Matthieu Boyer
matthieu.boyer@ens.fr

Lab session # 4
ALTEGRAD 2023

11/16/25

Introduction

In this report, I have added the answers to the theoretical questions and the figures and numerical results from certain tasks. The tasks are completed in the different files of the `/code` folder of the archive, based on the given template.

Task 1

The graph has $9\,877$ nodes and $25\,998$ edges.

Task 2

The graph has 429 connected components.
The largest has $8\,638$ nodes and $24\,827$ edges, representing 87.46% of total nodes and 95.50% of total edges.

Question 1

G can be seen as the graph coproduct of the graphs K_{20} and $K_{10,10}$. The complement of $K_{10,10}$ is (considering only its vertices in G) the coproduct of K_{10} and K_{10} (as all vertices in each component become connected and we remove all edges between components). The complement of K_{20} is the graph $\bar{K}_{20}$ with 20 vertices and no edges. There are four possible cases when considering three vertices in the complement of G

1. At least two vertices are in $\bar{K}_{20}$, and there are no edges and thus no triangles.
2. There is at least one vertex in each copy of K_{10}, there are no edges between the two and there are no triangles.
3. The three vertices are in the same copy of K_{10}: there are $2 \times \binom{3}{10}$ such possibilities which all yield to a different triangle.
4. Two vertices are in the same copy of K_{10} and the last in in K_{20}: there are $2 \times \binom{2}{10} \times 20 = 1800$ such possibilities.

In the end, there are $\boxed{2040}$ triangles in the complement of G.

Question 2

Taking the gradient with respect to x the expression defining the Rayleigh quotient of G with matrix A, we get:

$$\nabla_x R(A, x) = \frac{\|x\|^2 \nabla \langle x, Ax \rangle - \langle x, Ax \rangle \nabla \|x\|^2}{\|x\|^4} = \frac{2\|x\|^2 Ax - 2(\langle x, Ax \rangle)x}{\|x\|^4}$$

since A is symmetric as G is undirected. The gradient cancels if and only if $\langle x, x \rangle Ax = \langle x, Ax \rangle x$. This means that the gradient vanishes precisely when x is an eigenvector of A and thus, a non-zero vector x is a stationary point of $R(A, \cdot)$ if and only if x is an eigenvector of A.

Task 4

If we try for 50 clusters in the giant connected component, $8\,638$ out of $8\,638$ nodes are assigned, with a minimum cluster size of 5, a maximum cluster size of $7\,619$ and an average cluster size of 172.76.

Question 3

Let's start by computing modularities for each cluster arrangement. The graph has $m = 13$ edges.

1. For the graph on the left, the blue cluster has 5 nodes with $l_c = 7$ inner edges and $d_c = 15$ for its degree sum. The orange cluster has 4 nodes with $l_c = 5$ inner edges and $d_c = 11$ for its degree sum. As such,

$$Q_1 = \left[\frac{7}{13} - \left(\frac{15}{26}\right)^2\right] + \left[\frac{5}{13} - \left(\frac{11}{26}\right)^2\right] = 0.41$$

2. For the graph on the right, the blue cluster has 3 nodes with $l_c = 2$ inner edges and $d_c = 8$ for its degree sum. The orange cluster has 6 nodes with $l_c = 7$ and 18 for its degree sum. As such,

$$Q_2 = \left[\frac{2}{13} - \left(\frac{8}{26}\right)^2\right] + \left[\frac{7}{13} - \left(\frac{18}{26}\right)^2\right] = 0.12$$

As the modularity is much larger in the first clustering (with $Q_2 \simeq \frac{Q_1}{4}$), it would seem that the first clustering has a better clustering structure.

Task 6

We ran the experiment 50 times for random clustering, the value below is the mean of the modularities. The modularity obtained by spectral clustering with $k = 50$ clusters is 0.2094. The modularity obtained by random clustering with $k = 50$ clusters is -0.0004. Spectral clustering thus produces significantly better community structure than random assignment.

Question 4

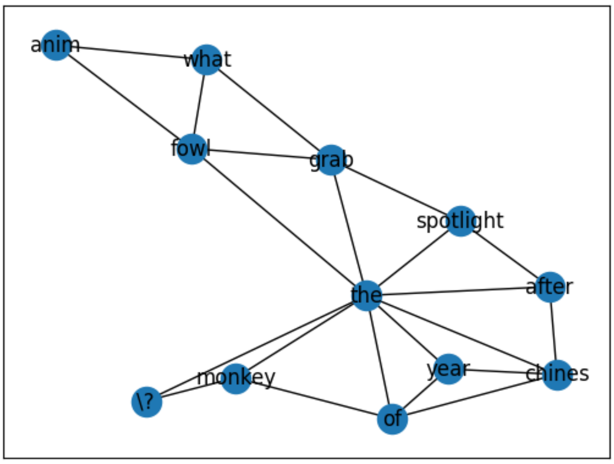
Considering edgeless graphs with different node numbers give us the result, as their number of shortest paths will always be zero for any length and they can't be isomorphic as they are different size.

Task 10

For the shortest path kernel we find an accuracy of 0.8947, whereas we get an accuracy of 0.6842 for the graphlet kernel. The difference in accuracy is 0.2105 or about 25% in relative error.

Task 11

Here is an example of a graph generated by the `create_graphs_of_words` function:



Task 13

The SVM classifier we trained gives an accuracy of 0.938.

Question 5

Initial Setup From Figure 5, we identify the graph structures:
Graph G: Nodes labeled $\{2, 1, 4, 5, 3, 1\}$ with edges: (2,1), (2,4), (2,5), (2,3), (1,5), (4,1), (4,3), (5,3)
Graph G': Nodes labeled $\{2, 1, 4, 5, 3, 4\}$ with edges: (2,1), (2,4), (2,5), (1,5), (4,3), (5,4), (5,3)

Step 1: Collect Neighbor Labels for Each Node **Graph G:**

- Node with label 2: neighbors = $\{1, 4, 5, 3\} \rightarrow$ signature: $(2, \{1, 3, 4, 5\})$
- Node with label 1 (top right): neighbors = $\{2, 4, 5\} \rightarrow$ signature: $(1, \{2, 4, 5\})$
- Node with label 4: neighbors = $\{2, 1, 3\} \rightarrow$ signature: $(4, \{1, 2, 3\})$
- Node with label 5: neighbors = $\{2, 1, 3\} \rightarrow$ signature: $(5, \{1, 2, 3\})$
- Node with label 3: neighbors = $\{2, 4, 5, 1\} \rightarrow$ signature: $(3, \{1, 2, 4, 5\})$
- Node with label 1 (bottom): neighbors = $\{4, 3\} \rightarrow$ signature: $(1, \{3, 4\})$

Graph G':

- Node with label 2: neighbors = $\{1, 4, 5\} \rightarrow$ signature: $(2, \{1, 4, 5\})$
- Node with label 1: neighbors = $\{2, 5\} \rightarrow$ signature: $(1, \{2, 5\})$
- Node with label 4 (left): neighbors = $\{2, 3\} \rightarrow$ signature: $(4, \{2, 3\})$
- Node with label 5: neighbors = $\{2, 1, 4, 3\} \rightarrow$ signature: $(5, \{1, 2, 3, 4\})$
- Node with label 3: neighbors = $\{4, 5, 4\} \rightarrow$ signature: $(3, \{4, 4, 5\})$
- Node with label 4 (bottom right): neighbors = $\{5, 3\} \rightarrow$ signature: $(4, \{3, 5\})$

Step 2: Generate New Labels Concatenate each node's label with its sorted neighbor labels.
Graph G:

- $(2, [1, 3, 4, 5]) \rightarrow$ "2-1-3-4-5"
- $(1, [2, 4, 5]) \rightarrow$ "1-2-4-5"
- $(4, [1, 2, 3]) \rightarrow$ "4-1-2-3"
- $(5, [1, 2, 3]) \rightarrow$ "5-1-2-3"
- $(3, [1, 2, 4, 5]) \rightarrow$ "3-1-2-4-5"
- $(1, [3, 4]) \rightarrow$ "1-3-4"

Graph G':

- $(2, [1, 4, 5]) \rightarrow$ "2-1-4-5"
- $(1, [2, 5]) \rightarrow$ "1-2-5"
- $(4, [2, 3]) \rightarrow$ "4-2-3"
- $(5, [1, 2, 3, 4]) \rightarrow$ "5-1-2-3-4"
- $(3, [4, 4, 5]) \rightarrow$ "3-4-4-5"
- $(4, [3, 5]) \rightarrow$ "4-3-5"

Step 3: Count Label Occurrences **Graph G label counts:** Each of the 6 labels appears exactly once.
Graph G' label counts: Each of the 6 labels appears exactly once.

Step 4: Compute Kernel Value The kernel value is the inner product of the label histograms (counting matching labels).
Common labels after 1 WL iteration: None of the refined labels match between G and G'.
Kernel value: $k(G, G') = 0$

Interpretation The kernel value of **0 indicates no structural similarity** between graphs G and G' after one WL iteration. Despite both graphs having 6 nodes with similar label distributions (two nodes labeled 1, two labeled 4, and one each of 2, 3, and 5), none of the nodes share the same local neighborhood structure after refinement. This means that the way nodes connect to their neighbors differs completely between the two graphs, revealing fundamentally different local topologies. The WL algorithm successfully distinguishes these two non-isomorphic graphs in just one iteration.

Task 14

I could not run the code due to time constraints on my computer.

References