

# Lab 4

Chahine Nejma

November 19, 2025

## Question 1

### 1. Constructing $\bar{G}$ (The Building Process)

We build the complement graph  $\bar{G}$  by inverting the edges of the original graph  $G$ . Imagine the transformation occurring in three zones:

- **$V_1$  Shatters (The Cloud):** The original solid  $K_{20}$  loses all internal connections. It becomes a "cloud" of **20 isolated vertices** (an Independent Set).
- **$V_2$  Solidifies (The Islands):** The original bipartite web disconnects the middle and fills in the sides. It transforms into **two separate, solid islands** (Cliques) of 10 vertices each ( $K_{10} \cup K_{10}$ ).
- **The Void Fills (The Bridge):** The previously empty space between  $V_1$  and  $V_2$  fills completely. Every single vertex in the "Cloud" is now connected to every single vertex in the "Islands."

### 2. Formula and Calculation

A triangle requires 3 connected vertices.

- We cannot choose 2 or more vertices from the "Cloud" ( $V_1$ ) because they don't touch.
- Therefore, we must anchor our triangles in the "Islands" ( $V_2$ ).

$$\text{Total} = \underbrace{n_1 \left[ \binom{10}{2} + \binom{10}{2} \right]}_{1 \text{ from Cloud, 2 from Islands}} + \underbrace{\left[ \binom{10}{3} + \binom{10}{3} \right]}_{\text{All 3 inside Islands}}$$

Substituting  $n_1 = 20$ :

$$\text{Total} = 20(45 + 45) + (120 + 120)$$

$$\text{Total} = 1800 + 240 = \mathbf{2040}$$

## Question 2

Since  $G$  is undirected,  $A$  is symmetric.

we have:  $\nabla(x^T Ax) = 2Ax$  and  $\nabla(x^T x) = 2x$ :

$$\nabla R(x) = \frac{2(x^T x)Ax - 2(x^T Ax)x}{(x^T x)^2}$$

Setting  $\nabla R(x) = 0$ :

$$Ax = \underbrace{\left( \frac{x^T Ax}{x^T x} \right)}_{\lambda} x \implies Ax = \lambda x$$

Thus,  $\nabla R(x) = 0 \iff x$  is an eigenvector of  $A$ .

article amsmath amssymb graphicx

## Question 3

First, we determine the total number of edges  $m$  and the degrees of all nodes in the underlying graph.

- **Degrees ( $d_i$ ):**  $d_1 = 3, d_2 = 3, d_3 = 4, d_4 = 2, d_5 = 5, d_6 = 4, d_7 = 2, d_8 = 3, d_9 = 2$ .
- **Total Sum of Degrees ( $2m$ ):**  $3 + 3 + 4 + 2 + 5 + 4 + 2 + 3 + 2 = 28$ .
- **Total Edges ( $m$ ):**  $m = 14$ .

### 1. Calculation for Graph (a)

The nodes are partitioned into two clusters: Blue ( $C_1$ ) and Orange ( $C_2$ ).

**blue cluster:**

- $l_1 : 8$ .
- $d_1 : 17$ .

**orange cluster:**

- $l_2 : 5$ .
- $d_2 : 11$ .

thus:

$$Q_a \approx 0.4056$$

### 2. Calculation for Graph (b)

The nodes are partitioned into two clusters: Blue ( $C_1$ ) and Orange ( $C_2$ ).

**blue cluster:**

- $l_1 : 2$ .
- $d_1 : 9$ .

**orange cluster:**

- $l_2 : 7$ .
- $d_2 : 19$ .

thus:

$$Q_b \approx 0.0791$$

- **Modularity of (a):**  $\approx 0.406$
- **Modularity of (b):**  $\approx 0.079$

Thus partition in **Figure 1(a)** is optimal.

## Question 4

The easiest examples of non-isomorphic graphs with the same shortest path kernel representation are the complete bipartite graph  $K_{3,3}$  and the Triangular Prism graph.

Their representation (the count of pairs for each shortest path length) is identical:

- **Length 1 (Edges):** Both graphs have 9 edges.
- **Length 2:** Both graphs have 6 pairs of vertices at distance 2.

Thus, the shortest path representation vector for both graphs is:

$$\Phi(G) = [9, 6, 0, \dots]$$

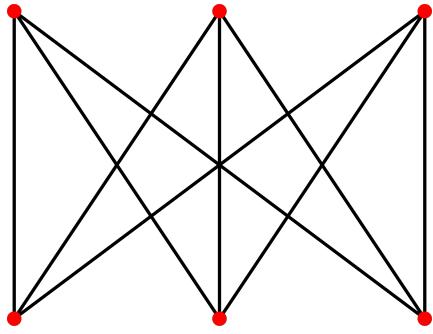


Figure 1:  $K_{3,3}$

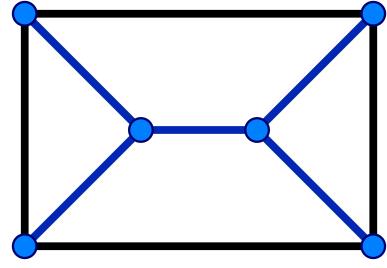


Figure 2: Triangular Prism

### Question 5

First, we identify the initial labels of nodes in both graphs and compare them.

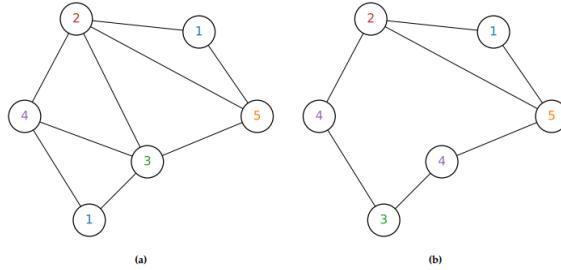


Figure 3: graphs  $G$  and  $G'$

#### Graph $G$ (Left):

- Nodes:  $\{1, 2, 3, 4, 5, 1\}$
- Label Count Vector  $\phi_0(G)$ :  $\{1 : 2, 2 : 1, 3 : 1, 4 : 1, 5 : 1\}$

#### Graph $G'$ (Right):

- Nodes:  $\{1, 2, 3, 4, 4, 5\}$
- Label Count Vector  $\phi_0(G')$ :  $\{1 : 1, 2 : 1, 3 : 1, 4 : 2, 5 : 1\}$

### Weisfeiler–Lehman Iteration (Aggregation and Relabeling)

For each node  $v$ , we collect the multiset of labels of its neighbors  $N(v)$ , sort them, and append them to the node's current label to form a signature string:  $s_v = (L_v, \text{sort}(\{L_u \mid u \in N(v)\}))$ .

#### Processing Graph $G$

Node Position	Current Label	Neighbor Labels	New Signature ( $L_{new}$ )
Top-Right	1	{2, 5}	1, 25
Top-Left	2	{1, 4, 3, 5}	2, 1345
Center	3	{2, 4, 1, 5}	3, 1245
Mid-Left	4	{2, 3, 1}	4, 123
Mid-Right	5	{1, 2, 3}	5, 123
Bottom	1	{4, 3}	1, 34

Node Position	Current Label	Neighbor Labels	New Signature ( $L_{new}$ )
Top-Right	1	{2, 5}	<b>1, 25</b>
Top-Left	2	{1, 4, 5}	<b>2, 145</b>
Mid-Right	5	{1, 2, 4}	<b>5, 124</b>
Mid-Left	4	{2, 3}	<b>4, 23</b>
Bottom	3	{4, 4}	<b>3, 44</b>
Bottom-Right	4	{5, 3}	<b>4, 35</b>

### Processing Graph G'

- Signatures in G: {1, 25, 2, 1345, 3, 1245, 4, 123, 5, 123, 1, 34}
- Signatures in G': {1, 25, 2, 145, 5, 124, 4, 23, 3, 44, 4, 35}

**Common Signatures:** There is only **one** matching signature between the two sets:

**1, 25**

(This corresponds to the Top-Right node labeled '1' in both graphs).

**Total Kernel Value (One Full Iteration):** The WL kernel value is the sum of matches at iteration 0 (original labels) and iteration 1 (new structural labels).

$$K_{WL}^{(1)}(G, G') = K_{h=0} + K_{h=1} = 5 + 1 = \mathbf{6}$$

### Interpretation of Structural Similarity

The kernel value reveals a sharp contrast between *what* the graphs are made of and *how* they are wired together. At a glance, the graphs seem nearly identical because they share the exact same building blocks, indicated by the high match count at the initial level. However, this similarity collapses when we look at the actual connections. The structural score drops drastically in the next step, showing that while the graphs use the same labels, those labels are arranged in completely different neighborhoods. The only exception is the node labeled '1': which connects to a '2' and a '5' in both cases; marking the only spot where the local structure is preserved.