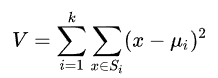
Алгоритмы кластеризации K-Means

Метод k-средних (англ. k-means) [[1]](#footnote-1) – стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров:





По аналогии с методом главных компонент центры кластеров называются также главными точками, а сам метод называется методом главных точек и включается в общую теорию главных объектов, обеспечивающих наилучшую аппроксимацию данных.

Алгоритм представляет собой версию EM-алгоритма, применяемого также для разделения смеси гауссиан. Он разбивает множество элементов векторного пространства на заранее известное число кластеров k.

Основная идея заключается в том, что на каждой итерации перевычисляется центр масс для каждого кластера, полученного на предыдущем шаге, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике.

Алгоритм (рис. 1) завершается, когда на какой-то итерации не происходит изменения внутрикластерного расстояния. Это происходит за конечное число итераций, так как количество возможных разбиений конечного множества конечно, а на каждом шаге суммарное квадратичное отклонение V уменьшается, поэтому зацикливание невозможно.

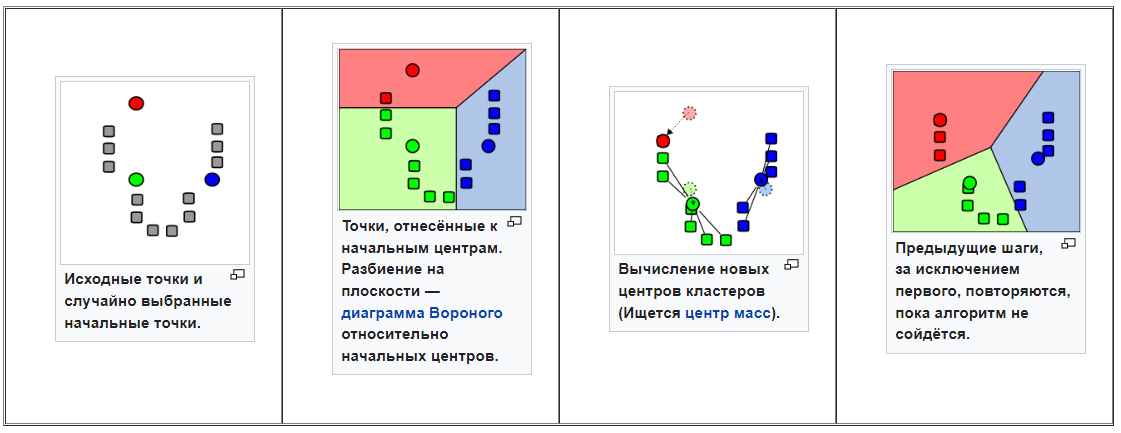


Рис. 1. Демонстрация алгоритма

Проблемные моменты алгоритма:

* Не гарантируется достижение глобального минимума суммарного квадратичного отклонения V, а только одного из локальных минимумов.
* Результат зависит от выбора исходных центров кластеров, их оптимальный выбор неизвестен.
* Число кластеров надо знать заранее.

Алгоритм DBSCAN

Density-based spatial clustering of applications with noise, DBSCAN [[2]](#footnote-2)– алгоритм кластеризации, основанной на плотности — если дан набор точек в некотором пространстве, алгоритм группирует вместе точки, которые тесно расположены, помечая как выбросы точки, которые находятся одиноко в областях с малой плотностью.

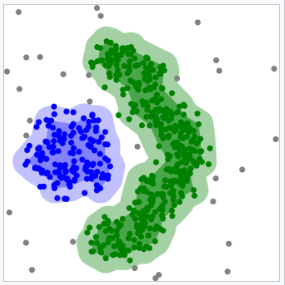
DBSCAN требует задания двух параметров: epsilon и минимального числа точек, которые должны образовывать плотную область (minPts). Алгоритм начинается с произвольной точки, которая ещё не просматривалась. Выбирается epsilon -окрестность точки и, если она содержит достаточно много точек, образуется кластер, в противном случае точка помечается как шум. Заметим, что эта точка может быть позже найдена в epsilon -окрестности другой точки и включена в какой-то кластер.

Если точка найдена как плотная точка кластера, её epsilon -окрестность также является частью этого кластера. Следовательно, все точки, найденные в epsilon -окрестности этой точки, добавляются к кластеру. Этот процесс продолжается, пока не будет найден связный по плотности кластер. Затем выбирается и обрабатывается новая не посещённая точка, что ведёт к обнаружению следующего кластера или шума.

DBSCAN может быть использован с любой функцией расстояния (а также с функцией похожести или логическим условием). Функция расстояния (dist) может поэтому рассматриваться как дополнительный параметр.

Алгоритм DBSCAN может быть разложен на следующие шаги:

1. Находим точки в epsilon окрестности каждой точки и выделяем основные точки с более чем minPts соседями.
2. Находим связные компоненты основных точек на графе соседей, игнорируя все неосновные точки.
3. Назначаем каждую неосновную ближайшему кластеру, если кластер является - соседним, в противном случае считаем точку шумом.

Наивная реализация алгоритма требует запоминания соседей на шаге 1, так что требует существенной памяти. Оригинальный алгоритм DBSCAN не требует этого за счёт того, что выполняет эти шаги для одной точки за раз.

DBSCAN (рис. 2) может найти нелинейно разделимые кластеры. Этот набор данных не может быть адекватно кластеризован методом k-средних или кластеризацией с помощью модели смеси гауссовых распределений[en] на основе максимизации математического ожидания.

Рис. 2.

Проблемные моменты алгоритма:

* DBSCAN не полностью однозначен — краевые точки, которые могут быть достигнуты из более чем одного кластера, могут принадлежать любому из этих кластеров, что зависит от порядка просмотра точек. Для большинства наборов данных эти ситуации возникают редко и имеют малое влияние на результат кластеризации — основные точки и шум DBSCAN обрабатывает однозначно. DBSCAN является вариантом, который трактует краевые точки как шум и тем самым достигается полностью однозначный результат, а также более согласованная статистическая интерпретация связных по плотности компонент.
* Качество DBSCAN зависит от измерения расстояния, используемого в функции regionQuery(P,ε). Наиболее часто используемой метрикой расстояний является евклидова метрика. Особенно для кластеризации данных высокой размерности эта метрика может оказаться почти бесполезной ввиду так называемого «проклятия размерности», что делает трудным делом нахождение подходящего значения epsilon. Этот эффект, однако, присутствует в любом другом алгоритме, основанном на евклидовом расстоянии.
* DBSCAN не может хорошо кластеризовать наборы данных с большой разницей в плотности, поскольку не удается выбрать приемлемую для всех кластеров комбинацию.
* Если данные и масштаб не вполне хорошо поняты, выбор осмысленного порога расстояния epsilon может оказаться трудным.

1. <https://ru.wikipedia.org/wiki/Метод_k-средних> [↑](#footnote-ref-1)
2. <https://ru.wikipedia.org/wiki/DBSCAN> [↑](#footnote-ref-2)