DBSCAN

Algorithms

Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm.

Consider a set of points in some space to be clustered. For the purpose of DBSCAN clustering, the points are classified as core points, (density-)reachable points and outliers, as follows:

- A point `p` is a core point if at least `minpts` points are within distance `eps` (`eps` is the maximum radius of the neighborhood from `p`) of it. Those points are said to be directly reachable from `p`.
- A point `q` is directly reachable from `p` if point `q` is within distance `eps` from point `p` and `p` must be a core point.
- A point `q` is reachable from `p` if there is a path `p1`, ..., `pn` with `p1` = `p` and `pn` = `q`, where each `pi+1` is directly reachable from `pi` (all the points on the path must be core points, with the possible exception of `q`).
- All points not reachable from any other point are outliers.

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.

Implementations

Python is a simple language that is easy enough to understand directly. So it's not difficult to see and understand the code right away. But here are tips for Python beginners. Given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away).

`DBSCAN`

`_grow`

Grow a new cluster with label from seed point index. Search through given data to find all points that belongs to this cluster.

`_neighbor`

Find all points in given dataset with distance less than `eps`. Calculate Euclidean distance between each points in dataset and filter if less than `eps`

`_dbscan`

Generate clusters from given dataset with DBSCAN algorithm. First of all, find neighbors from `index` point. If neighbors bigger than `minpts`, It generate new cluster.

Requirements

- NumPy: is the fundamental package for scientific computing with Python.

- Pandas: is providing high-performance, easy-to-use data structures and data analysis tools for the Python.

install packages using pip

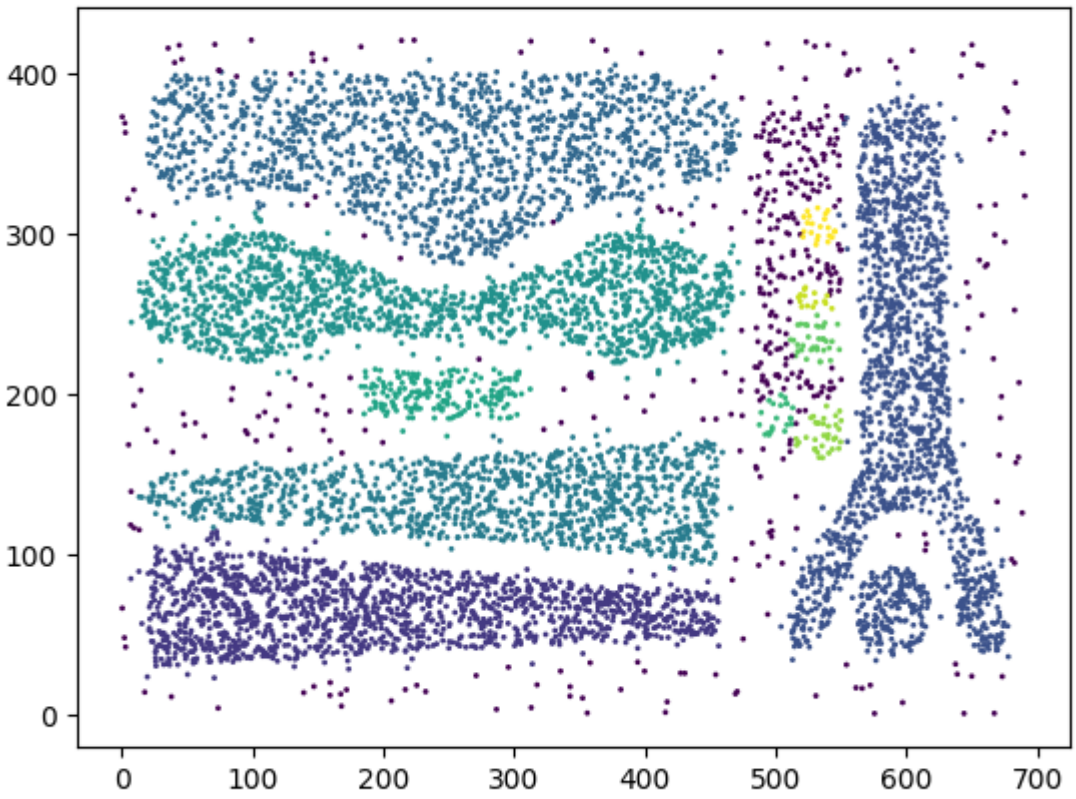
```
pip3 install -r requirements.txt
```

Tested @ python3.5 in Ubuntu 16.04 LTS, macOS High Sierra and Windows 10

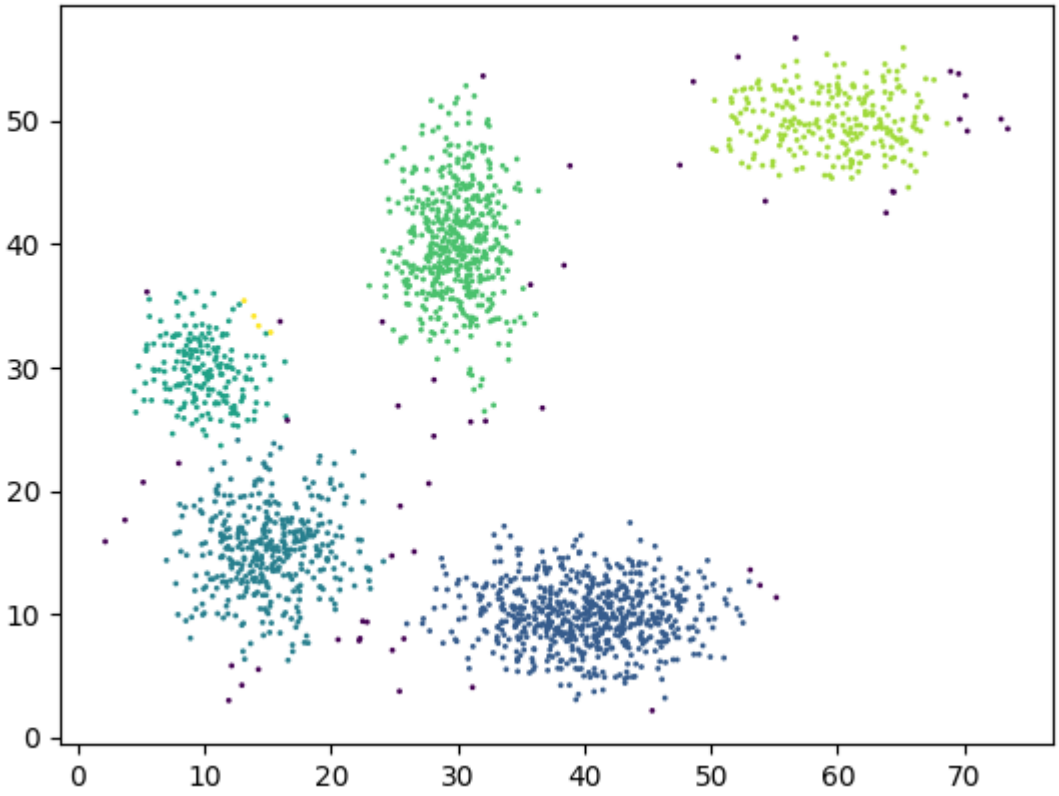
Run as below

```
python3 dt.py (input) (n) (eps) (min) [--output output_path] [--image]
```

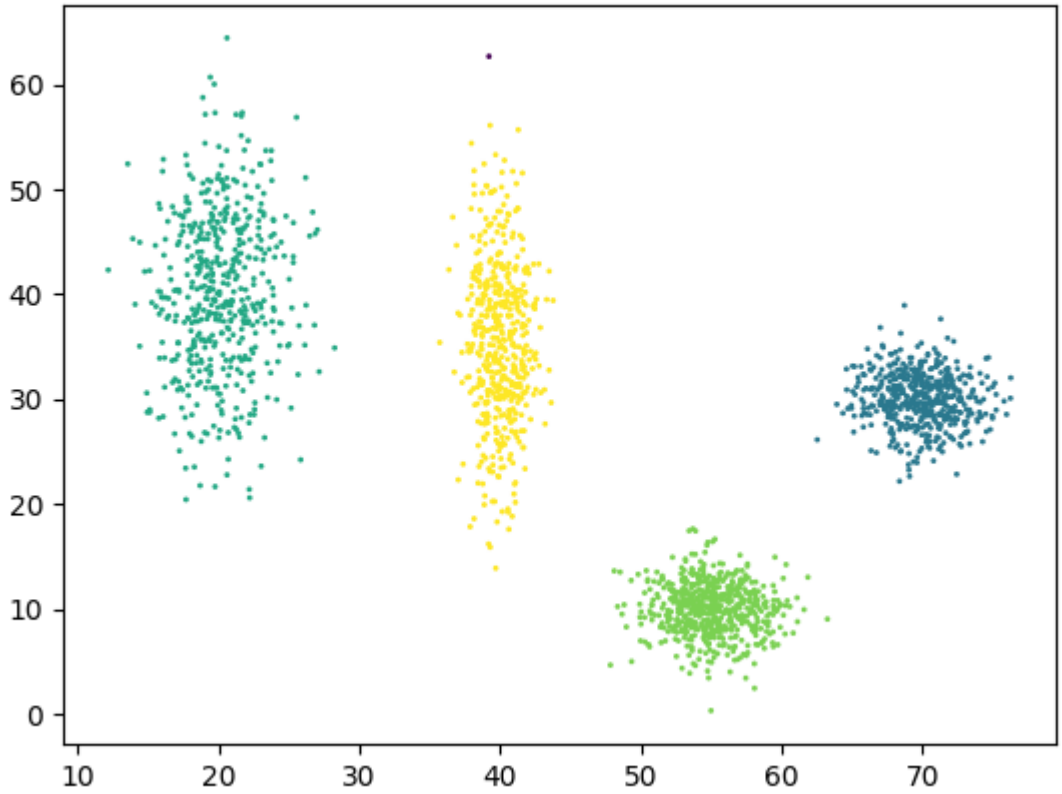
Performance



input1



input2



input3