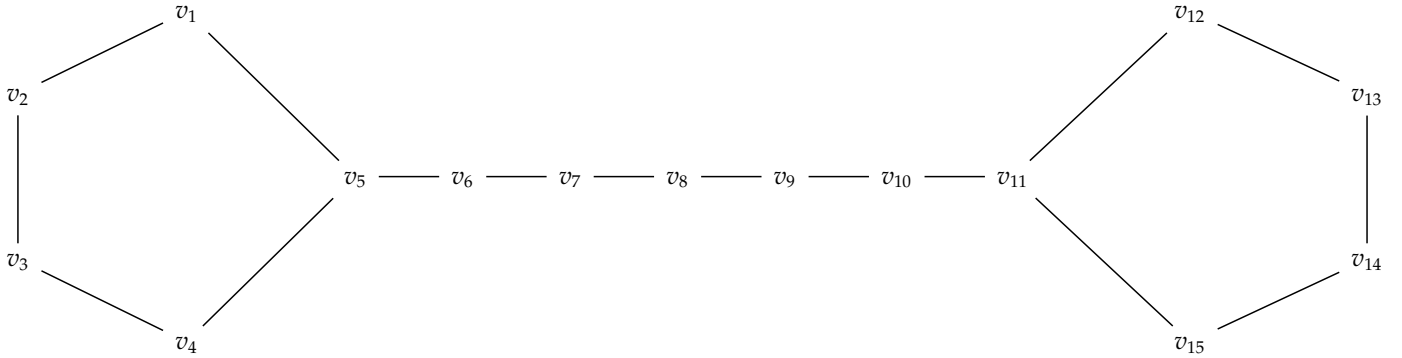


# CSE Template

Rohan Mukherjee

November 28, 2023

1. For the above graph, order the vertices as follows: the first vertex is right above the point connecting the line to the complete graph  $K_n$ , then proceed counterclockwise. Once returned to this connecting point, proceed down the line, and repeat clockwise on the second  $K_n$ . Example with  $n = 5$ : (I have omitted the rest of the edges in the complete graph in the picture but not in the following proofs)



With this setup, it is clear that  $D\mathbb{I} = (\underbrace{n, \dots, n}_n, \underbrace{2, \dots, 2}_n, \underbrace{n, \dots, n}_n)^T$ . Now, we consider the vector  $x = (\underbrace{1, \dots, 1}_n, \underbrace{1 - 2/n, \dots, -1 + 2/n}_n, \underbrace{-1, \dots, -1}_n)^T$ . Then, if  $\mathcal{L}$  is the normalized Laplacian of this graph, we have

$$\frac{x^T \mathcal{L} x}{x^T x} = \frac{\sum_{i \sim j} (x_i - x_j)^2}{\sum_i \deg(i) x_i^2}$$

We see that

$$\sum_{i \sim j} (x_i - x_j)^2 = 2 \times \binom{n}{2} \times 0 + \sum_{i=n+1}^{2n} \left(\frac{2}{n}\right)^2$$

The 2 comes from the 2 complete graphs, the  $\binom{n}{2}$  from the  $\binom{n}{2}$  possible pairs of vertices (all which have edges), and the 0 since  $x_i = x_j = 1$  for all vertices of the two complete graphs. The next sum is just evaluating the middle path. And also that,

$$\sum_i \deg(i) x_i^2 \geq 2 \cdot (n-1) \cdot (n \cdot 1^2) = 2n^2$$

Since the degree of every vertex in the complete graph is  $(n - 1)$ , we have  $n$  vertices in the complete graph, and each  $x_i = 1$  for the complete graph. We conclude that

$$\frac{x^T \mathcal{L} x}{x^T x} = \frac{n \cdot 4/n^2}{(n - 1)n}$$

And, since  $\langle x, D\mathbb{I} \rangle = 0$ , this tells us that  $\lambda_2 = \min_{\langle y, D\mathbb{I} \rangle = 0} \frac{y^T \mathcal{L} y}{y^T y} \leq O\left(\frac{1}{n^3}\right)$ . Intuitively, this proof is assigning positive values to vertices that should be partitioned into the left complete graph, and similarly negative values to those that should go with the right complete graph. This is likely optimal but I leave the verification of this fact to the reader.

2. Here is my code:

---

```
import numpy as np
from numpy.random import rand, randint, normal

eps = 0.01

def perfect_matching(n):
    vertices = [i for i in range(n)]
    A = np.zeros((n, n))
    while len(vertices) >= 2:
        vertex_1 = vertices[randint(0, len(vertices))]
        vertices.remove(vertex_1)
        vertex_2 = vertices[randint(0, len(vertices))]
        vertices.remove(vertex_2)
        A[vertex_1][vertex_2] = A[vertex_2][vertex_1] = 1
    return A

def d_regular(d, n):
    A = np.zeros((n, n))
    for i in range(d):
        A += perfect_matching(n)
    return A

def second_largest_eigenvalue(A):
    n = len(A[0])
    B = A @ A
    v1 = np.array([1/np.sqrt(n) for i in range(n)])
    v2 = normal(0, 1, n)
    v2 = v2 - (v1 @ v2) * v1
    for i in range(int(np.log2(n)/eps)):
        v2 = B @ v2
        v2 = v2 - (v1 @ v2) * v1
        v2 = v2 / np.linalg.norm(v2)
    return np.sqrt(v2 @ (B @ v2))
```

```

for d in range(3, 11):
    A = d_regular(d, 10000)
    print("d = ", d, ", n = 10000")
    print("Approximate second largest eigenvalue: ",
          second_largest_eigenvalue(A/d))
    print("True second largest eigenvalue: ",
          sorted(np.abs(np.linalg.eigvals(A/d)))[-2])

```

---

I learned recently that you can use @ for matrix multiplication instead of np.matmul(A, B), which was a LIFESAVER!

My results are as follows:

d = 3 , n = 10000

Approximate second largest eigenvalue: 0.9422257194941585

True second largest eigenvalue: 0.9423078725050605

d = 4 , n = 10000

Approximate second largest eigenvalue: 0.8650364072587419

True second largest eigenvalue: 0.8650586977793632

d = 5 , n = 10000

Approximate second largest eigenvalue: 0.7993390179651007

True second largest eigenvalue: 0.7993423285813461

d = 6 , n = 10000

Approximate second largest eigenvalue: 0.745164569985666

True second largest eigenvalue: 0.745167394756143

d = 7 , n = 10000

Approximate second largest eigenvalue: 0.7000123767558103

True second largest eigenvalue: 0.7000161811816583

d = 8 , n = 10000

Approximate second largest eigenvalue: 0.6603331791598479

True second largest eigenvalue: 0.6604691793990919

d = 9 , n = 10000

Approximate second largest eigenvalue: 0.627422373723629

True second largest eigenvalue: 0.6274648749934267

d = 10 , n = 10000

Approximate second largest eigenvalue: 0.6002963095846212

True second largest eigenvalue: 0.6004042904092797

The power method runs MUCH faster than finding the true eigenvalues! Like, 100 times faster!

3. First, using the lemma, we see that

$$|E(S, \bar{S})| \leq \frac{d \cdot |S| \cdot |\bar{S}|}{n} + d \cdot \lambda^* \sqrt{|S||\bar{S}|}$$

Now, by the AM-GM inequality, we have that

$$|S| \cdot |\bar{S}| \leq \left( \frac{|S| + |\bar{S}|}{2} \right)^2 \leq \frac{n^2}{4}$$

So we have that,

$$\begin{aligned} \max_S |E(S, \bar{S})| &\leq \frac{dn}{4} + \frac{dn\lambda^*}{2} = \frac{dn}{4}(1 + 2\lambda^*) \\ (1 - 2\lambda^*) \max_S |E(S, \bar{S})| &\leq \frac{\max_S |E(S, \bar{S})|}{1 + 2\lambda^*} \leq \frac{dn}{4} \end{aligned}$$

By the trivial bound  $1 - x \leq \frac{1}{1+x}$  for  $x > 0$ . Now, for any set  $T \subset V$  with  $|T| = n/2$ , we have, by the triangle inequality, that

$$|E(T, \bar{T})| \geq \frac{dn}{4} - \frac{dn\lambda^*}{2} = \frac{dn}{4}(1 - 2\lambda^*) \geq (1 - 2\lambda^*)^2 \max_S |E(S, \bar{S})| \geq (1 - 4\lambda^*) \max_S |E(S, \bar{S})|$$

Since  $(1 - 2\lambda^*)^2 = 1 - 4\lambda^* + 4\lambda^{*2} \geq 1 - 4\lambda^*$ . Our algorithm is this: If  $n$  is even return  $T =$  the first  $n/2$  vertices. Else, let  $T$  be any subset of  $V$  with size  $(n + 1)/2$  w.p.  $1/2$ , and size  $(n - 1)/2$  o.w. Now, since

$$\begin{aligned} |E(T, \bar{T})| &\geq \frac{d|T||\bar{T}|}{n} - d\lambda^* \sqrt{|T||\bar{T}|}, \\ \Rightarrow \mathbb{E} \left[ |E(T, \bar{T})| - \frac{d|T||\bar{T}|}{n} + d\lambda^* \sqrt{|T||\bar{T}|} \right] &\geq 0 \\ \Rightarrow \mathbb{E} [|E(T, \bar{T})|] &\geq \mathbb{E} \left[ \frac{d|T||\bar{T}|}{n} - d\lambda^* \sqrt{|T||\bar{T}|} \right] = \frac{dn}{4} - \frac{dn\lambda^*}{2} \\ &\geq (1 - 4\lambda^*) \max_S |E(S, \bar{S})| \end{aligned}$$

$E(T, \bar{T})$  is a discrete distribution whose expectation is  $(1 - 4\lambda^*) \max_S |E(S, \bar{S})|$ , so it either has something larger or some equal to  $(1 - 4\lambda^*) \max_S |E(S, \bar{S})|$  in it. This proves the odd case, and we are done.