

## ex3

January 10, 2023

```
[ ]: # Create a Toy dataset
```

```
[ ]: !pip install scikit-learn
```

Collecting scikit-learn

Downloading

scikit\_learn-1.2.0-cp39-cp39-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (9.5 MB)

9.5/9.5 MB

1.4 MB/s eta 0:00:0000:0100:010m

Collecting joblib>=1.1.1

Using cached joblib-1.2.0-py3-none-any.whl (297 kB)

Requirement already satisfied: scipy>=1.3.2 in

/home/pranav/miniconda3/envs/gal/lib/python3.9/site-packages (from scikit-learn) (1.9.3)

Requirement already satisfied: numpy>=1.17.3 in

/home/pranav/miniconda3/envs/gal/lib/python3.9/site-packages (from scikit-learn) (1.24.1)

Collecting threadpoolctl>=2.0.0

Using cached threadpoolctl-3.1.0-py3-none-any.whl (14 kB)

Installing collected packages: threadpoolctl, joblib, scikit-learn

Successfully installed joblib-1.2.0 scikit-learn-1.2.0 threadpoolctl-3.1.0

```
[ ]: import numpy as np
from scipy import sparse
from sklearn.neighbors import kneighbors_graph
from sklearn.cluster import KMeans
from sklearn.metrics import adjusted_rand_score, homogeneity_score, \
    completeness_score, v_measure_score
import networkx as nx
# import make moons
from sklearn.datasets import make_moons
```

```
[ ]: import matplotlib.pyplot as plt
random_state = 213
np.random.seed(random_state)
data_size = 150
```

```
features,y = make_moons(n_samples=data_size, noise=0.07, random_state=213)
```

```
[ ]: n_neighbors = 10
      knn_dist_graph = kneighbors_graph(X=features,
      n_neighbors=n_neighbors,
      mode='distance',
      metric='euclidean',
      n_jobs=6
      )
```

```
[ ]: sigma = 1
```

```
[ ]: similarity_graph = sparse.csr_matrix(knn_dist_graph.shape)
nonzeroindices = knn_dist_graph.nonzero()
similarity_graph[nonzeroindices] = np.exp( -np.
    ↳ asarray(knn_dist_graph[nonzeroindices])**2 / 2.0 * sigma**2)
similarity_graph = 0.5 * (similarity_graph + similarity_graph.T)
similarity_graph.todense()[0]
```

```

/home/pranav/miniconda3/envs/gal/lib/python3.9/site-
packages/scipy/sparse/_index.py:146: SparseEfficiencyWarning: Changing the
sparsity structure of a csr_matrix is expensive. lil_matrix is more efficient.
    self._set_arrayXarray(i, j, x)

```

[illegible]

```

0.          , 0.          , 0.99597112, 0.          , 0.          ,
0.          , 0.97162455, 0.          , 0.          , 0.          ,
0.          , 0.          , 0.          , 0.          , 0.98839004,
0.          , 0.          , 0.          , 0.          , 0.          ,
0.          , 0.          , 0.          , 0.          , 0.          ,
0.          , 0.          , 0.          , 0.          , 0.98710353,
0.          , 0.          , 0.          , 0.          , 0.          ,
0.          , 0.          , 0.          , 0.          , 0.          ]]

```

```

[ ]: degree_matrix = similarity_graph.sum(axis=1)
     diagonal_matrix = np.diag(np.asarray(degree_matrix).reshape(data_size,))
     L = diagonal_matrix - similarity_graph

```

```

[ ]: L

```

```

[ ]: matrix([[ 9.88985997,  0.          , -0.99736371, ...,  0.          ,
               0.          ,  0.          ],
 [ 0.          ,  9.84957185,  0.          , ...,  0.          ,
               0.          ,  0.          ],
 [-0.99736371,  0.          ,  9.40776641, ...,  0.          ,
               0.          ,  0.          ],
 ...,
 [ 0.          ,  0.          ,  0.          , ...,  9.88244354,
               0.          ,  0.          ],
 [ 0.          ,  0.          ,  0.          , ...,  0.          ,
               9.87138597,  0.          ],
 [ 0.          ,  0.          ,  0.          , ...,  0.          ,
               0.          , 10.81779121]])

```

```

[ ]:

```