The University of Texas at Austin
Chandra Department of Electrical
and Computer Engineering
Cockrell School of Engineering

Fall 2023

ECE 381K: Machine Learning on Real World Networks

HW1: Assigned 09/12/23, DUE 09/26/23 (midnight - 11:59:59pm CDT)

Problem 1 (15 pts)

a) Consider 10 isolated nodes. For each pair of nodes, throw a (fair) die and connect them if the number on the die is 1. Describe the graph you obtain. Is it connected or not? What is the average degree and the degree distribution? Are there any cycles?

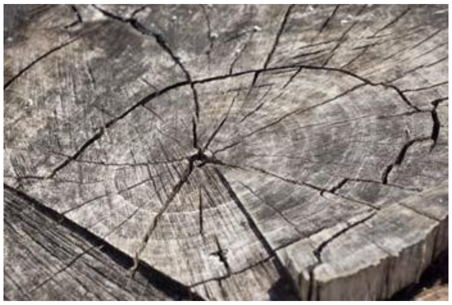
Now repeat the experiment and connect nodes if the number on the die is 1 or 2. How is the graph different from the previous case?

b) Install the following software tool: Gephi: http://gephi.org/ on your computer. Use it to draw and analyze the graphs you obtained in part a), *i.e.*, report the number of unreachable pairs of nodes, average distance among reachable pairs of nodes, network diameter, clustering coefficient. Provide a screenshot of your work.

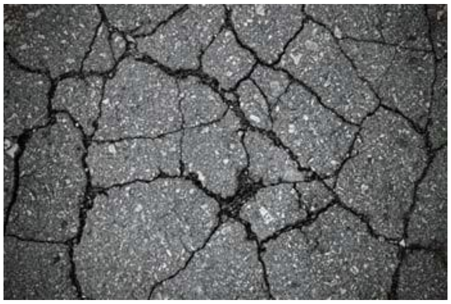
c) Now let us build and analyze a few real-world networks. More specifically, let us look at a few microstructures of crack lines and compare how these propagate through different materials.

Using the images[1] below, model the fractures of the wood, asphalt, and glass as three different networks in Gephi using the *Edge* and *Node Pencil* tools. To this end, you should model material fractures as a network, *i.e.*, a node represents the intersection of two or more cracks, while a link represents the crack itself. Do not get too caught up on matching each intersection to the images, but rather add enough edges such that you can identify the origin (i.e., wood, asphalt, or glass) from the network alone.

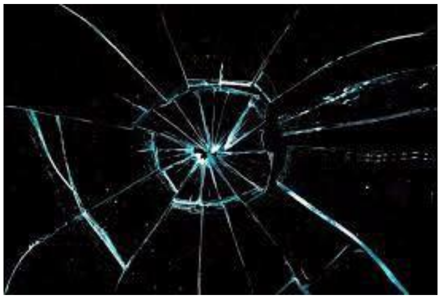
After you build the network based on these images, determine various network properties (*i.e.*, average degree, path length, connected components, and clustering coefficient). Compare the properties across materials and provide possible explanations as to how these network properties may or may not reflect the material mechanical properties (*e.g.*, plasticity, brittleness, stiffness, *etc.*).



wood



asphalt



glass

[1] To build the Gephi network, use the higher quality JPG images available on Canvas under the HW1 folder.

Problem 2 (10 pts)

Using Gephi, plot the network of loans (as directed arcs) between 16 Renaissance Florentine families given in the text file *Renaissance.net*[2]

a) What is the degree distribution of the network?

b) What is the average distance between the nodes of the largest strongly connected component? How does this compare to the network diameter?

c) What are the most important nodes in terms of total degree and betweenness? Illustrate one case where these two statistics are different and explain why.

Problem 3 (20 pts)

File "arXiv_lcc.gml" is an undirected network of scientific collaborations between authors who submitted to General Relativity and Quantum Cosmology.

a) Using Gephi: Visualize and calculate the average degree, average path length, diameter, and average clustering coefficient of this network.

b) Using Python or any programming language: Plot the degree distribution using both probability density function (PDF) and rank frequency plot (both in log-log scale).

c) Using Python or any programming language: Generate 5,000 exponentially distributed numbers with a mean identical to the average degree. Draw the rank frequency plot in log-log scale and contrast it to the result in part (b). What kind of network is this collaboration network? Justify your answer.

Problem 4 (25 pts)

Based on the concepts of characteristic path length and clustering coefficient learned in class, study the impact of increasing randomness in the network structure using a regular lattice with $N = 100$ nodes and connectivity $k = 4$ (*i.e.*, each node has 4 direct links to its neighbors as described below). More precisely, to introduce some randomness into the regular lattice proceed as follows:

- Consider first a regular lattice where N=100 nodes are distributed uniformly over a circle; each node is directly connected to its first and second immediate neighbors, that is, $k = 4$ as the “braid” model discussed in class. Note, that the circle is not divided into 16 parts as in the lecture notes, but instead in 100 parts.
- Starting from any node, iterate through each edge that connects that node to its first and second order neighbors. For each edge, re-wire this edge with a probability p to a new node chosen uniformly at random over the entire circle of nodes. If the two nodes are already connected, then leave the edge as it was. Iterate through all the nodes clockwise and stop iterating when each node is considered exactly once.
- Using a logarithmic scale, vary the probability p between 0.0001 and 1, with increasing increments (e.g., 0.0001, 0.0005, 0.001, 0.005, 0.01, etc.). On the *same* graph, plot the normalized values of the characteristic path length and clustering coefficient, that is $L(p)/L(0)$ and $C(p)/ C(0)$ similar to what we discussed in class. Explain the results you obtain.

[2] The *Renaissance.net* file can be found in the HW1 folder and contains a network of 16 nodes and 20 edges.

Problem 5 (30 pts)

Based on what we have learned in class, study the characteristics of an Erdos-Renyi (ER) graph and a Scale-Free (SF) network generated as described below.

Scale-free: Given N = 1,000 nodes and a preferential attachment mechanism for linking nodes, construct a SF graph as follows:

- Initially, choose m = 5 seed nodes and randomly connect them. Make sure you do not add self-loops and multiple edges between any two nodes.
- For each iteration, pick a node from the remaining (N - m) nodes and connect the new node to at most m nodes with a probability defined in equation (1). More precisely, the probability that a link of the new node will connect to node i is proportional with its degree k_i; this probability can be written as:

$$P(i) = \frac{k_i}{\sum_{j=1}^{N} k_j} \qquad (1)$$

Note that we consider only undirected links so if i and j are connected, then both entries $A(i,j)$ and $A(j,i)$ in the adjacency[3] matrix are set to 1. Do not add self-loops to the generated graph.

- Continue this process until all the initial nodes are linked at least once to the growing network.

Erdos-Renyi: Generate the ER graph as follows:

- Start with N = 1,000 isolated nodes.
- At each iteration, pick two random nodes and if they are not already connected, then link them with probability $p = 0.05$.
- Run the process until the ER graph reaches e edges, where e is the number of edges in the SF graph.

Save the two adjacency matrices as “ER.txt” and “SF.txt” files, respectively. Answer the following questions:

a) Using Gephi: Compute the degree distribution of the two networks and explain their properties (*e.g.*, maximum and minimum degree, presence/absence of hubs).

b) Estimate and plot the empirical PDF of the degree distributions for the two networks. Use linear-linear and log-log coordinates, as appropriate. Use any software tool to display the two networks.

c) Compute the clustering coefficients for the two networks.

d) For both graphs remove now 2% of the nodes with the highest degree. Re-compute the clustering coefficients for both graphs and compare the results with the previously computed coefficients. Explain what you observe.

Note:

* For Problem 3(b) and (c), Problem 4, and Problem 5, you can use Python or any other software you may find useful. If you are using MATLAB, the following library may prove very helpful: http://strategic.mit.edu/downloads.php?page=matlab_networks. To install MATLAB, use the UT Software Service to request a license and download MATLAB. https://ut.service-now.com/sp?id=ut_bs_service_detail&sys_id=f9d65c7c4ff9d200f6897bcd0210c77d. The MATLAB installation time is significant (a couple of hours or so) so plan accordingly.

***** Use the codes like *ave_path_length.m*, *clus_coeff.m* directly to get the desired results. You are welcome to find similar libraries for Python (*e.g.*, *networkX*).

[3] The adjacency matrix A of a finite undirected graph of N vertices and E edges is a matrix of size NxN, where each row corresponds to a distinct node and the non-zero entries in a row denote an edge between two nodes. For instance, if $A(i,j)$=1 then it means that there is an undirected edge between nodes i and j. Note that we do not allow self-loops in these graphs so the diagonal elements of $A(i,i)$ = 0. Moreover, the sum of all the elements in the upper triangular part of the adjacency matrix A should equal the total number of edges E in the graph.

***** For all problems in this homework, everything is handled electronically. Prepare the answers using either Word or Latex and ***create a single PDF file for the submission***. Also, put the source code and all related files for each problem (as needed) in a separate folder.

***** Finally, compress everything and name it as *"yourEID_hw1.zip"*, and deposit it on the Canvas under Assignments > HW1. In this zip file, include the write-up describing your solution and organize all the relevant files to each problem under a separate folder labeled suggestively Problem 1, Problem 2, …, Problem5.

Work individually on all problems.

Good luck!