Report

2016025196

김동규

1. 실행 환경 및 실행 방법

환경 : Windows 10, python 3.9.2, numpy1.19.5

실행 방법 : assignment3.py (input.txt)

2. 알고리즘 요약

임의의 한 점에서 시작하여 eps조건과 minpts조건을 확인한다. 이때, minpts가 넘으면 해당 점에서 클러스터 구성을 시작하며, 연쇄적으로 퍼져나가면서 클러스터를 구성한다. minpts조건이 안 될 경우 추후 다른 점에 의해 density connected가 될 때까지 outlier로 처리한다. Remain points가 남지 않을 때까지 반복한다.

3. 코드설명

Cluster\_id는 0에서 시작하여 클러스터 집합이 구성될때마다 1씩 증가한다. Remain\_DB는 원래 DB상태에서 시작하여 outlier를 포함한 클러스터에 속하지 않는 점들을 제외하며 줄어든다.

크게 2단계의 과정으로 클러스터링한다.

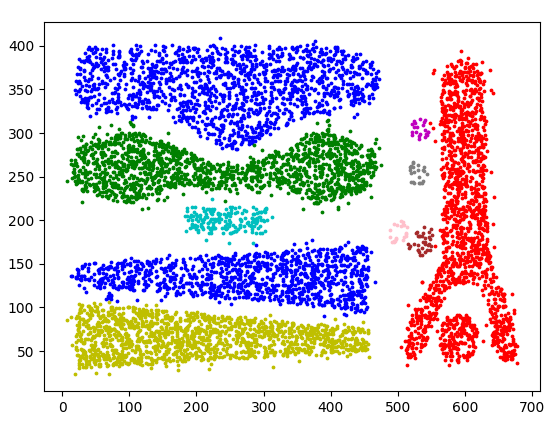
1. Remain\_DB에서 임의의 한 점을 선택한다. 해당 점에서 eps와 minpts를 만족할 경우 클러스터를 구성하고 주변 점을 큐에 넣는다.
2. 큐가 비어있을 때까지 한 점을 꺼내 클러스터 집합을 구성한다. 주변 점도 큐에 다시 넣는다.

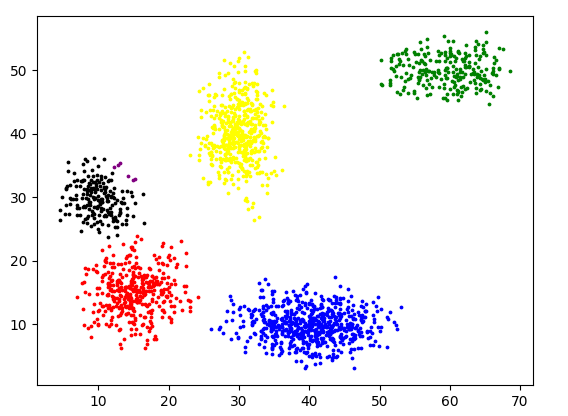
이후 출력결과를 쓸 때는 DB에 저장한 cluster id 값에 따라 별도의 파일로 저장한다.

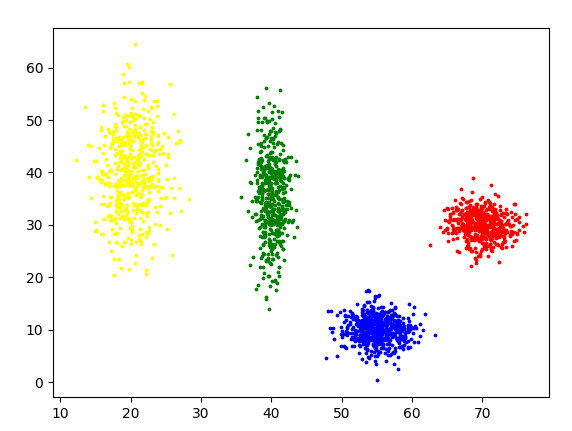
텍스트, 영수증, 스크린샷이(가) 표시된 사진

자동 생성된 설명

4. 실험 결과

input1

input2

 input3

결과값의 측정을 위해 matplot으로 클러스터 집합을 출력하는 코드를 작성했다. 텍스트이(가) 표시된 사진

자동 생성된 설명

PA3.exe의 값은 25~40점인데 반해 실제적인 클러스터링 결과는 잘 되어있음을 알 수 있었다.