AMATH 563 Spring 2018 Homework 4
Introduction to Networks

Tun Sheng Tan

Abstract. This assignment focuses on analyzing *C.elegans* neural networks. We are able to calculate the diameter of the networks and perform simple simulation with the networks.

1. Introduction. This assignment explores simple concept in network/graph theory by working on an example. In section 2, We introduce the definition of graph and network and also some basic properties of a network. In section 3, we discuss the implementation to find the shortest path in a graph. In section 4, we perform some simple calculation on *C.elegans* networks and try to use Kuramoto oscillators a simulation for the network.

2. Theoretical Background.

2.1. Graph. According to Wolfram MathWorld, a graph is a collection of points and lines connecting some subset of them [2]. A vertex (or node) refers a point on a graph. An edge refers to the lines connecting two vertices on a graph. The most basic graphs are undirected graph, which consists of vertices and edges, and directed graph, which consists of vertices and edges with a unique direction. Figure 1 shows an example of undirected graph, directed graph and network. In this assignment, we will be working with networks. A network is a undirected/directed graph with real numbers assigned to each edge.

Here are some basic properties of a network.

Definition 2.1. *Size of a network refers to the number of vertices.*

Definition 2.2. *Density of a network refers to the ratio of the number of edges to the number of possible edges in a network in a network of size N.*

Definition 2.3. *Diameter of a network refers to the longest shortest path in a network.*

Definition 2.4. *Degree of a node in a network is the number of connections it has to other nodes.*

Definition 2.5. *Degree distribution is the probability distribution of the degrees of a node over the whole network.*

One of the most important algorithm for graph theory is Dijkstra's algorithm. The algorithm enables one to compute the shortest paths between nodes in a graph. The algorithm branches outward from the source node and considers every node that is closer in distance until it reaches the destination node.

2.2. Kuramoto Oscillator. Kuramoto model is a collection of N oscillators with intrinsic frequency ω_i obeying the following equations:

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N}\sin(\theta_j - \theta_i) \qquad i = 1, ..., N \tag{2.1}$$

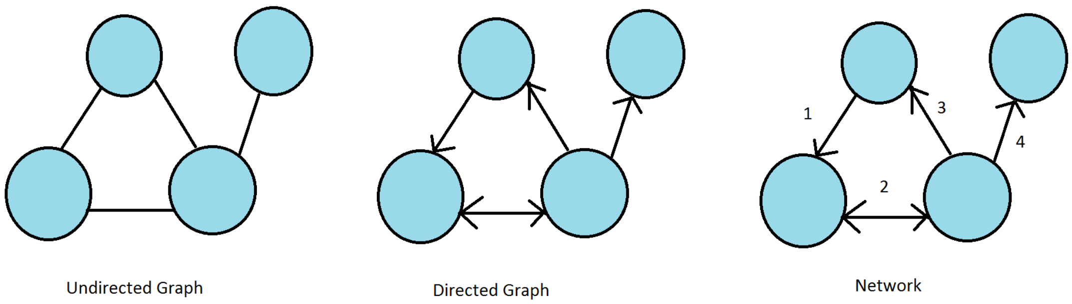

Figure 1. *Example of undirected, directed graphs and network.*

Algorithm 3.1 Pseudocode for Dijkstra's algorithm

```
Set all distances dist to be infinity
Set distance to source vertex, dist[s] to be zero
Define S to be the list of visited vertices to be empty list.
Define Q to be the queue containing all vertices.
while Q is not empty do
    Set u to be min distance between Q and dist
    Add u to list S.
    for v ∈ neighbors of u do
        if dist[v] > dist[u] + length(u, v) then
            dist[v] = dist[u] + length(u, v)
        end if
    end for
end while
return dist
```

where K is the coupling strength and θ_i is the phase at the i^{th} oscillator.

This model exhibits three different behaviors depending on the coupling strength. For small coupling, the oscillators are independent of each other. For large coupling, we will observe phase locking where all the oscillators evolve with the same frequency.

3. Algorithm Implementation and Development.

3.1. Diameter calculation. To calculate the shortest path, we will use the **distances** function from Matlab which implements Dijkstra's algorithm. The pseudocode is shown in Algorithm 3.1. Then, the diameter is the longest shortest path in the network.

3.2. Degree of distribution. To calculate the degree of distribution, we will use the **degree** function from Matlab which computes the sum of rows(columns) for each node.

4. Computational Results. Caenorhabditis elgans or commonly known as *C.elegans* is a nematode that lives in soil environment. It is the first multicellular organism that has been

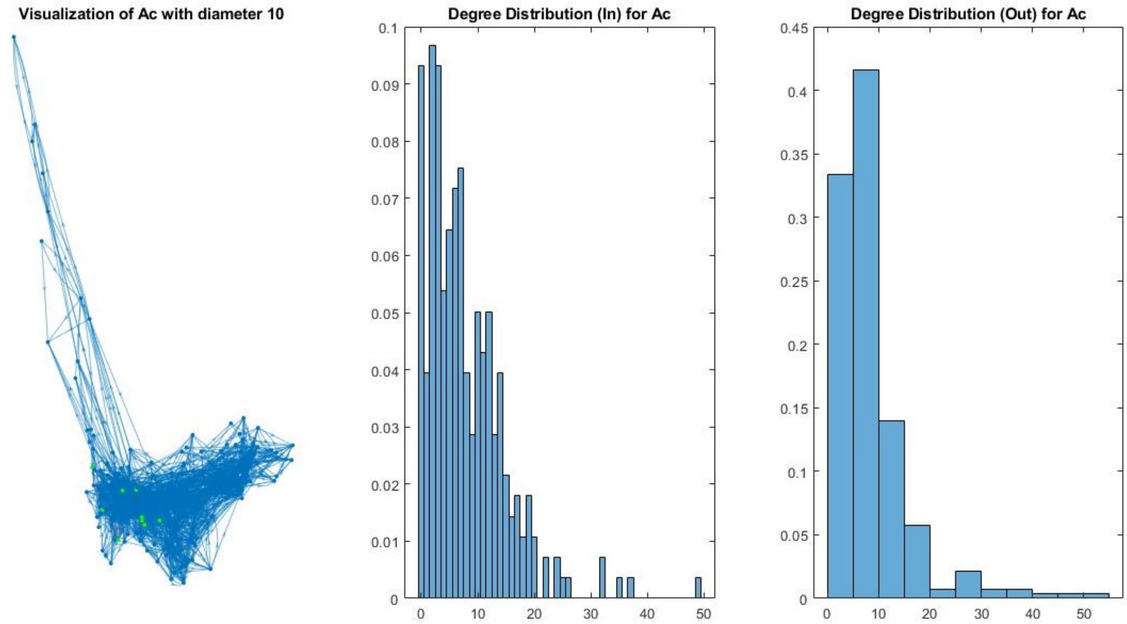

Figure 2. *(Left) Visualization of the chemical synaptic network. Green dots represents the pass through nodes for the diameter of the network. (Mid) Degree of distribution into node. (Right) Degree of distribution out of node.*

genome sequenced by scientists. C.elegans is popular in neuroscience because its nervous system is simple with only 302 neurons.

In this assignment, we will be working with the gap junction (L) and chemical synaptic (Ac) data of *C.elegans*. The gap junctions are electrical synapses. They form a undirected network while the chemical synaptic forms a directed network.

We computed the degree of distribution for each node in the network and the diameter of the network. We found that the diameter for the network Ac is 10 and most nodes have connections of 10 or less as shown in Figure 2. For network L, we found that the diameter is 12 and most nodes has connections less than 8 as shown in Figure 3.

Next, we make every node on the networks a Kuramoto oscillator and observe their behavior at different coupling strength ($K/N = 1/279,\ 100/279,\ 1$). When we refer to small coupling we mean $K/N = 1/279$ and large coupling refers to $K/N = 1$ which is different from standard convention. When $K \to \infty$, synchronization will occur regardless which is not interesting. We are compare which network is more sensitive to coupling. For network Ac, we observe that at weak coupling, there is no synchronization occurring. But from intermediate to high coupling strength, we observe synchronization between connected nodes as shown in Figure 4. For network L, we observe no synchronization in for any coupling strength as shown in Figure 4. Based on that we can see that, the chemical synaptic network achieve synchronization more easily than the gap junction network. Network Ac is more correlated than network L under Kuramoto model.

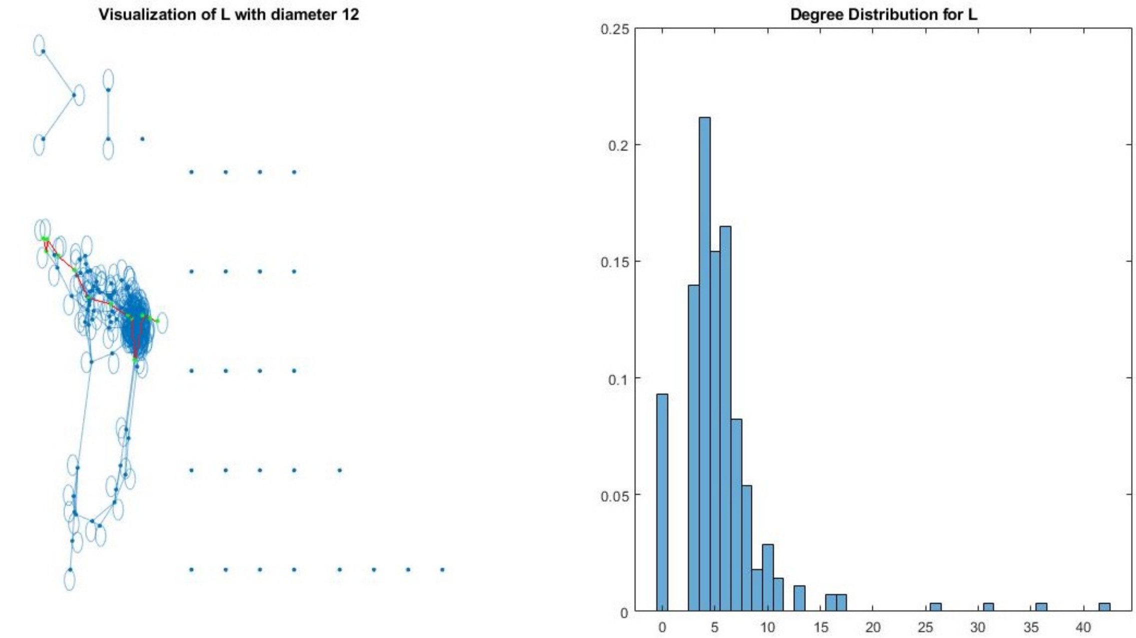

Figure 3. *(Left) Visualization of the gap junction network. Green dots represents the pass through nodes for the diameter of the network. Red line represents the path for diameter. (Right) Degree of distribution of network L.*

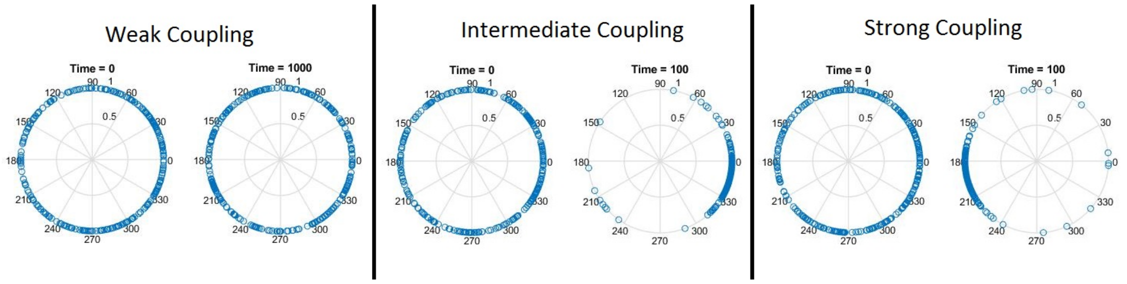

Figure 4. *Behavior of chemical synaptic network, Ac as a collection of Kuramoto oscillators at different coupling strength.*

5. Summary and Conclusions. We are able to compute basic properties of a network. In addition, we are able to run simulation on the network with a simple Kuramoto model. However, there are more information that we are not able to infer from the network. Even with a simple simulation, we had a hard time interpreting the result and not sure whether the simulation produces any meaningful that are interpretable by us.

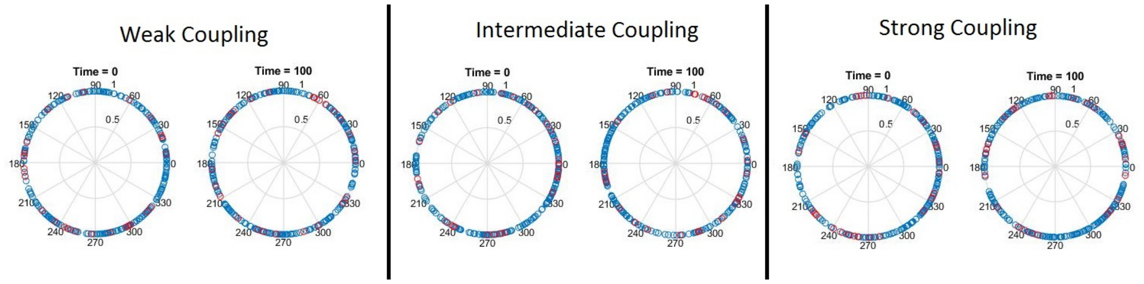

Figure 5. *Behavior of gap junction network, L as a collection of Kuramoto oscillators at different coupling strength. Red circles represent isolated nodes.*

Appendix A. Important Matlab functions used and brief explanation [1].

1. **graph** - return the graph object given adjacency matrix
2. **distances** - return the shortest path between all pairs of nodes in a graph.
3. **(out/in)degree** - return degree of distribution for directed/undirected graph
4. **ode45** - return the solution of a given differential equation

Appendix B. MATLAB codes.

B.0.1. Main Script.

1. Main Script

```
% Amath 563 Homework 4
% Tun Sheng Tan
% Due: June 6 2018

% 1.  Compute the following
%        (a) the degree distribution for each node in the network
%            (for both the synaptic and gap junction connectivity graphs)
%        (b) compute the c. elegans network diameter for both
%            synaptic and gap junctions

% 2.  Make each node (neuron) on the network a Kuramoto oscillator
%     and run a ?virtual? model of the worm to see what happens
%     with the given gap junction and synaptic connectivity graphs.


% Given files:
% 1.  Kuramoto oscillator code  (2 files)
%    (one for the dynamics and another is the right hand side function)
% 2.  The connectivity graph structure of the c. elegans worm.

% The fields of interest:
% 1. 'Ac' is the adjacency matrix for the chemical synaptic network.
%     Directed and weighted.
% 2. 'L' is the Laplacian matrix for the gap junction network
%     -- should be able to just zero the diagonal and
%     flip the sign to get the adjacency back.
%     Undirected and weighted.
```

```
%% Load Data
clear all; close all; clc
load('prams.mat');

%% Computing diameter of a network
% Definition:  the diameter of a network as the longest of all the
%              calculated shortest paths in a network

dAc = distances(digraph(Ac ~= 0));
dL  = distances(graph(L ~= 0));

dAc(dAc == inf) = 0; % Removing disconnected Nodes
dL(dL == inf) = 0;   % Removing disconnected Nodes

diameterA = max(max(dAc)); % Diameter of a network
diameterL = max(max(dL));   % Diameter of a network

%% Degree Distribution Computation
% Reference: https://mathinsight.org/degree_distribution

% Definition: Degree of a node in a network is the number of
%             connections it has to other nodes.
% Definition: Degree distribution is the probability distribution of
%             the degrees of a node over the whole network.

% For chemical synaptic network (directed)
% [m,n] = size(Ac);
% degreeNodeIn = zeros([ m 1]);
% degreeNodeOut = zeros([ m 1]);
% for i=1:m
%     degreeNodeIn(i) = sum(Ac(i,:) ~= 0);
%     degreeNodeOut(i) = sum(Ac(:,i) ~= 0);
% end

% For gap junction network (non-directed)
% [m,n] = size(Ac);
% degreeNode = zeros([ m 1]);
% for i=1:m
%     degreeNode(i) = sum(L(i,:) ~= 0);
% end

figure;
subplot(1,3,1);
h1=plot(digraph(Ac));
[I1,J1] = ind2sub([279, 279], find(dAc==diameterA));
path1 = shortestpath(digraph(Ac~=0),I1(1),J1(1));
highlight(h1,path1,'EdgeColor','r','NodeColor','g');
title("Visualization of Ac with"+" diameter "+diameterA)

subplot(1,3,2);histogram(outdegree(digraph(Ac)),'Normalization',...
    'probability'); title("Degree Distribution (In) for Ac");
subplot(1,3,3);histogram(indegree(digraph(Ac)),'Normalization',...
    'probability'); title("Degree Distribution (Out) for Ac");
```

```
figure;
subplot(1,2,1);
[I,J] = ind2sub([279, 279], find(dL==diameterL));
h = plot(graph(L));
[path2,d] = shortestpath(graph(L~=0),I(1),J(1));
highlight(h,path2,'EdgeColor','r','NodeColor','g');
title("Visualization of L with diameter "+diameterL);

subplot(1,2,2);histogram(degree(graph(L)),'Normalization','probability');
title("Degree Distribution for L");



%% Make each node (neuron) on the network a Kuramoto oscillator

t=0:1:100;

K=100;  % coupling strength
n=279; % number of oscillators
rad=ones(n,1);
thetai=2*randn(n,1);
omega=rand(n,1)+0.5;

[t,y]=ode45('kura_rhs',t,thetai,[],omega,n,K,Ac);
figure;
subplot(1,2,1);polar(y(1,:).',rad,'o');title("Time = 0");
subplot(1,2,2);polar(y(1000,:).',rad,'o');title("Time = 1000");

% for j=1:length(t)
%         ynow=y(j,:);
%         polar(ynow.',rad,'o');
%         drawnow
% end
```

B.1. Helper Functions.

1. Kuramoto differential equation for **ode45** written by Nathan Kutz

```
function rhs=kura_rhs(t,theta,dummy,omega,n,K,A)

coupling=zeros(n,1);
for j=1:n
  C=0;
  for jj=1:n
    C=C+A(jj,j)*sin(theta(jj)-theta(j));
  end
  coupling(j,1)=C;
end

rhs=omega+(K/n)*coupling;
```

REFERENCES

[1] MATLAB and S. Toolbox, 9.3.0.713579 (R2017b).
[2] E. W. Weisstein, *Graph*. http://mathworld.wolfram.com/Graph.html, 2018.