ‘sokalsneath’, ‘sqeuclidean’, ‘yule’]

Матрица попарных расстояний может быть сосчитана заранее по любой формуле, тогда испрользуем *metric=’precomputed’*

Комментарии

DBSCAN очень похож на обобщение иерархического кластерного анализа, когда расстояние между кластерами вычисляется методом ближайшего соседа. Вместо ближайшего соседа используется m–й ближайший сосед.

DBSCAN может выявлять ленточные кластеры.

DBSCAN плохо работает, когда шаровые скопления соединены перемычками.

DBSCAN плохо кластеризует наборы данных с большой разницей в плотности, так как у всех кластеров одна и та же пара параметров **m** и **ε**

DBSCAN выигрывает, когда у нас в данных присутствуют кластеры на фоне равномерно распределенного набора точек

Дискуссия с другими текстами

http://zabaykin.ru/?p=604

https://www.dummies.com/programming/big-data/data-science/how-to-create-an-unsupervised-learning-model-with-dbscan/

"Несмотря на простоту и скорость k-means, у него есть одна очень большая проблема — это заранее заданное количество кластеров. В реальном же мире чаще всего нам не известно на сколько групп следует разбить данные"

Ответ. «В реальном же мире чаще всего нам не известны значения **m** и **ε** "

Проблема с k-means решается ценой увеличения объема вычислений.

"часть данных могут быть вредными выбросами "

k-means тоже находит выбросы, хотя возможно и хуже, чем DBSCAN

"DBSCAN отлично разбивает множество на оптимальное количество кластеров и не учитывает выбросы, "

Абсолютно рекламное заявление, совершенно неизвестно, как искать оптимум

Кластеры, найденные DBSCAN могут иметь любую форму, в отличии от алгоритма k-means (которые предполагают только выпуклую форму)

Про k-means верно, но иерархический кластерный анализ умеет искать ленточные

Отсутствие инструментов подбора параметров алгоритма DBSCAN — основной недостаток метода

Подбор параметров (Из Википедии)

Вариант 1

Перебираем параметры на решетке. Сравниваем значения «силуэта».

Недостаток метода. «силуэт» неприменим для измерения качества ленточных кластеров.

Вариант 2

m:

Если m = 2 получаем иерархический кластерный анализ примененный с методом ближайшего соседа.

Несколько эмпирических правил

m >= 3.

m >= D+1.

m >= 2*D

m = ln(n)

ε :

Если **ε** выбрана слишком малыми, большая часть данных будет отнесена к выбросам, а для слишком больших значений **ε** кластеры будут сливаться и большинство объектов окажутся в одном кластере.

Значение **ε** может быть выбрано с помощью графика, похожего на каменистую осыпь. Упорядочиваем расстояния до m-го ближайшего соседа в возрастающем порядке. Искомое значение ε соответствует «излому» графика.

Обычно малые значения **ε** предпочтительнее.

Расстояние:

Расстояние должно отражать наше представление о схожести объектов.

Дальнейшее развитие метода DBSCAN

OPTICS

OPTICS можно рассматривать как обобщение DBSCAN, в котором параметр ε заменяется максимальным значением, наиболее воздействущим на эффективность. MinPts тогда становится минимальным размером кластера. Хотя алгоритм существенно проще в области выбора параметров, чем DBSCAN, его результаты труднее использовать, так как он обычно даёт иерархическую кластеризацию вместо простого разделения, которое даёт DBSCAN.