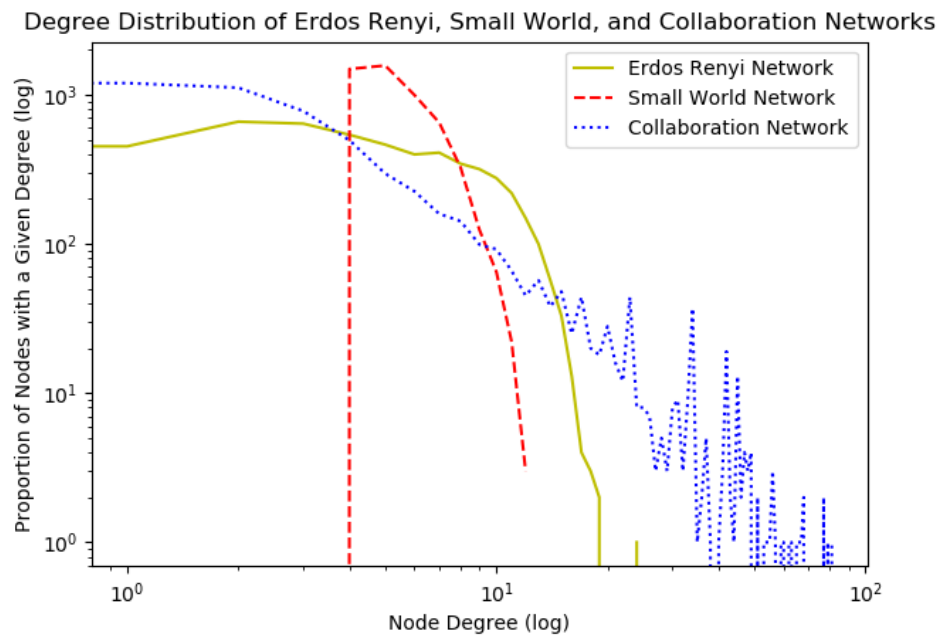


The log-log degree distribution plot is shown as belows. We can see that

- The Small World Network has the most narrow plot comparing to other two networks.
- Erdos Renyi Network is more similar with Collaboration Network, which means from the degree distribution perspective, Erdos Renyi models the real networks better.
- Compared to random graph networks, the degree distribution of Collaboration Network is more wider.
- It has more long-tail nodes which has large degree and small degree.



The average clustering coefficient for each network:

- Clustering Coefficient for Erdos Renyi Network: 0.001236
- Clustering Coefficient for Small World Network: 0.297775
- Clustering Coefficient for Collaboration Network: 0.529636

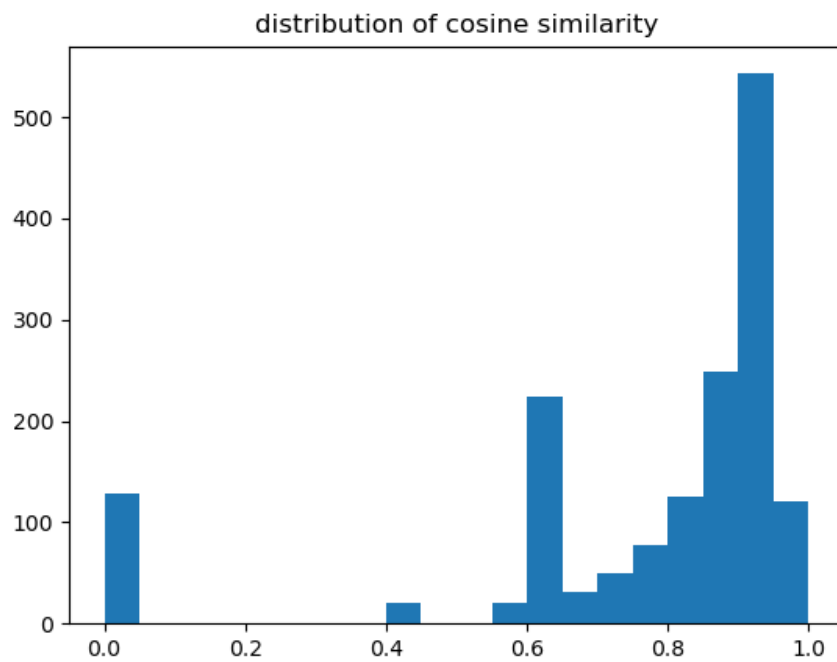
We can see that Collaboration Network has the largest average clustering coefficient.

Since in real life, authors are likely to establish collaboration with those who have collaborated with their collaborators, so the local structure of this subgraph tends to be dense, which results in a larger average clustering coefficient.

- Feature vector of node 9 is:  $[6, 4, 7]$
- Top 5 nodes at most similar to node 9 are:  $(415, 0.9987327913808193)$   $(286, 0.9970704162402443)$   $(288, 0.9970704162402443)$   $(16, 0.9950371902099892)$   $(17, 0.9950371902099892)$

- Top 5 nodes at most similar to node 9 are: (973, 0.9977857208153201) (415, 0.9967754809774014) (296, 0.9922005791814624) (189, 0.9908884045260696) (275, 0.9908884045260696)
- There is only one common node which is 415.
- Because features generated via aggregation also consider information about the neighbors and the neighbors of neighbors, which are more expressive than basic features.

Firstly, we draw the histograms as below:

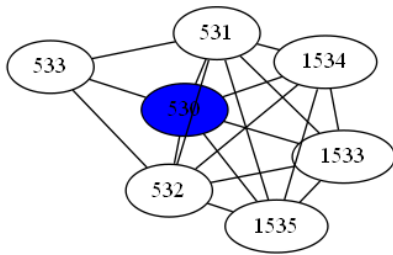


From the picture we can observe 4 groups/roles.

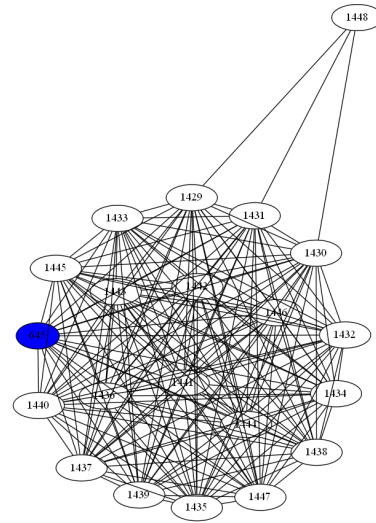
Question 2.3b, Homework 1, CS224W

---

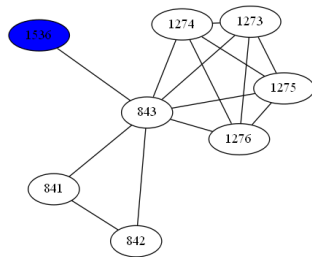
Randomly select a point for each group and draw the subgraph of its 2nd neighbors as below:



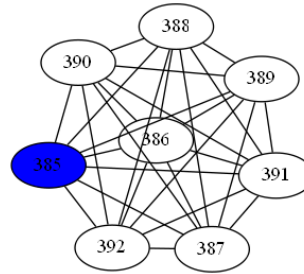
subgraph 1



subgraph 2



subgraph 3



subgraph 4

**Question 3.1, Homework 1, CS224W**

---

According to the formula of modularity, it's easy to get that, for all nodes in a community, e.g., C, the second part in the formula, i.e.,  $\frac{d_i d_j}{2m}$ , is the  $(\frac{\sum_i k_i}{2m})^2$ . So firstly, we can calculate the modularity of C and i before i joins in C.

$$Q_{before} = \frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2$$

Then we can calculate the modularity after i joins in C.

$$Q_{after} = \frac{(\sum_{in} + k_{i,in})}{2m} - (\frac{\sum_{tot} + k_i}{2m})^2$$

Finally, we can get the modularity gain.

$$\begin{aligned} \Delta Q &= Q_{after} - Q_{before} \\ &= [\frac{(\sum_{in} + k_{i,in})}{2m} - (\frac{\sum_{tot} + k_i}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2] \end{aligned}$$

For Graph  $H$ ,

- 1
- 6
- $Q = 4 * [\frac{12}{56} - (\frac{14}{56})^2] = \frac{17}{28} \approx 0.607$

For Graph  $J$ ,

- 2
- 13
- $Q = 2 * [\frac{26}{56} - (\frac{28}{56})^2] = \frac{3}{7} \approx 0.429$



### Question 3.3, Homework 1, CS224W

---

For Graph  $H_{big}$ ,

- 1
- 6
- $Q = 32 * [\frac{12}{448} - (\frac{14}{448})^2] = \frac{185}{224} \approx 0.826$

For Graph  $J_{big}$ ,

- 2
- 13
- $Q = 16 * [\frac{26}{448} - (\frac{28}{448})^2] = \frac{97}{112} \approx 0.866$

**Question 3.4, Homework 1, CS224W**

---

The definition of modularity focuses on the local structure of a cluster, which may lead to 'suboptimal' when we need to trade off the number of communities and the modularity of each community. This intuition is consistent with Fortunato and Barthelemy's work *Resolution limit in community detection*.

- (i) From the formula we can see that, for every edge from  $j$  to  $k$ , will contribute 1 to  $L_{j,j}$  and  $L_{k,k}$  and  $-1$  to  $L_{j,k}$  and  $L_{k,j}$ . It follows the way to compute  $L$ , i.e.,  $L = D - A$ .
- (ii)  $x^T Lx = \sum_i \sum_j x_i L_{i,j} x_j$ , so for each edge from  $i$  to  $j$ , will contribute  $x_i^2$  and  $x_j^2$  and  $-(x_i * x_j + x_j * x_i)$  to the final result, i.e.,  $(x_i - x_j)^2$ .
- (iii) We can calculate  $x^T Lx$  in three parts. For  $x_i$  and  $x_j$  in the same  $S$ ,  $x_i - x_j = 0$ . So

$$x^T Lx = \frac{\bar{S}}{S} + \frac{S}{\bar{S}} - 2 = cut(S) \frac{(vol(\bar{S}) + vol(S))^2}{2vol(\bar{S})vol(S)} = (vol(\bar{S}) + vol(S)) * NCUT(S)$$

- (iv)  $x^T Lx = \sum_i \sum_j x_i L_{i,j} x_j = \sum_i x_i \sum_j L_{i,j} x_j = \sum_i x_i * 0 = 0$
- (v)  $x^T Dx = \sum_i x_i^2 * D_{i,i} = vol(S) + vol(\bar{S}) = 2m$

Firstly, let  $z = D^{1/2}x$  and we can rewrite the optimization problem as

$$\begin{aligned} & \underset{z \in \mathbb{R}^n}{\text{minimize}} && \frac{z^T \tilde{L} z}{z^T z} \\ & \text{s.t.} && z^T D^{1/2} e = 0, z^T z = 2m \end{aligned}$$

As  $\tilde{L}$  is symmetric, so its eigenvectors are orthonormal and we can write  $z$  in this form

$$z = \sum_i w_i v_i$$

where  $w_i$ s are weights and  $v_i$ s are  $\tilde{L}$ 's eigenvectors. The optimization objective can also be written as

$$\frac{\sum_i w_i^2 \lambda_i}{2m}$$

where  $\lambda_i$ s are  $\tilde{L}$ 's eigenvalues. Besides, since  $\tilde{L} * D^{1/2} e = D^{-1/2} L D^{-1/2} * D^{1/2} e = 0 * e$ , so  $D^{-1/2}$  is the corresponding eigenvector of  $\tilde{L}$  with a eigenvalue 0, which means  $w_1$  has to be 0 according to the subject  $z^T D^{1/2} e = 0$ . So to minimize the objective, we need to put all weights in the second smallest eigenvalue and thus we have  $z = (2m)^{1/2} v$  as the minizer, where  $v$  is the corresponding eigenvector. Finally, we get

$$x = (2m)^{1/2} D^{-1/2} v$$

$$\begin{aligned}
 Q(y) &= \frac{1}{2m}(-2\text{cut}(S) + \text{vol}(S) + \text{vol}(\hat{S}) - \frac{1}{2m}(\text{vol}(S)^2 + \text{vol}(\hat{S})^2)) \\
 &= \frac{1}{2m}(-2\text{cut}(S) + 2m - \frac{1}{2m}(\text{vol}(S)^2 + \text{vol}(\hat{S})^2)) \\
 &= \frac{1}{2m}(-2\text{cut}(S) - \frac{1}{2m}(\text{vol}(S) + \text{vol}(\hat{S}))^2 - \text{vol}(S)^2 + \text{vol}(\hat{S})^2)) \\
 &= \frac{1}{2m}(-2\text{cut}(S) - \frac{1}{m}\text{vol}(S)\text{vol}(\hat{S}))
 \end{aligned}$$

# Information sheet

## CS224W: Machine Learning with Graphs

**Assignment Submission** Fill in and include this information sheet with each of your assignments. This page should be the last page of your submission. Assignments are due at 11:59pm and are always due on a Thursday. All students (SCPD and non-SCPD) must submit their homework via GradeScope (<http://www.gradescope.com>). Students can typeset or scan their homework. Make sure that you answer each (sub-)question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code on Gradescope. Put all the code for a single question into a single file and upload it.

**Late Homework Policy** Each student will have a total of *two* late periods. *Homework are due on Thursdays at 11:59pm PT and one late period expires on the following Monday at 11:59pm PT.* Only one late period may be used for an assignment. Any homework received after 11:59pm PT on the Monday following the homework due date will receive no credit. Once these late periods are exhausted, any assignments turned in late will receive no credit.

**Honor Code** We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently, i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (GitHub/Google/previous year's solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

**Your name:** \_\_\_\_\_

**Email:** \_\_\_\_\_ **SUID:** \_\_\_\_\_

Discussion Group: \_\_\_\_\_

I acknowledge and accept the Honor Code.

(Signed) \_\_\_\_\_