

Markiyan Varhola

9016820171

## DBSCAN Implementation

The main goal of the program was to implement the DBSCAN algorithm and apply it to the testing data sets.

The DBSCAN algorithm requires 2 parameters - epsilon, which specifies how close points should be to each other to be considered a part of a cluster; and minPts, which specifies how many neighbors a point should have to be included into a cluster. The DBSCAN algorithm clusters points in the following manner:

1. Start with an arbitrary starting point that has not been visited.
2. Extract the neighborhood of this point using  $\epsilon$  (All points which are within the  $\epsilon$  distance are neighborhood).
3. If there are sufficient neighborhood around this point then clustering process starts and point is marked as visited else this point is labeled as noise (Later this point can become the part of the cluster).
4. If a point is found to be a part of the cluster then its  $\epsilon$  neighborhood is also the part of the cluster and the above procedure from step 2 is repeated for all  $\epsilon$  neighborhood points.  
This is repeated until all points in the cluster is determined.
5. A new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.
6. This process continues until all points are marked as visited.

### Instructions on running script:

The script was tested on Python 2.7.3, and running it on Python 3> will result in an error. In order to run the script, use the format as outlined in the original assignment document.

### To run the script, use the following format:

```
python dbscan.py {input testing file} {number of clusters} {eps} {min_points}
```

*For example:*

```
[Assignment 3] python dbscan.py data/input3.txt 4 5 5          master *  
Hold on! This might take a while...  
[ ] 0%
```

This will produce the required output files in the ./test/ directory, where then can be tested with the testing program.

### Current script output:

```
[test] wine PA3.exe input1          master *  
84.18187?%  
[test] wine PA3.exe input2          master *  
91.6039?%  
[test] wine PA3.exe input3          master *  
99.97736?%
```