CS224W: Analysis of Networks **Fall 2018**

Homework 1

Due 11:59pm PDT October 11, 2018

This problem set should be completed individually.

General Instructions

These questions require thought, but do not require long answers. Please be as concise as possible. You are allowed to take a maximum of 1 late period (see the information sheet at the end of this document for the definition of a late period).

Submission instructions: You should submit your answers via Gradescope and your code via the SNAP submission site. Register for Gradescope at `http://gradescope.com` using your Stanford e-mail (if not SCPD) and include your student ID number with sign-up. Use the entry code **9ZZ2XY** to sign up for CS224W.

Submitting answers: Prepare answers to your homework in a single PDF file and submit it via Gradescope. Make sure that the answer to each sub-question is on a *separate, single page*. The number of the question should be at the top of each page. Please use the submission template files included in the bundle to prepare your submission. Failure to use the submission template file will result in a reduction of 2 points from your homework score.

Information sheet: Fill out the information sheet located at the end of this problem set or at the end of the submission template file, and sign it in order to acknowledge the Honor Code (if typesetting the homework, you may type your name instead of signing). This should be the last page of your submission. Failure to fill out the information sheet will result in a reduction of 2 points from your homework score.

Submitting code: Upload your code at `http://snap.stanford.edu/submit`. Put all the code for a single question into a single file and upload it. Failure to submit your code will result in reduction of all points for that part from your homework score.

Homework survey: After submitting your homework, please fill out the Homework 1 Feedback Form. Respondents will be awarded extra credit.

Questions

1 Network Characteristics [20 points – Alex Wang]

One of the goals of network analysis is to find mathematical models that characterize real-world networks and that can then be used to generate new networks with similar properties. In this problem, we will explore two famous models—Erdős-Rényi and Small World—and compare them to real-world data from an academic collaboration network. Note that in this problem all networks are *undirected*. You may use the starter code in q1-starter.py for this problem.

- *Erdős-Rényi Random graph ($G(n, m)$ random network):* Generate a random instance of this model by using $n = 5242$ nodes and picking $m = 14484$ edges at random. Write code to construct instances of this model, i.e., do not call a SNAP function.

- *Small-World Random Network:* Generate an instance from this model as follows: begin with $n = 5242$ nodes arranged as a ring, i.e., imagine the nodes form a circle and each node is connected to its two direct neighbors (e.g., node 399 is connected to nodes 398 and 400), giving us 5242 edges. Next, connect each node to the neighbors of its neighbors (e.g., node 399 is also connected to nodes 397 and 401). This gives us another 5242 edges. Finally, randomly select 4000 pairs of nodes not yet connected and add an edge between them. In total, this will make $m = 5242 \cdot 2 + 4000 = 14484$ edges. Write code to construct instances of this model, i.e., do not call a SNAP function.
- *Real-World Collaboration Network:* Download this undirected network from http://snap.stanford.edu/data/ca-GrQc.txt.gz. Nodes in this network represent authors of research papers on the arXiv in the General Relativity and Quantum Cosmology section. There is an edge between two authors if they have co-authored at least one paper together. Note that some edges may appear twice in the data, once for each direction. Ignoring repeats and self-edges, there are 5242 nodes and 14484 edges. (Note: Repeats are automatically ignored when loading an (un)directed graph with SNAP's `LoadEdgeList` function).

1.1 Degree Distribution [10 points]

Generate a random graph from both the Erdős-Rényi (i.e., $G(n, m)$) and Small-World models and read in the collaboration network. Delete all of the self-edges in the collaboration network (there should be 14,484 total edges remaining).

Plot the degree distribution of all three networks *in the same plot* on a log-log scale. In other words, generate a plot with the horizontal axis representing node degrees and the vertical axis representing the proportion of nodes with a given degree (by "log-log scale" we mean that both the horizontal and vertical axis must be in logarithmic scale). In one to two sentences, describe one key difference between the degree distribution of the collaboration network and the degree distributions of the random graph models.

1.2 Clustering Coefficient [10 points]

Recall that the local clustering coefficient for a node v_i was defined in class as

$$C_i = \begin{cases} \frac{2|e_i|}{k_i \cdot (k_i - 1)} & k_i \geq 2 \\ 0 & \text{otherwise,} \end{cases}$$

where k_i is the degree of node v_i and e_i is the number of edges between the neighbors of v_i. The *average clustering coefficient* is defined as

$$C = \frac{1}{|V|} \sum_{i \in V} C_i.$$

Compute and report the average clustering coefficient of the three networks. For this question, write your own implementation to compute the clustering coefficient, instead of using a built-in SNAP function.

Which network has the largest clustering coefficient? In one to two sentences, explain. Think about the underlying process that generated the network.

What to submit

Page 1:
- Log-log degree distribution plot for all three networks (in same plot)
- One to two sentence description of a difference between the collaboration network's degree distribution and the degree distributions from the random graph model.

Page 2:
- Average clustering coefficient for each network.
- Network that has the largest average clustering coefficient.
- One to two sentences explaining why this network has the largest average clustering coefficient.

2 Bowtie Structure of Non-Web Networks [25 points – Jayadev Bhaskaran]

In this problem, we explore the structure of a directed social network, namely the Epinions Social Network (dataset and more information available at `http://snap.stanford.edu/data/soc-Epinions1.html`) and a communication network, namely the EU Email Communication Network (dataset and more information available at `http://snap.stanford.edu/data/email-EuAll.html`). Working out way through this question, we will observe that the structure of these networks resembles the bowtie structure of the web graph.

We will use methods similar to the ones Broder et al. employed in their seminal paper where they determined that the web graph is structured like a bowtie. The authors discovered that the web graph (Figure 1) had a large strongly connected component (SCC) which could be reached from any node in IN, and could go to any node of OUT. There were also TENDRILS hanging off IN and OUT, containing nodes reachable from portions of IN or nodes going to portions of OUT. TENDRILS going from IN to OUT without touching SCC formed TUBES. There are also some DISCONNECTED components isolated from the rest of the graph.

Note: You can use any SNAP functions for any part of this question. The comments within the starter code (q2-starter.py) contain some functions that could be helpful, but feel free to use anything else too.

2.1 Node Position [5 points]

Using outgoing BFS (breadth first search following outgoing edges from a given node) and incoming BFS (the same, but using incoming edges to a given node), how would you determine whether a given node lies in SCC, IN or OUT?

Consider the node with ID 2018 in the Email graph, and the node with ID 224 in the Epinions graph. Run outgoing BFS and incoming BFS on these two nodes (in their respective graphs) and determine whether they lie in SCC, IN or OUT. (*Hint*: You may want to use the SNAP function `GetBfsTree`.)

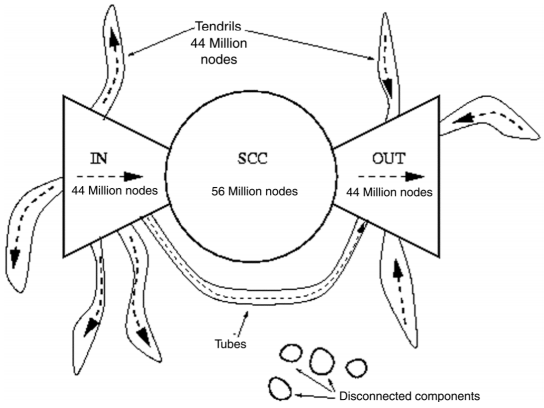

Figure 1: Bowtie structure of the web, as described in Broder et al., showing SCC, IN, OUT, TENDRILS, TUBES and DISCONNECTED components of the graph.

2.2 Random-start BFS [8 points]

For each of the two networks, choose 100 nodes at random and do one forward and one backward BFS traversal for each node. Plot the cumulative number of the nodes reached in these BFS runs, like in the paper by Broder et al. (Figure 2 below). Create one figure for the forward BFS and one for the backward BFS (you should have a total of 4 figures, 2 for each network). Based on the figures, what can you say about the relative size of SCC, IN and OUT in each network? Explain in a few lines.

Hint: You may want to use the SNAP function `GetRndNId` *to get random node IDs.*

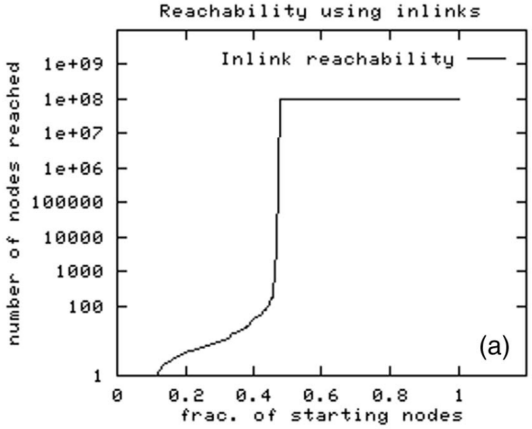

Figure 2: Cumulative distribution on the number of nodes reached by incoming BFS started from randomly chosen nodes.

2.3 Size of Bowtie Regions [7 points]

We will now try to determine the sizes of each region in the network. How many nodes are in the SCC, IN, OUT, TENDRILS+TUBES (referring to TENDRILS and TUBES combined), and DISCONNECTED regions of each of the two networks? Describe how you calculate the size of

each of the above components in a few lines. (*Hint*: You may want to use the SNAP functions `GetMxWcc` and `GetMxScc` along with BFS.)

2.4 Probability of a Path Existing Between Two Randomly Chosen Nodes [5 points]

Broder et al. found in their paper that given a pair of randomly chosen start and finish webpages, one can get from the start page to the finish page by traversing links only approx 25% of the time. For each of the Epinions and the Email networks, what is the probability that a path exists between two nodes chosen uniformly from the graph?
(*Hint*: One way you can approach this question is to first sample many node pairs at random and then report the fraction of times pairs were reachable. You may find the SNAP function `GetShortPath` useful here.)
As the number of node pairs sampled grows larger, what would you expect the fraction of reachable pairs to converge to?

What to submit

Page 4:
- Explanation for how to determine whether a node lies in SCC, IN or OUT
- For each of the two nodes, whether they lie in SCC, IN or OUT

Page 5:
- Four plots (cumulative number of nodes reached for incoming and outgoing BFS for each of two networks)
- 1-2 sentence explanation (in terms of relative sizes of SCC, IN and OUT) of the observed BFS traversal behavior

Page 6:
- Size of SCC, IN, OUT, TENDRILS+TUBES, DISCONNECTED regions for each of the two graphs
- Explanation for how you computed the size of each component

Page 7:
- Results of experiments performed on at least 100 nodes: probability of a path existing between a pair of nodes chosen from the entire graph, for each of the two networks.
- 1-2 sentence explanation on expected probability as number of sampled nodes grows large.

3 Link Analysis [20 points – Megha Jhunjhunwala]

3.1 Personalized PageRank I [7 points]

Personalizing PageRank is a very important real-world problem: different users find different pages relevant, so search engines can provide better results if they tailor their page relevance estimates to the users they are serving. Recall from class that PageRank can be specialized with clever modifications of the teleport vector. In this question, we will explore how this can be applied to personalize the PageRank algorithm.

Assume that people's interests are represented by a set of representative pages. For example, if Zuzanna is interested in sports and food, then we could represent her interests with the set of pages {`www.espn.com`, `www.epicurious.com`}. For notational convenience, we will use integers as names for webpages.

Suppose you have already computed the personalized PageRank vectors for the following users:

- Agatha, whose interests are represented by the teleport set $\{1, 2, 3\}$
- Bertha, whose interests are represented by the teleport set $\{3, 4, 5\}$
- Clementine, whose interests are represented by the teleport set $\{1, 4, 5\}$
- Dolo, whose interests are represented by the teleport set $\{1\}$

Assume that the weights for each node in a teleport set are uniform. Without looking at the graph, can you compute the personalized PageRank vectors for the following users? If so, how? If not, why not? Assume a fixed teleport parameter β.

i. **[2 points]** Eloise, whose interests are represented by the teleport set $\{2\}$.

ii. **[2 points]** Felicity, whose interests are represented by the teleport set $\{5\}$.

iii. **[3 points]** Glynnis, whose interests are represented by the teleport set $\{1, 2, 3, 4, 5\}$ with weights 0.1, 0.2, 0.3, 0.2, 0.2, respectively.

3.2 Personalized PageRank II [3 points]

Suppose that you've already computed the personalized PageRank vectors of a set of users (denote the computed vectors V). What is the set of all personalized PageRank vectors that you can compute from V without accessing the web graph?

3.3 SimRank in Citation Networks I [5 points]

SimRank is a link-based similarity measure that is based on the philosophy that "two objects are considered to be similar if they are referenced by similar objects". It is very commonly applied on citation graphs to compute the similarity of academic papers.

For a given citation graph $G = (V, E)$ where V represents the set of papers and $E \in V \times V$ is a set of a paper-pair (a, b), $S(a, b)$, is computed as follows:

$$S(a,b) = \begin{cases} 1, & a = b \\ \frac{C}{|I_a||I_b|} \sum_{i \in I_a} \sum_{j \in I_b} S(i,j), & a \neq b \end{cases} \tag{1}$$

where I_a denotes a set of papers directly citing paper a, $|I_a|$ is the size of I_a, and $C \in (0, 1)$ is a damping factor. If $I_a = \phi$ or $I_b = \phi$, $S(a, b) = 0$. Equation 1 is a recursive formula started by $S_0(a, b) = 1$ if $a = b$; $S(a, b) = 0$ otherwise. For successive iterations $k = 1, 2, ...$, we have

$$S_k(a,b) = \begin{cases} 1, & a = b \\ \frac{C}{|I_a||I_b|} \sum_{i \in I_a} \sum_{j \in I_b} S_{k-1}(i,j), & a \neq b \end{cases} \tag{2}$$

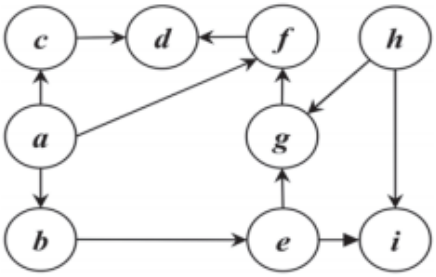

Figure 3: A citation network.

In similarity computation, SimRank does not only consider the papers directly citing a and b but also considers the papers indirectly citing them.

Consider paper-pairs (b, c) and (g, i) in the citation graph shown in Figure 3.

3.3 (a) [3 points]

Express the SimRank score of (b, c) and (g, i) in terms of C. (where C is the damping constant).

3.3 (b) [2 points]

The number of common papers citing b and c (i.e., only a) is less than that for g and i (i.e., e and h). Do you think this is consistent with the SimRank scores computed in part (a)? Explain your answer.

3.4 SimRank in Citation Networks II [5 points]

3.4 (a) [3 points]

Express the SimRank score for the paper pair (c, e) in terms of C. (where C is the damping constant). Do you think the SimRank score makes sense for the paper pair (c, e)? Explain in 1-2 sentences.

3.4 (b) [2 points]

Express the SimRank score for the paper pair (e, h) in terms of C. Explain how this can be a problem and suggest modifications to solve this.

What to submit

Page 8:
- For each of (i),(ii), and (iii), 'yes' or 'no' and a brief explanation of why or why not.

Page 9:
- A mathematical expression for the set in terms of V and a brief explanation.

Page 10:
- (part a) Expressions for $S(b, c)$ and $S(g, i)$
- (part b) A 'yes' or 'no' and a brief explanation for why or why not.

Page 11:
- (part a) Expression for $S(c, e)$
- (part a) A 'yes' or 'no' and a brief explanation for why or why not.
- (part b) Expression for $S(c, h)$
- (part b) One or two sentences stating the problem with the SimRank score of (e, h) and a modification to solve it.

4 Network Characteristics [35 points – Alex Haigh]

An interesting - and useful - application of the network inference tools discussed in lecture is to model the interactions between genes and diseases; doing so can help identify clusters of related genes (which might previously have been unknown) and the genetic basis of certain diseases.

This homework problem is inspired by the seminal paper "The Human Disease Network" (Goh et al., 2007: http://www.pnas.org/content/104/21/8685) and uses the dataset from http://www.disgenet.org/ds/DisGeNET/results/all_gene_disease_associations.tsv.gz. This dataset contains a list of diseases and the genes that those diseases are associated with. The source of the dataset is DisGeNET, one of the largest publicly available collections of genes and variants associated with human diseases. If you are interested in how this data was collected, check out this paper: https://academic.oup.com/nar/article/45/D1/D833/2290909.

Note that reading these papers is not necessary to complete the exercise, and no prior biology knowledge is necessary.

4.1 The Gene-Disease Network [7 points]

Using the dataset linked to above, construct the bipartite Gene-Disease Network using the geneId and diseaseId columns (i.e., columns 1 and 3 of the dataset). Formally, the Gene-Disease Network can be defined as a bipartite graph where the nodes are the set of all diseases and genes, and there is an edge between gene x and disease y if there is an association between the two documented in the dataset.

Hint: SNAP libraries only accept integer Node IDs, so you will need to modify the disease ID category when loading the graph. Also, make sure that there are no genes with the same ID as a disease: they are separate identification schemes in the dataset, but node IDs must be unique in SNAP. If there are overlapping IDs, you will lose the bipartite structure since one node will be both a disease and a gene.

Report the following statistics on the graph:

- How many nodes are there in total? How many diseases? How many genes?
- How many edges are there?
- Plot the degree distribution of genes and the degree distribution of diseases *on the same plot* on a log-log scale. In one to two sentences, describe one key difference or similarity between the degree distribution of the genes and the degree distributions of the diseases.

4.2 The Human Disease Network [8 points]

The bipartite structure of the Gene-Disease Network allows us to analyze the relationships between genes or between diseases by creating a network projection. In the "Human Disease Network" (HDN), nodes represent diseases, and two diseases are connected to each other if there is at least one gene in which mutations are associated with both diseases. In other words, two nodes in the network projection are connected if they have at least one common neighbor.

Formally, the HDN is a graph $G'(V', E')$ with V' = the set of all diseases, and there is an edge (i, j) between diseases i and j if there is a gene y such that $(i, y) \in G$ and $(j, y) \in G$ (where G is the original Gene-Disease Network).

Construct the HDN using the definition above, and report the following statistics on the graph:

- How many nodes are in the graph? How many edges are in the graph?
- What is the density of the graph? What is the average clustering coefficient C? For this problem, you may use the built-in SNAP function to calculate C. Computing the exact clustering coefficient is computationally expensive, so you can instead compute it over a sample of the nodes.

Note: Folding the graph will take a long time to run (on the order of an hour) due to the graph size and the computational complexity of the algorithm

4.3 Cliques in the Human Disease Network [5 points]

While the HDN provides a potentially invaluable tool to analyze the relationships between human diseases, it is difficult to analyze in practice because the graph has many large cliques. Answer the following questions:

- Why do these cliques arise?
- In the general case, what does the size k_{max} of the largest clique in the HDN represent?
- Calculate k_{max} in the HDN created in part (4.2).

The first two questions should require no more than a 1-2 sentence answer each.

4.4 Edge Contraction in the HDN [8 points]

We can make the HDN more usable by combining cliques into single "supernodes" using an edge contraction algorithm. Edge contraction removes an edge from a network while simultaneously merging the two vertices that the edge previously joined. Formally, merging vertices i and j on a graph $G(V, E)$ leaves us with a single vertex i' that has an edge (i', x) $\forall x \mid (i, x) \in G$ or $(j, x) \in G$.

Apply this algorithm to the HDN network to contract cliques with more than $k = 250$ nodes (iteratively remove edges in the clique until the clique is a single supernode). Report the following statistics of the graph:

- How many nodes are in the graph? How many edges are in the graph?
- What is the density of the graph? What is the average clustering coefficient C? How does it compare to the clustering of HDN before contraction?
  For the clustering coefficient, you may use the built-in SNAP function on this problem. Computing the exact clustering coefficient is computationally expensive, so you can instead compute it over a sample of the nodes.

Hint: Use your insight from part (4.3) to efficiently determine cliques of size $> k$. A brute force algorithm to find all cliques will have a very long runtime on a graph of this size.

4.5 Disease Similarity in the Gene-Disease Network [7 points]

Recall the two node "similarity" measures used in class: Common-Neighbors (CN) and Jaccard Index (JA, also known as Intersection over Union, or IoU). For two nodes x and y, the metrics are defined as

$$CN(x, y) = |\Gamma_x \cap \Gamma_y|$$
$$JA(x, y) = \frac{|\Gamma_x \cap \Gamma_y|}{|\Gamma_x \cup \Gamma_y|}$$

where Γ_i is the set of neighbors of node i.

We can apply these metrics to the Gene-Disease Network and identify diseases that have a similar genetic basis. This further deepens our understanding of the biological basis of disease, and, clinically, could aid in both discovering new drug targets as well as identifying diseases that doctors could treat with existing drugs (e.g. a drug already on the market for disease X might successfully treat disease Y if Y and X are "similar," since they'd have a similar genetic cause).

For Crohn's Disease (diseaseID = C0010346) and Leukemia (diseaseID = C0023418), report the top 5 most similar diseases by both metrics as well as their scores. In 1-2 sentences, explain which metric provides a better set of similarity scores and why. **Reminder: these similarities are based on the original, unfolded Gene-Disease Network, not the folded HDN.**

Fun Fact: Crohn's disease, as well its 3 most "similar" diseases, are all treatable with the same drug!

What to submit

Page 12:
- Number of Nodes in the Gene-Disease Network.
- Number of Edges in the Gene-Disease Network.
- Number of Genes and number of Diseases in the Gene-Disease Network.
- Log-log degree distribution plot for genes and diseases (treated separately but on the same plot).
- One to two sentence analysis of the similarity or difference in degree distribution of genes and diseases.

Page 13:
- Number of nodes and edges in the HDN.
- Density of the HDN.
- Clustering coefficient of the HDN.

Page 14:
- One to two sentence explanation of why the HDN has many large cliques.
- One to two sentence explanation of the meaning of k_{max} in the HDN.
- k_{max} for the HDN created in (4.2)

Page 15:
- Number of nodes and edges in the contracted HDN.
- Density of the contracted HDN.
- Clustering coefficient of the contracted HDN.

Page 16:
- 5 diseases most similar to Crohn's Disease (C001346) by both the Common-Neighbors and Jaccard Index scores.
- 5 diseases most similar to Leukemia (C0023418) by both the Common-Neighbors and Jaccard Index scores.
- One sentence explanation of which of the two metrics proved more useful.

Information sheet
CS224W: Analysis of Networks

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 homeworks via Gradescope (http://www.gradescope.com). Students can typeset or scan their homeworks. 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 at http://snap.stanford.edu/submit. Put all the code for a single question into a single file and upload it. Please do not put any code in your Gradescope submissions.

Late Homework Policy Each student will have a total of *two* free late periods. *Homework are due on Thursdays at 11:59pm PDT and one late period expires on the following Monday at 11:59pm PDT.* Only one late period may be used for an assignment. Any homework received after 11:59pm PDT 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 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) _____