수치해석 #Term-Proect

2017029970
우원진

1. Training Data 설정

Mean

```
means = [(-1, -1, -1), (0, 0, 0), (1, 1, 2), (3, 4, -1), (4, 4, 4)]
```

위와 같이 데이터를 Cluster 별로 생성하기 위한 처음 Center로는 각 Cluster마다 (-1, -1, -1), (0, 0, 0), (1, 1, 2), (3, 4, -1), (4, 4, 4) 로 잡았습니다.
나중에 Labeling이나 색깔 구분을 쉽게 하기 위해 x축의 좌표가 작은 것부터 차례대로 설정했습니다.
나중에도 설명하겠지만 각각 center별로 순서대로 데이터 그래프에서 아래와 같은 색깔로 나타납니다.

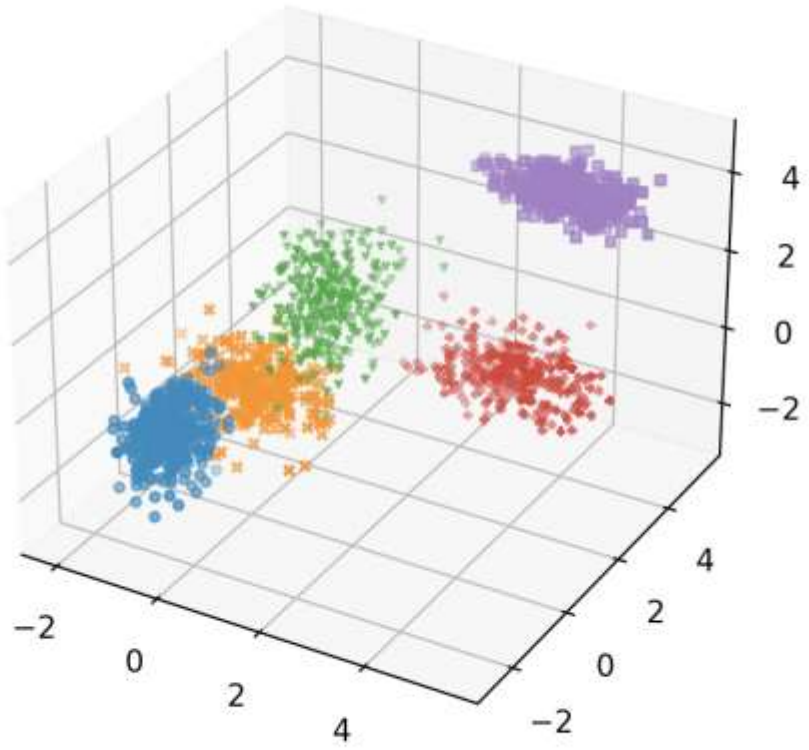
```
# blue, orange, green, red, purple
```

Covariance

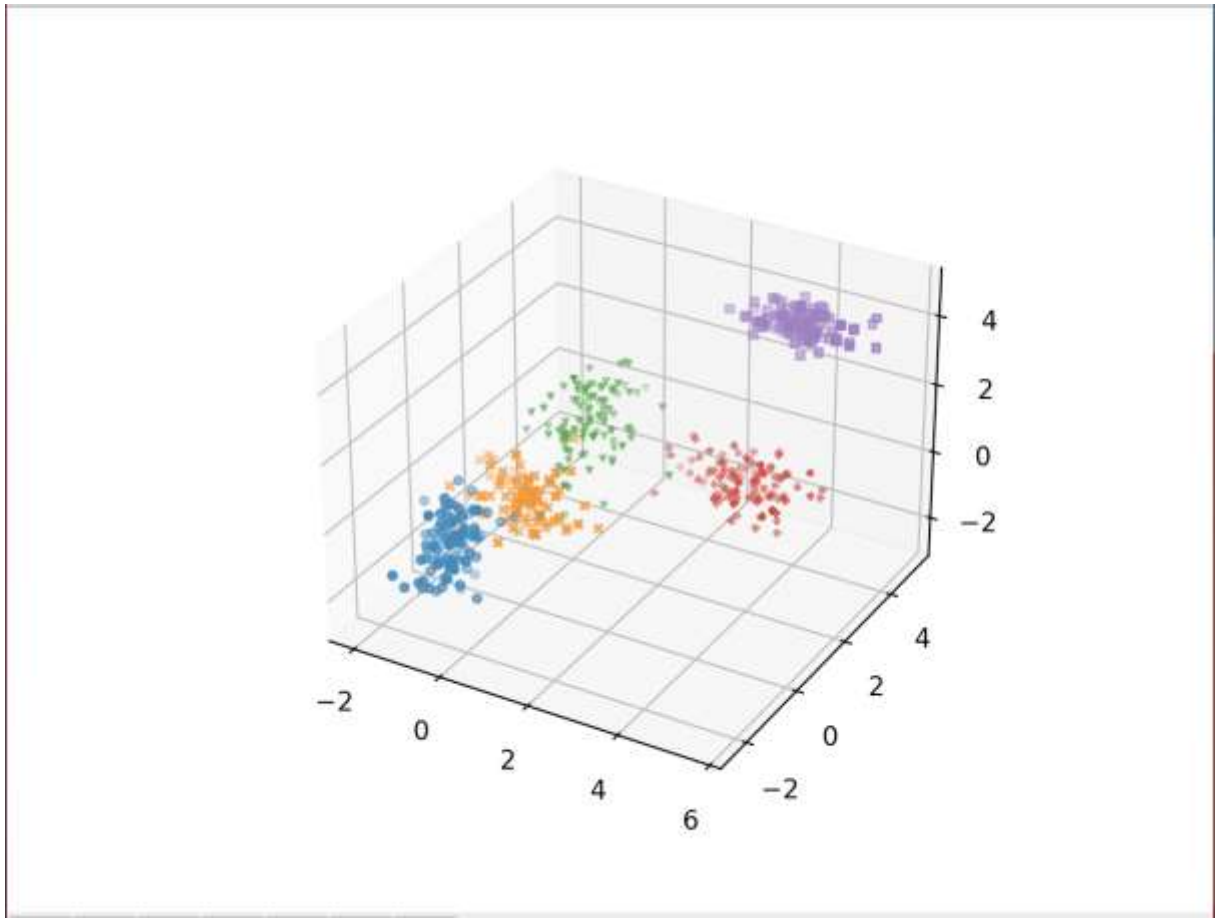
```
covs = [
    [[0.1, 0, 0], [0, 0.3, 0], [0, 0, 0.4]],
    [[0.4, 0, 0], [0, 0.2, 0], [0, 0, 0.3]],
    [[0.2, 0, 0], [0, 0.5, 0], [0, 0, 0.6]],
    [[0.6, 0, 0], [0, 0.2, 0], [0, 0, 0.2]],
    [[0.4, 0, 0], [0, 0.1, 0], [0, 0, 0.1]],
]
```

Covariance는 데이터가 3차원이기 때문에 Cluster 별로 3x3 행렬로 나타내었습니다.
그리고 데이터에서 X와 Y와 Z 축은 각각 독립이라고 생각했기때문에 서로의 연관성은 없다고 나타내기 위해 0 으로 두어 Diagonal Matrix로 나타냈습니다.

Training Cluster Data 결과



2. Training Data Set와 같은 Center, Covariance를 가지고 만든 Test Data분포



2 를 가지고 테스트 한 결과

```
[올바르게 labeling., data와의 distance가 다른 mean cluster보다 크다., 최소거리에서의 label은 맞았지만 데이터 와의 거리가 max_dist를 넘어간다.]
[[86, 4, 10], [83, 10, 7], [86, 4, 10], [86, 0, 14], [94, 0, 6]]
```

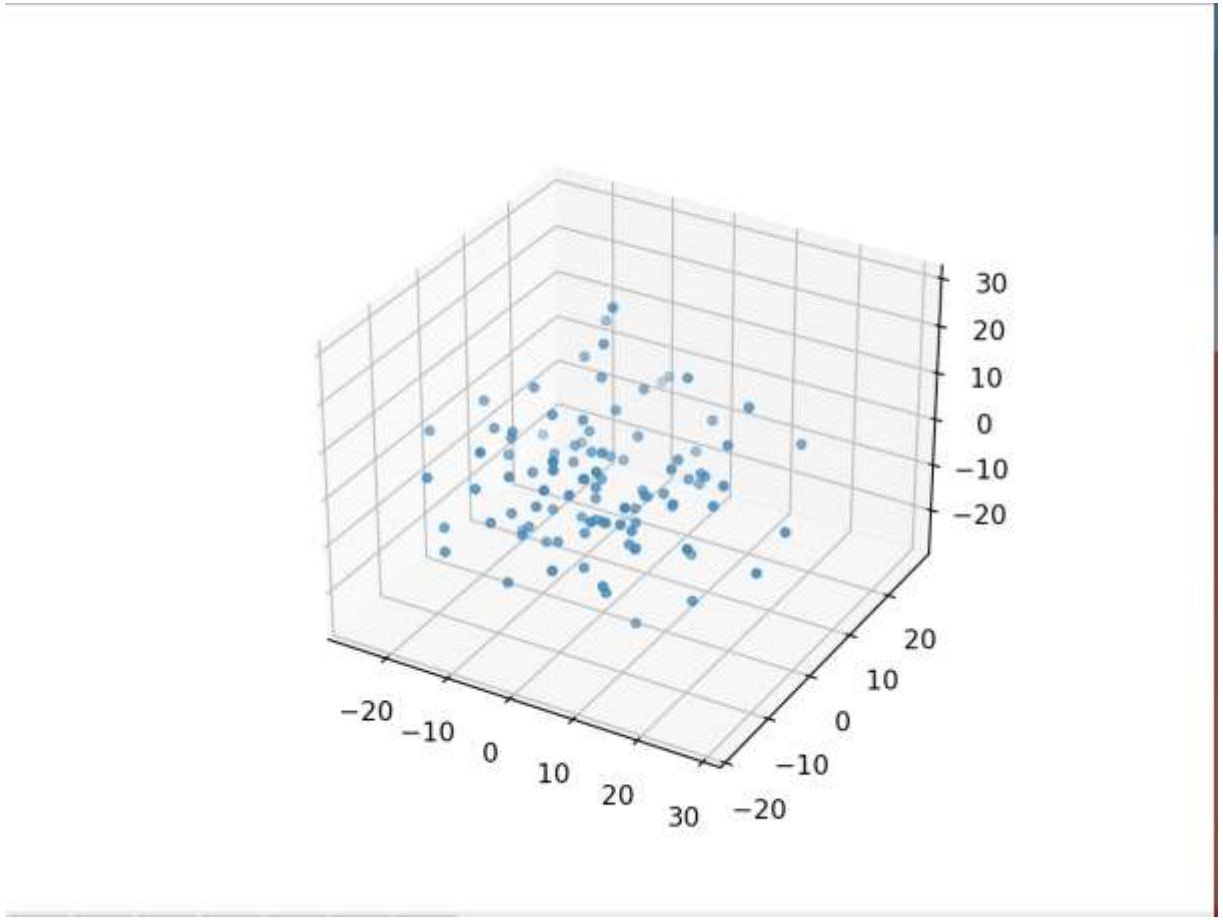
위의 결과가 의미하는 바는
[86, 4, 10]은 Cluster 0 과 같은 Center과 Covariance로 만든 테스트 데이터는 이론적으로 Cluster 0 으로 Labeling 되어야합니다. 여기서 86은 100개 중 이를 만족하여 Cluster 0으로 Labeling된 갯수 이고, 4 는 Cluster 0 의 Mean과의 거리가 다른 Cluster의 Mean과의 거리보다 크기 때문에 다른 Cluster로 Labeling 된 경우입니다. 또 10은 Cluster 0으로 Labeling은 되었지만 Cluster 0 과의 거리가 제가 설정한 max_distance를 넘어가기 때문에 noise로 보고, 에러라고 판단한 데이터입니다.(여기서 max_distance의 기준은 뒤에서 설명드리겠습니다.)
위의 [86, 4, 10], [83, 10, 7], [86, 4, 10], [86, 0, 14], [94, 0, 6]은 위의 과정을 각각 Cluster0, Cluster1, Cluster2, Cluster3, Cluster4에 대해 한 결과입니다.

2 의 결과에 대한 분석

2는 Training Cluster Data와 같은 Mean, Covariance를 가지고 했기 때문에 거의 정확도가 높아야 한다고 생각합니다. 하지만 max_dist에 대한 것은 noise 즉 에러라고 보고도 Labeling이 잘못되는 결과가 발생하는데 이는 Cluster0, Cluster1, Cluster2에서 발생합니다.

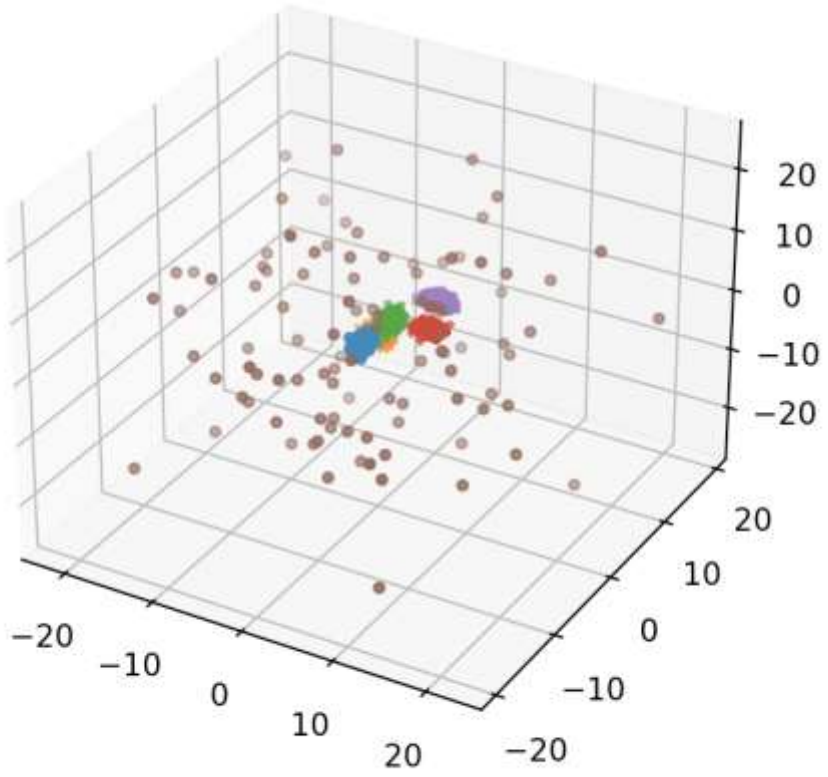
왜냐하면 Cluster3 과 Cluster4 같은 경우에는 거의 독립적으로 분포하고있지만 Cluster0, Cluster1, Cluster2 같은 경우에는 서로 데이터들이 boundary를 넘어 섞여있기 때문입니다. 이는 Mean을 조금더 떨어뜨려주거나 Covariance를 작게 해주면 해결됩니다.

3. Randomly Distributed Test Data 분포



이는 (0, 0, 0)의 Mean과
[[100, 0, 0],
[0, 100, 0],
[0, 0, 100]]
의 Covariance를 가지고 만든 Test Data 입니다.
이렇게 분산을 크게 둬서
데이터가 최대한 퍼지도록
했습니다.

3 에 대한 해석



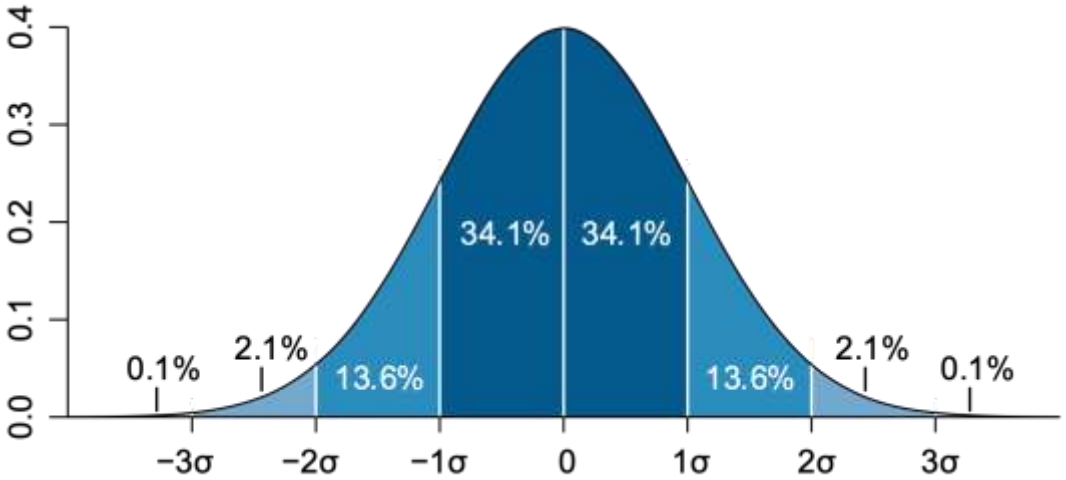
이는 Training에 의해 만들어진 Cluster과 3으로 만들어진 Data와의 관계를 보여줍니다.
이와 같이 거의 겹치는 부분이 없다고 볼 수 있습니다.

3 의 결과

```
[ 0.35043829 -0.85007696  0.12639338]  is in cluster num(1)
Random distribution N([0, 0, 0], sigma_x^2=100, sigma_y^2=100, sigma_z^2=100) Result :  [0, 1, 0, 0, 0] are in clusters[0, 1, 2, 3, 4]
```

코드를 여러번 돌려봤지만 대부분 Cluster에 들지 못했고 간혹 가다가 위의 결과와 같이 Cluster에 속하는 경우도 있었습니다.

Data 분석 방법



여기서 사용한 데이터들은 모두 Normal Distribution을 사용하고 있기 때문에 왼쪽의 그림과 같은 신뢰도를 가진다고 생각했습니다.
그래서 max_distance를 설정할때 기준을 3*sigma를 사용하면 거의 정확한 데이터를 측정할 수 있다고 생각했습니다.

```
num = 3
max_dist = [
    [
        num * math.sqrt(covs[0][0][0]),
        num * math.sqrt(covs[0][1][1]),
        num * math.sqrt(covs[0][2][2]),
    ],
    [
        num * math.sqrt(covs[1][0][0]),
        num * math.sqrt(covs[1][1][1]),
        num * math.sqrt(covs[1][2][2]),
    ],
    [
        num * math.sqrt(covs[2][0][0]),
        num * math.sqrt(covs[2][1][1]),
        num * math.sqrt(covs[2][2][2]),
    ],
    [
        num * math.sqrt(covs[3][0][0]),
        num * math.sqrt(covs[3][1][1]),
        num * math.sqrt(covs[3][2][2]),
    ],
    [
        num * math.sqrt(covs[4][0][0]),
        num * math.sqrt(covs[4][1][1]),
        num * math.sqrt(covs[4][2][2]),
    ],
]
```

그래서 왼쪽의 코드와 같이 3*sigma를 max Distance를 설정하는데 사용했습니다. 이는 타원체의 식 x^2/a^2 + y^2/b^2 + z^2/c^2 = 1 에서 a, b, c를 각 Cluster의 Mean에서 X, Y, Z축에 대한 3*sigma 라고 생각했습니다. 이를 이용해 아래의 코드와 같이 데이터가 타원체 내부에 존재해야 max_distance를 만족한다고 생각하여 x^2/a^2 + y^2/b^2 + z^2/c^2 <= 1 를 기준으로 테스트를 했습니다.

```
if (
    pow(min_center[0] - x, 2) / max_dist[min_idx][0]
    + pow(min_center[1] - y, 2) / max_dist[min_idx][1]
    + pow(min_center[2] - z, 2) / max_dist[min_idx][2]
    <= 1
):
```