

Assignment #3

2017029970 우원진

1. Algorithm

먼저 시간을 단축해야겠다는 생각에 Dictionary를 사용하기로 결정했다. 왜냐하면 list에서 원소를 제거하려면 $O(N)$ 의 시간이 걸리지만 Dictionary에서 del은 $O(1)$ 이기 때문이다.

```
while(data_dict.keys()):
    # random_idx = random.randint(0, len(data_dict)-1)
    criteria = list(data_dict.keys())[0]
    queue = deque([criteria]) # core candidate
    clusters.append([])
    while(queue):
        criteria_candidate = queue.popleft()
        if not criteria_candidate in visited_idx:
            clusters[cnt].append(criteria_candidate)
            visited_idx.append(criteria_candidate)
            x1, y1 = data_dict[criteria_candidate]
            temp_list = [] # for check min_pts
            for key in data_dict.keys():
                x2, y2 = data_dict[key]
                if is_in_distance(x1, y1, x2, y2, eps):
                    temp_list.append(key)
            if len(temp_list) >= min_pts:
                for key in temp_list:
                    queue.append(key)
        for d in clusters[cnt]:
            del data_dict[d]
        cnt += 1
```

이 부분이 main 알고리즘이다. data_dict는 {index:[x_coordinate, y_coordinate]}형태이다. 그리고 각각의 클러스터에 대해 core가 될 수도 있는 후보 index들이 queue에 담기게 된다. 그리고 visited_idx 라는 list도 있는데 이 visited_idx라는 list는 그 index가 기준이 되어 min_pts와 eps에 대해 계산 된 과거가 있는 index들이 담겨있다. 그리고 cnt는 클러스터를 구분하는 기준으로 clusters라는 list가 선언되어있어, 이 list에 index마다 해당 클러스터에 해당하는 index들이 모여 있는데 여기서 cnt는 cluster의 index를 의미한다.

그리고 이미 cluster로 구분된 애들은 다음 cluster를 찾을때 필요가 없으므로 이를 제거하는 코드도 추가했다.

2. Instruction for Compiling

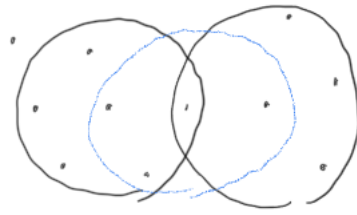
```
uwonjin-ui-MacBookPro:Assignment#3 woowonjin$ python3 clustering.py input2.txt 5 2 7
```

파이썬을 사용했기때문에 위 사진처럼 python3 clustering.py input_file n eps min_pts 이런식으로 명령어를 작성하면 실행까지 한번에 된다.

3. Others

처음 DBSCAN을 고민해봤을때는 랜덤하게 기준을 뽑는다고 해도 모든 상황에서 결과가 똑같이 나와야한다고 생각했다. 그런데 실제로 Random하게 하니 결과가 조금씩 다르게 나타났다. 그래서 이 부분에대해 조금 고민을 해봤는데 결론적으로

$$\min\{p\} = 5$$



이 상황과 같이 한 점이 양쪽 Cluster의 border가 되었을때, 랜덤하게 한다면 어떤걸 먼저 하냐에 따라 점의 Cluster도 바뀔수 있겠다고 생각했다. 하지만 결과의 일관성을 위해 Random하게 기준점을 잡지 않고, 위의 data_dict의 첫번째 키로 기준점을 잡았다.