**DBSCAN (distance between nearest points)**

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a base algorithm for density-based clustering. It can discover clusters of different shapes and sizes from a large amount of data, which is containing noise and outliers.

The DBSCAN algorithm uses two parameters:

* **minPts:** The minimum number of points (a threshold) clustered together for a region to be considered dense.
* **eps (ε):** A distance measure that will be used to locate the points in the neighborhood of any point.

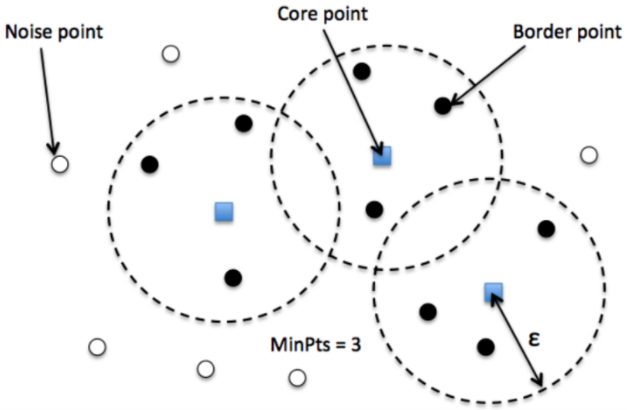
These parameters can be understood if we explore two concepts called Density Reachability and Density Connectivity.

**Reachability** in terms of density establishes a point to be reachable from another if it lies within a particular distance (eps) from it.

**Connectivity**, on the other hand, involves a transitivity based chaining-approach to determine whether points are located in a particular cluster. For example, p and q points could be connected if p->r->s->t->q, where a->b means b is in the neighborhood of a.

There are three types of points after the DBSCAN clustering is complete:

* **Core** — This is a point that has at least *m* points within distance *n* from itself.
* **Border** — This is a point that has at least one Core point at a distance *n*.
* **Noise** — This is a point that is neither a Core nor a Border. And it has less than *m* points within distance *n* from itself.



**DBSCAN algorithm can be abstracted in the following steps –**

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density connected points and assign them to the same cluster as the core point.  
   A point *a* and *b* are said to be density connected if there exist a point *c* which has a sufficient number of points in its neighbors and both the points *a* and *b* are within the *eps distance*. This is a chaining process. So, if *b* is neighbor of *c*, *c* is neighbor of *d*, *d* is neighbor of *e*, which in turn is neighbor of *a* implies that *b* is neighbor of *a*.
4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

**Parameter Estimation**

Every data mining task has the problem of parameters. Every parameter influences the algorithm in specific ways. For DBSCAN, the parameters **ε** and **minPts** are needed.

* **minPts**: As a rule of thumb, a minimum *minPts* can be derived from the number of dimensions *D* in the data set, as ***minPts* ≥ *D* + 1**. The low value ***minPts* = 1** does not make sense, as then every point on its own will already be a cluster. With ***minPts* ≤ 2**, the result will be the same as of [hierarchical clustering](https://en.wikipedia.org/wiki/Hierarchical_clustering) with the single link metric, with the dendrogram cut at height ε. Therefore, *minPts* must be chosen at least 3. However, larger values are usually better for data sets with noise and will yield more significant clusters. As a rule of thumb,***minPts* = 2·*dim*** can be used, but it may be necessary to choose larger values for very large data, for noisy data or for data that contains many duplicates.
* **ε**: The value for ε can then be chosen by using a [k-distance graph](https://en.wikipedia.org/wiki/Nearest_neighbor_graph), plotting the distance to the ***k* = *minPts*-1** nearest neighbor ordered from the largest to the smallest value. Good values of ε are where this plot shows an “elbow”: if ε is chosen much too small, a large part of the data will not be clustered; whereas for a too high value of ε, clusters will merge and the majority of objects will be in the same cluster. In general, small values of ε are preferable, and as a rule of thumb, only a small fraction of points should be within this distance of each other.
* **Distance function**: The choice of distance function is tightly linked to the choice of ε, and has a major impact on the outcomes. In general, it will be necessary to first identify a reasonable measure of similarity for the data set, before the parameter ε can be chosen. There is no estimation for this parameter, but the distance functions need to be chosen appropriately for the data set.

**Disadvantage Of K-MEANS:**

1. K-Means forms spherical clusters only. This algorithm fails when data is not spherical ( i.e. same variance in all directions).
2. K-Means algorithm is sensitive towards outlier. Outliers can skew the clusters in K-Means in very large extent.
3. K-Means algorithm requires one to specify the number of clusters a priory etc.

Advantages Of DBSCAN :

* DBSCAN algorithm overcomes all the above-mentioned drawbacks of K-Means algorithm.
* DBSCAN algorithm identifies the dense region by grouping together data points that are closed to each other based on distance measurement.
* In DBSCAN we don’t have to specify the number of clusters to use it. All you need is a function to calculate the distance between values and some guidance for what amount of distance is considered “close”.
* DBSCAN also produces more reasonable results than k-means across a variety of different distributions.