TERM PROJECT

2016024766
김서현

프로젝트 설계(1/2)

- 5가지의 각기 다른 정규분포를 따르는 랜덤한 3차원 좌표들을 300개씩 생성합니다.
  - *각 cluster들은 적절하게 overlap되어야 합니다.*
- 생성된 1500개의 3차원 좌표들에 대해서 K Means clustering을 합니다.
  - *K 값은 5입니다.*
- K Means clustering을 통해 계산한 Mean vectors를 이용해서 각 cluster에 속한다고 판단하는 boundary를 정해주는 maximum distance를 설정합니다.
  - *Cluster마다 적절한 값으로 maximum distance를 설정합니다.*

프로젝트 설계(2/2)

- 기존에 생성한 좌표들과 동일한 정규분포를 갖는 랜덤한 3차원 좌표들을 100개씩 생성하고, 기존에 없던 정규분포를 갖는 랜덤한 3차원 좌표들을 100개 생성합니다.
- 생성된 600개의 좌표들을 가지고 올바른 cluster로 잘 인식이 되는지 테스트를 진행합니다.

Maximum distance 설정

- 처음에 생성한 5개의 정규분포 clusters를 이용해 maximum distance를 설정했습니다.

1. 각 cluster 내의 300개의 좌표들마다 가장 가까운 mean vector와의 거리를 구하고, 두번째로 가까운 mean vector와의 거리를 구했습니다.
2. 위에서 구한 두 거리를 300개의 좌표들에 대해 각각 평균을 구했습니다.
3. Cluster마다 가장 가까운 mean vector와의 거리 평균과 두번째로 가까운 mean vector와의 거리 평균을 구하였는데, 그 두 거리 평균의 중간값을 maximum distance로 설정했습니다.
    - ✓ Cluster마다 각각 다른 maximum distance를 갖게 됩니다.

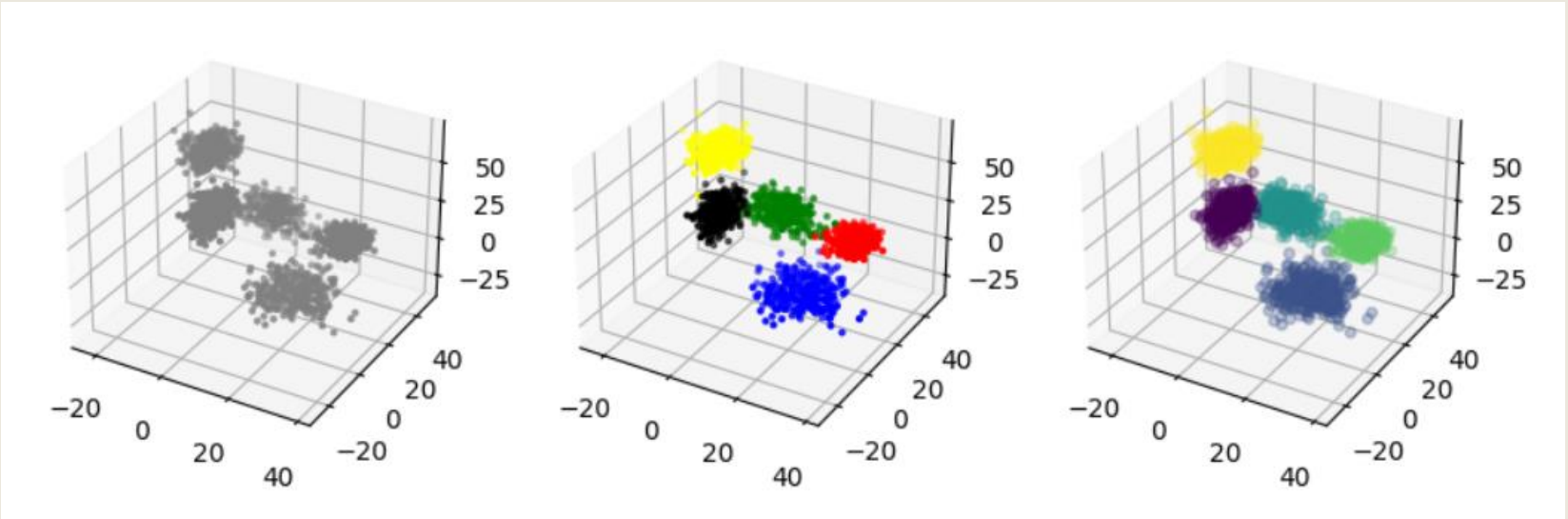
➢ **이런 방법으로 maximum distance를 설정하면 기존과 동일한 정규분포를 갖는 점들은 포함하고, 아예 다른 분포를 가진 점들은 배제할 수 있을 것이라 생각했습니다.**

프로젝트 구현

- Numpy의 random.normal()을 이용해서 랜덤한 좌표들을 뽑았습니다.
- Sklearn의 Kmeans()를 이용해서 K Means clustering을 하였습니다.
- 유클리드 거리를 계산해서 좌표들과 mean vector들과의 거리를 구하고, maximum distance도 설정하였습니다.
- 테스트를 할 때에는 각 좌표와 가장 가까운 mean vector를 구하고, 그 mean vector의 cluster의 maximum distance보다 좌표와 mean vector의 거리가 더 가까우면 그 cluster에 속한다고 판단하고, maximum distance가 더 크다면 어떤 cluster에도 속하지 않는다고 판단하도록 구현했습니다.

프로젝트 결과 #1

- 각 cluster들이 거의 overlap 되지 않는 경우



1500개 좌표 초기 clusters K Means

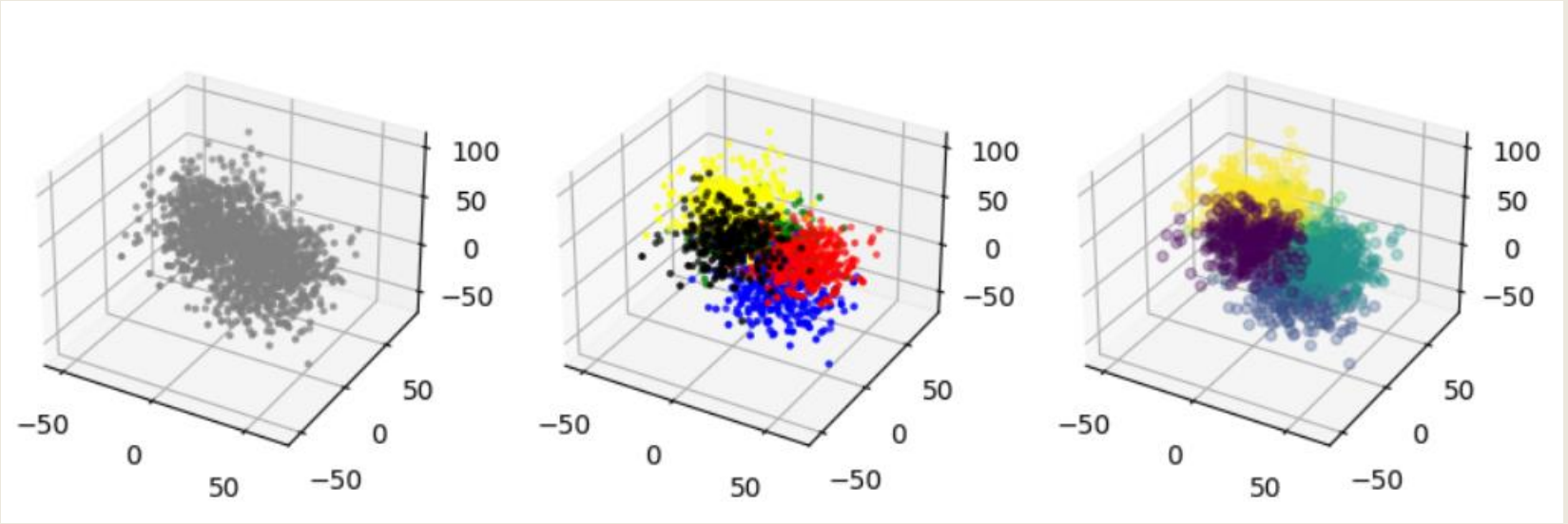
프로젝트 결과 #1

- 각 cluster들이 거의 overlap 되지 않는 경우

- cluster1 인식률: 100%
- cluster2 인식률: 100%
- cluster3 인식률: 100%
- cluster4 인식률: 100%
- cluster5 인식률: 100%
- 없는 데이터 인식률: 95%

프로젝트 결과 #2

- 각 cluster들이 많이 overlap 되는 경우 (결과#1에서 분산만 증가시킴, mean은 동일)



1500개 좌표 초기 clusters K Means

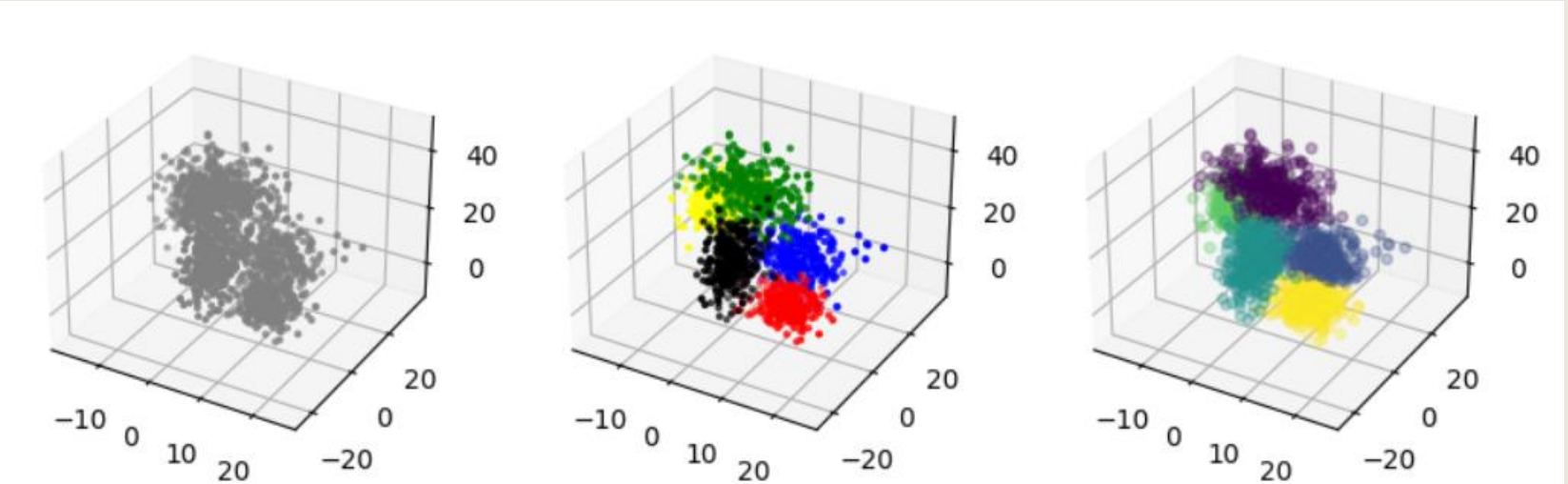
프로젝트 결과 #2

- 각 cluster들이 많이 overlap 되는 경우 (결과#1에서 분산만 증가시킴, mean은 동일)

- cluster1 인식률: 70%
- cluster2 인식률: 54%
- cluster3 인식률: 73%
- cluster4 인식률: 85%
- cluster5 인식률: 77%
- 없는 데이터 인식률: 50%

프로젝트 결과 #3

- 각 cluster들이 많이 overlap 되는 경우 (결과#1에서 mean을 서로 가깝게 설정, 분산은 동일)



1500개 좌표 초기 clusters K Means

프로젝트 결과 #3

- 각 cluster들이 많이 overlap 되는 경우 (결과#1에서 mean을 서로 가깝게, 분산은 동일)

- cluster1 인식률: 90%
- cluster2 인식률: 72%
- cluster3 인식률: 99%
- cluster4 인식률: 78%
- cluster5 인식률: 95%
- 없는 데이터 인식률: 92%

결과 분석

- 결과#1의 인식률이 매우 높은 것은 보아 각 clusters를 생성할 때 각 mean들끼리 거리가 멀고, 분산을 작게 설정하면 인식이 잘 되는 것을 알 수 있습니다.
- 결과#2는 cluster 생성 시 결과#1에서와 동일한 mean을 사용했지만, 분산을 더 크게 조정해서 clusters끼리 서로 많이 overlap되도록 설정한 결과입니다. 결과 #1에 비교해서 전체적인 인식률이 많이 떨어지는 것을 알 수 있습니다.
- 결과#3은 cluster 생성 시 결과#1에서와 동일한 분산을 사용했지만, 각 cluster의 mean들끼리 더 가깝게 조정해서 서로 많이 overlap되도록 설정한 결과입니다. 결과#1에 비하면 인식률이 떨어지긴 했지만 결과#2보다는 훨씬 높습니다.

  - **이를 통해 cluster를 잘 인식하는 데에는 각 clusters의 mean 사이의 거리보다는 각 cluster의 분산이 작은 것이 더 중요하다는 것을 알 수 있습니다.**