Unsupervised learning- you do not know the number of clusters or cluster membership

DBscan a density based algorithm, clusters are chosen by density

Parameters of Interest:

eps (radius) - The maximum distance between two samples for one to be considered in the neighborhood of the other. This is not a maximum bound on the distances of points within a cluster. This is the most important DBSCAN parameter to choose appropriately since it determines density (ie smaller eps means core points must have surrounding points closer in order to be a core point)

min_samples - The minimum number of samples within eps of a point for that point to be considered a core point. This includes the point itself.

Time complexity

O(nlogn) although for large datasets it approximates O(n**2), prefer DBSCAN++ for faster alternative

Problems

Vulnerable to overlapping clusters
Have to pick eps and min_samples
Is not guaranteed to produce the same result every run (it depends on which cluster the algorithm starts with)
choosing min_samples and eps sets a minimum density of points (this many points within a radius of eps). Other, less dense, clusters are missed.

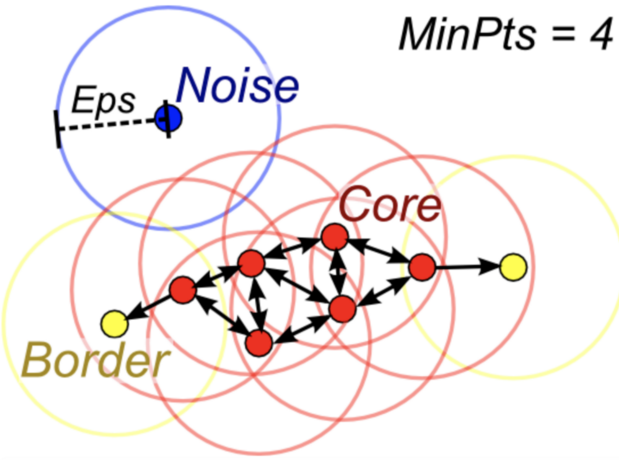
Algorithm (first choose eps, min_samples)

```
while (new points)
    pick a new point
        If there are at least min_samples within eps distance of new point, it is a core point
```

Now all core points are classified

1. Randomly pick a non clustered core point (CPn)
2. Assign to a cluster
3. for every non assigned core point within eps of CPn
4. assign this core point to that Cpn's cluster
5. Add any non-core point within eps of any core point in the cluster, to the cluster
6. If any leftover, unassigned core points, go to step 1

Any remaining points are called outliers



Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

HDBscan similar to DBscan with the addition of handling varying density clusters

Parameters of Interest:
min_cluster_size: minimum number of points needed to be considered a cluster
min_samples: same as DBscan, minimum number of neighbors to a core point. Make this high, then clusters are dense and more points in non-dense space are marked as outliers.

(note that eps is gone, which makes sense since different cluster densities will have varying eps. And if eps varies, we dont fix it as a hyperparameter)

Time complexity
O(nlogn)

Problems:
Not part of scikit-learn but performs like scikit learn estimators.
min_samples parameter is somewhat unintuitive

Algorithm
Not applicable for this course