



Let us now look at a non-parametric density clustering algorithm DBSCAN. It was proposed by Martin Ester, [Hans-Peter Kriegel](#), Jörg Sander and Xiaowei Xu in 1996.^[1]

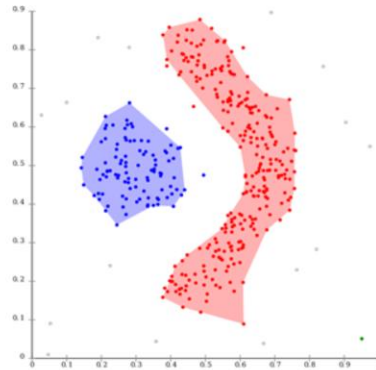
Density Based Clustering

- Density based clustering algorithms make an assumption that clusters are dense regions in space separated by regions of lower density.
- A dense cluster is a region which is “density connected”, i.e. the density of points in that region is greater than a minimum.
- Since these algorithms expand clusters based on dense connectivity, they can find clusters of arbitrary shapes.
- Density Based Spatial Clustering of Applications with Noise (DBSCAN) is an example of density based clustering algorithm.

Given a set of points in some space, the algorithm groups points that are closely packed, i.e., neighbors, together, while marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). The amount of density, and the closeness of the neighbors are values that are set by the modeler.

Density-Based Clustering II

- Clustering based on density (local cluster criterion), such as density-connected points
- Each cluster has a considerable higher density of points than outside of the cluster



This dataset cannot be adequately clustered with k-means, but DBSCAN can find non-linearly separable clusters. Notice that the density of points within clusters is much higher than those outside of the clusters.

DBSCAN

- Published by Ester et. al. in 1996
- The algorithm finds dense areas and expands these recursively to find dense arbitrarily shaped clusters.
- Two main parameters to DBSCAN are 'ε' and 'minPoints' :
 - 'ε' defines radius of the 'neighborhood region' and
 - 'minPoints' defines the minimum number of points that should be contained within that neighborhood.
- Since it has a concept of noise, it works well even with noisy datasets.

There are two main parameters that need to be defined. The first is epsilon, the parameter specifying the radius of a neighborhood with respect to some point, and minPoints, the number of points that should be contained within the neighborhood.

DBSCAN: Classification of Observations

- Given ' ϵ ' and 'minPoints', DBSCAN algorithms categorizes the objects into three exclusive groups: core points, border points and noise
- A point is a **core** point if it has more than a specified number of points (MinPts) within ϵ . These are points that are at the interior of a cluster.
- A **border** point has fewer than minPoints within ϵ , but is in the neighborhood of a core point.
- A **noise** point is any point that is not a core point nor a border point.

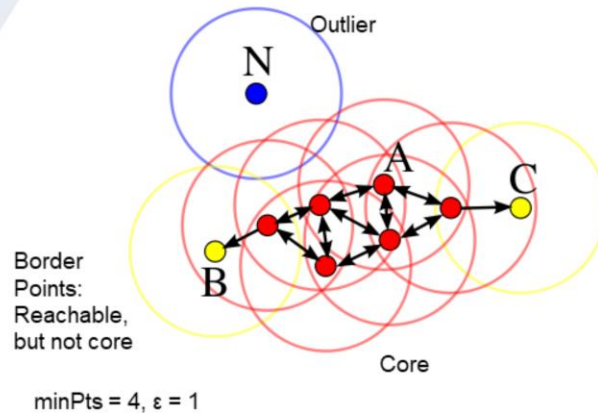
Reference: <https://en.wikipedia.org/wiki/DBSCAN>

Points are classified as core, border, or outlier as follows.

- A point p is a *core point* if at least minPts points are within distance ϵ of it (including p).
- A point q is *directly reachable* from p if point q is within distance ϵ from core point p . Points are only said to be directly reachable from core points.
- A point q is *reachable* from p if there is a path p_1, \dots, p_n with $p_1 = p$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i . Note that this implies that all points on the path must be core points, with the possible exception of q . These points we label as border points.
- All points not reachable from any other point are *outliers* or *noise points*.

Now if p is a core point, then it forms a *cluster* together with all points (core or non-core) that are reachable from it. Each cluster contains at least one core point; non-core points can be part of a cluster, but they form its "edge", since they cannot be used to reach more points.

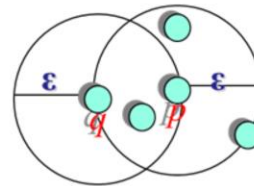
DBSCAN: Classification of Observations II

Reference: <https://en.wikipedia.org/wiki/DBSCAN>

In this diagram, $\text{minPts} = 4$. Point A and the other red points are core points, because the area surrounding these points in an ϵ radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable.

Density-Connectivity

- Directly density-reachable : An object q is directly density-reachable from object p if p is a core object and q is in p 's ϵ -neighborhood.
- q is directly density-reachable from p
- p is not directly density-reachable from q ?
- Density-reachability is asymmetric.



MinPts = 4

Reachability is not a symmetric relation since, by definition, no point may be reachable from a non-core point, regardless of distance (so a non-core point may be reachable, but nothing can be reached from it). In the diagram, p is a core point, but q is not.

DBSCAN Algorithm

```
Input: The data set D
Parameter:  $\epsilon$ , MinPts
For each object p in D
  if p is a core object and not processed then
    C = retrieve all objects density-reachable from p
    mark all objects in C as processed
    report C as a cluster
  else mark p as outlier
  end if
End For
```



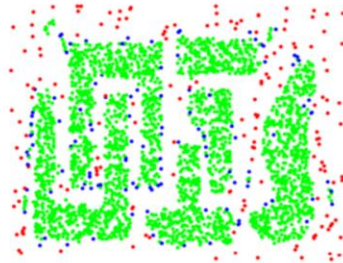
DBSCAN requires two parameters: ϵ (eps) and the minimum number of points required to form a dense region^[a] (minPts). It starts with an arbitrary starting point that has not been visited. This point's ϵ -neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized ϵ -environment of a different point and hence be made part of a cluster.

If a point is found to be a dense part of a cluster, its ϵ -neighborhood is also part of that cluster. Hence, all points that are found within the ϵ -neighborhood are added, as is their own ϵ -neighborhood when they are also dense. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

DBSCAN Example I



Original Points



Point types: core, border
and noise

Eps = 10, MinPts = 4



Here is an example of DBSCAN in action.

DBSCAN Example II



Original Points



Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes