Martin BRICENO
jbriceno@telecom-paris.fr

Lab session # 4
ALTEGRAD 2020

12/15/20

1 Question 1

Since our graph is composed of two connected components, we can analyze them separately.

- The number of edges in the complete graph of 100 nodes is determined with the following formula because it is a combinatorial analysis of the number of nodes in 2.

$$n\_edges = \frac{n*(n-2)}{2} = \frac{100*99}{2} = 4950 \tag{1}$$

- The number of edges in the complete bipartite graph of 50 nodes in each partition set is determined with the following formula

$$n\_edges = n*n = 50^2 = 2500 \tag{2}$$

- The total number of edges in the graph is:

$$\mathbf{n\_edges = 4950 + 2500 = 7450} \tag{3}$$

Since the bipartite graph has not triangles, we calculate only the number of triangles in the complete graph of 100 nodes, we calculate it as a combinatorial analysis of the number of nodes in 3.

$$n\_triangles = \frac{n*(n-1)*(n-2)}{6} = \frac{100*99*98}{6} \tag{4}$$

$$\mathbf{n\_triangles = 16170} \tag{5}$$

2 Question 2

Since the clustering coefficient is a relationship between the closed triplets and the total triplets in the graph. If we consider that all the triplets are closed, then **this coefficient will be 1 as maximum**.

Since we are looking for all the triples of the graph to be closed triplets, **we are talking about a complete graph**. This will give us the maximum value of clustering coefficient.

3 Question 3

We know that the smallest eigenvector for the laplace matrix L_{rw} is 0. If the graph is connected its corresponding eigenvector is v = $\mathbb{1}^T = (1,1,1,...,1)^T$ as we can see:

$$L_{rw}v = 0*v \tag{6}$$

$$(I - D^{-1}A)v = 0*v \tag{7}$$

If v = $\mathbb{1}^T$ and d is a vector of degrees:

$$(I - D^{-1}A)\mathbb{1}^T = 0*\mathbb{1}^T \tag{8}$$

$$\mathbb{1}^T - D^{-1}A\mathbb{1}^T = 0 \tag{9}$$

$$\mathbb{1}^T - D^{-1}d^T = 0, \tag{10}$$

$$\mathbb{1}^T - \mathbb{1}^T = 0, \tag{11}$$

The multiplicity of eigenvalue 0 equals the number of connected components. If vertices i and j are connected, then in a certain eigenvector corresponding to the eigenvalue zero, the ith and jth components are set to 1.

Then v = $\mathbb{1}^T$ is eigenvector of L_{rw} corresponding to the eigenvalue 0 and also v is of multiplicity 1 since the graph is connected.

Regarding the performance the intuition is that the smaller the eigenvalue, the fewer the edges between the two partitions. So Knowing that the graph is connected, the eigenvector associated to eigenvalue 0 does not provide more information, **so removing it from the spectral clustering algorithm is better, since we will apply kmeans over a set of points with a smaller dimension and we will prioritize the other eigenvectors corresponding to the other smaller eigenvalues.**

4 Question 4

The first stage of the algorithm is deterministic, the second stage is also deterministic, the third stage may include small calculation errors when computing the eigenvectors. However, the fourth stage introduces randomness to the algorithm, given that the results of Kmeans depend very much on its initialization and this is random, so it is normal that the **outputs (the cluster corresponding to each node) are relatively random (stochastic)** despite its objective of minimizing its score.

5 Question 5

We have:

$$Q = \sum^{n_c} \left[\frac{l_c}{m} - (\frac{d_c}{2m})^2 \right] \tag{12}$$

In both figures we have that $n_c = 2$ and $m = 13$.

- In figure a), we have $l_1 = 6$, $l_2 = 6$ and $d_1 = 13$, $d_2 = 13$

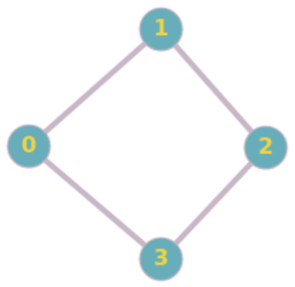
$$Q = \left[\frac{6}{13} - (\frac{13}{26})^2 \right] + \left[\frac{6}{13} - (\frac{13}{26})^2 \right] \approx 0.42 \tag{13}$$

- In figure b), we have $l_1 = 2$, $l_2 = 4$ and $d_1 = 11$, $d_2 = 15$

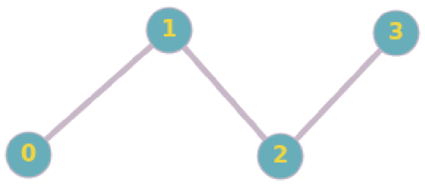
$$Q = \left[\frac{2}{13} - (\frac{11}{26})^2 \right] + \left[\frac{4}{13} - (\frac{15}{26})^2 \right] \approx -0.05 \tag{14}$$

6 Question 6

We have:



(a) Cyclic graph C4.



(b) Path graph P4.

To calculate the shortest path kernel, we need to compute the function $\phi(.)$. So, we need to compute the shortest path of each node in the graph.

- Computing the shortest path length for each node in C4 graph. We will use a vector of size equal the number of nodes in the graph, where each element of this vector indicates the number of the shortest path of size 0, 1, 2 and 3.

$$node0 = [1, 2, 1, 0]$$
$$node1 = [1, 2, 1, 0]$$
$$node2 = [1, 2, 1, 0]$$
$$node3 = [1, 2, 1, 0]$$

- Now $\phi(C4)$ is computed as the sum of this vectors.

$$\phi(C4) = [4, 8, 4, 0]$$

- Computing the shortest path length for each node in P4 graph as before.

$$node0 = [1, 1, 1, 1]$$

$$node1 = [1, 2, 1, 0]$$

$$node2 = [1, 2, 1, 0]$$

$$node3 = [1, 1, 1, 1]$$

- Computing $\phi(P4)$ with the same idea.

$$\phi(C4) = [4, 6, 4, 2]$$

- The shortest path kernel $SK(C4, C4)$ is computed as the dot product between $\phi(C4)$ and $\phi(C4)$.

$$\mathbf{SK(C4, C4)} = < \phi(\mathbf{C4}), \phi(\mathbf{C4}) > = \mathbf{96}$$

- The shortest path kernel $SK(C4, P4)$ is computed as the dot product between $\phi(C4)$ and $\phi(P4)$.

$$\mathbf{SK(C4, P4)} = < \phi(\mathbf{C4}), \phi(\mathbf{P4}) > = \mathbf{80}$$

- The shortest path kernel $SK(P4, P4)$ is computed as the dot product between $\phi(P4)$ and $\phi(P4)$.

$$\mathbf{SK(P4, P4)} = < \phi(\mathbf{P4}), \phi(\mathbf{P4}) > = \mathbf{72}$$