DBSCAN

- DBSCAN refers to Density-based spatial clustering of applications with noise
- DBSCAN works fairly well with large data and can handle noise and outliers very efficiently.

Density and Dense Region

- At a certain point P, density at point P is the number of points within a hypersphere centered at P with a radius of $epsilon$
- Now, consider any region around the point P within eps radius, if there are more data points than $minpts$, we call the region a **Dense** region.
- For example, let's say we have eps=1 and $minpts$=10. Consider two points P_1 and P_2, both with a radius of eps
    - Suppose there are 20 points around point P_1, and only 6 points around point P_2, within the radius of eps, then we say the region around point P_1 is dense and the region around point P_2 as non-dense.

Min Points($minpts$) and Epsilon(eps)

- $minpts$ is the minimum number of points that we need in a hypersphere around point P with the radius of eps for considering the region as a **Dense** region.
- $minpts$ acts like a certain threshold and eps is the radius of the hypersphere

Core Point

- If a point P has points $\geq minpts$ within the radius of eps, then P is a core point.
- This also implies that point P has a dense region around it

Border Point

- A point P can be defined as a border point if:
    - P is not a core point
    - Point P lies in the **neighborhood** of point Q such that point Q is a core point

Neighborhood

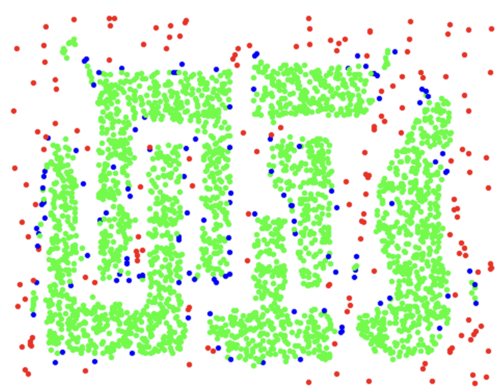
- A point P is said to be in the neighborhood of point Q if the distance between point P and Q is less than eps value; i.e. $dist(P,Q) \leq eps$

Noise Point

- It is a point that is **neither a core point nor a border point.**
- Suppose around core point P, a border point Q, and a point R which is in a non-dense region, the point R is said to be a noise point

- One thing to understand is that, when using DBSCAN, we fix two things:
  1. Min Points
  2. Epsilon.
- By fixing these hyperparameters, we get core points, border points, and noise points as well

Original Points



Point types: core, border and noise

Density Edges and Density Connected Points

- If points P and Q are two core points and the distance between point P and Q is less than or equal to eps value, then an edge between point P and Q is known as a **density edge.**

- Points P and Q can be said as density-connected points;
    - if both points are core points
    - if there exist other density edges connecting the points P and Q

- Example: Imagine we have two core points, point P, and Q, and there are other core points connecting point P with point Q; say P_1,P_2,....Pn, where the distance between each point P_1,P_2,....Pn is less than eps
    - Then point P and point Q are said to be density-connected points.

DBSCAN Algorithm

Step-1:

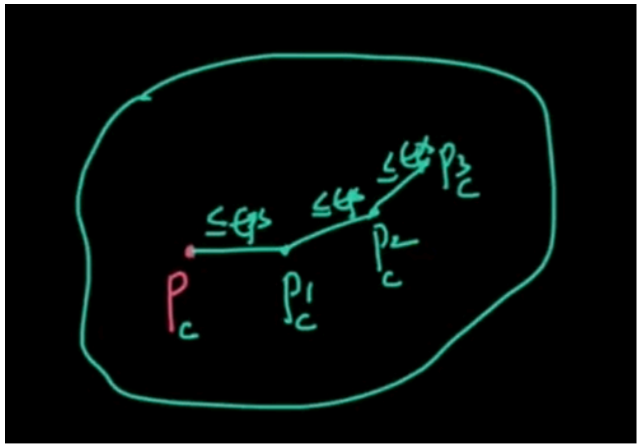
- For each point, xi that belongs to the dataset D, label it as either core point, border point, or noise point.
- The time complexity of this step would be $O(n*logN)$

Step-2:

- Remove all the noise points from the dataset
- The time complexity of this step would be $O(n)$
- This is a noise-removal step

Step-3:

- For each core point P that is not yet assigned to any clustered:
    - create a new cluster with point P
    - Add all points that are density connected to point P, to the P's cluster
- To understand this with an example, Consider a core point P and there are three core points P_1,P_2 and P_3 which are density connected.
- Then, we group all three points in the cluster of point P
- The time complexity of this step would be $O(n*logN)$



Step-4:

- For each border point, we assign it to the nearest core points' cluster.
    - For example, if we have a cluster having core points P_1,P_2....P_9, and a border point P_{10} which is near the cluster.
    - We merge border point P_{10}, into the cluster of core points P_1,P_2....P_9
- The time complexity of this step would be $O(n)*logN$

Adjusting MinPoints

- The value of $minpts$ should be greater than or equal to d+1; where d is the dimensionality of the data
    - a lot of libraries use the value of $minpts$ approximately equal to 2$*d$
- Given an epsilon value, if the dataset is noisy, we pick larger $minpts$

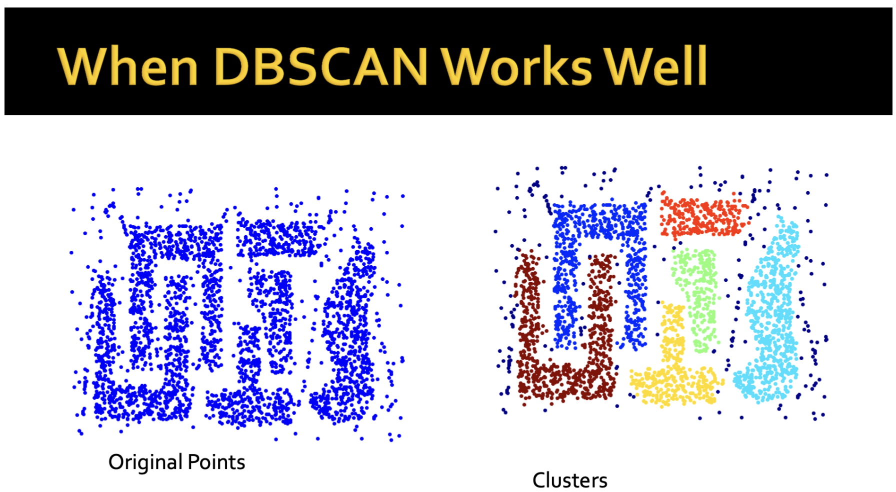
Adjusting Epsilon

- Let's assume we've fixed the value of $minpts$ = 4.
- **Step 1:**
    - for every point xi in the dataset, we compute a distance di
    - di refers to the distance from xi to xi's 4th nearest neighbor (because we've set $minpts$ = 4)
- **Step 2:**
    - Sort the values of di's and plot them. You'll notice that the distance will increase gradually and then suddenly, at a certain point, the value of distance will get boosted

   - So, the index at which the value of di distance got boosted will be used as the value of eps
   - The indices having higher values of di's will be outliers

Advantages of DBSCAN

- It's resistant to noise
- Can handle clusters of different shapes and sizes.
- It doesn't require one to specify the number of clusters a priori.
- It requires only two parameters: MinPts and Epsilon.



Limitations of DBSCAN

- Even with a small change in the hyperparameters, we can get a completely different type of cluster. So, it's quite sensitive to the choice of hyperparameters.
- Cannot handle varying densities and data with higher dimensions.