

1. (70 pts total) The impact of community structure on spreading processes.

- (a) (10 pts) The “planted partition” model is a one-parameter version of the stochastic block model that generates simple synthetic networks with community structure of varying strength.

Let n be a large and even number of vertices, let every vertex have a constant mean degree $c \geq 0$, and let $q = 2$ be the number of equal-sized communities in the model. If we define the probability of an edge existing within a group as $p_{in} = c_{in}/n$ and the probability of an edge existing between two groups as $p_{out} = c_{out}/n$, then the identity $2c = c_{in} + c_{out}$ is implied.

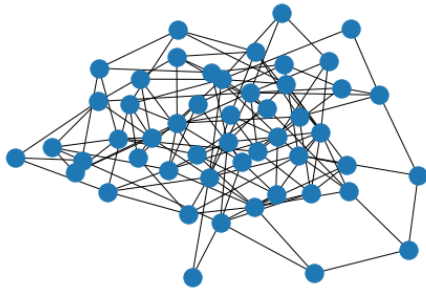
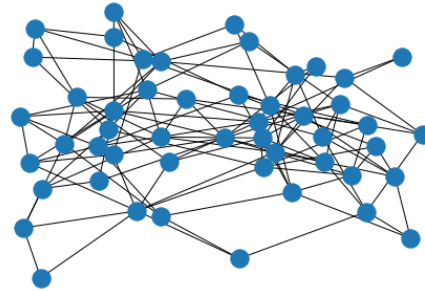
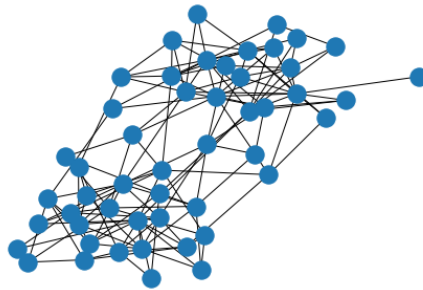
- Derive expressions for p_{in} and p_{out} in terms of only constants, c , n , and the parameter $\epsilon = c_{in} - c_{out}$, and hence show that this is a one parameter model.

$$\begin{aligned}
 2c &= c_{in} + c_{out} \\
 \therefore c_{out} &= c_{in} - 2c \\
 \epsilon &= c_{in} - c_{out} = 2c_{in} - 2c \\
 \therefore c_{in} &= \frac{2c + \epsilon}{2} \\
 p_{in} &= \frac{c_{in}}{n} = \frac{2c + \epsilon}{2n}
 \end{aligned}$$

$$\begin{aligned}
 2c &= c_{in} + c_{out} \\
 \therefore c_{in} &= 2c - c_{out} \\
 \epsilon &= c_{in} - c_{out} = 2c - 2c_{out} \\
 \therefore c_{out} &= \frac{2c - \epsilon}{2} \\
 p_{out} &= \frac{c_{out}}{n} = \frac{2c - \epsilon}{2n}
 \end{aligned}$$

Thus, $p_{in} = (2c + \epsilon)/2n$ and $p_{out} = (2c - \epsilon)/2n$. This makes this a one parameter model.

- Generate and create simple visualizations of three graphs, for $n = 50$, $q = 2$, $c = 5$, and $\epsilon = \{0, 4, 8\}$. Comment on the strength of the community structure each of these graphs exhibits.

Graph when $\epsilon = 0$ Graph when $\epsilon = 4$ Graph when $\epsilon = 8$ 

For $\epsilon = 0$, there no community structure. Any given node is equally likely to connect to any node. There is no clear distinction between the two groups due to this. Visually, there are no distinct communities in the graph.

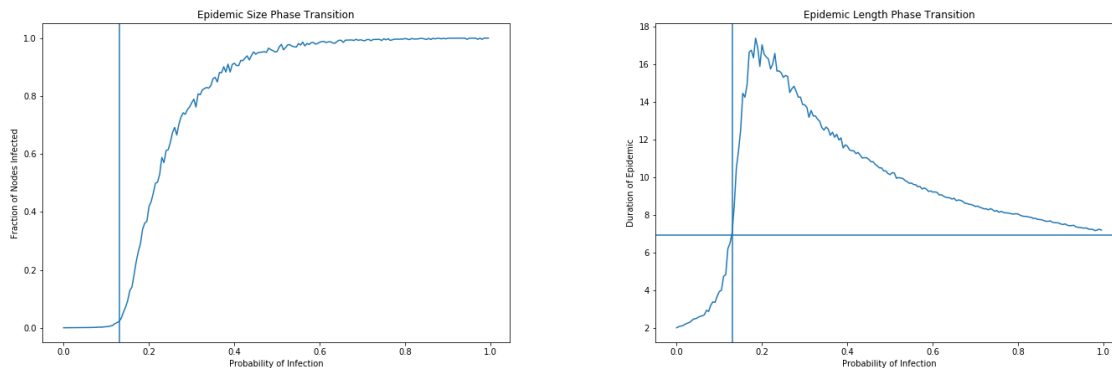
For $\epsilon = 4$, there is some community structure. The two groups are on the sides of the graph. They have more connections to each other than the other group. However there are still a decent number of connections between the two groups.

For $\epsilon = 8$, there is strong community structure. The nodes are highly connected in two groups. Causing the two connected components on the graph. However, the groups have some connections between them. However, the community structure can be clearly seen visually.

(b) Consider a simple discrete time SI spreading process.

Using the planted partition model from part (1a), you can generate any number of synthetic networks to use as a substrate for studying the behavior of the above simple SI spreading process. Using $n = 1000$, $c = 8$, and $\epsilon = 0$, measure the average epidemic size $\langle s/n \rangle$ and epidemic length $\langle \ell \rangle$, as a function of $p \in [0, 1]$.

- Make two figures, showing these measured relationships. On the length figure, include a horizontal line showing $\langle \ell \rangle = \log(n)$. On both figures, include a vertical line at the “critical value” of p .
- Comment on the qualitative behavior of these measured relationships as a function of the transmission probability p , and discuss your results relative to your expectations.
- Give an estimate p_{crit} and an intuitive explanation of why this value is special.

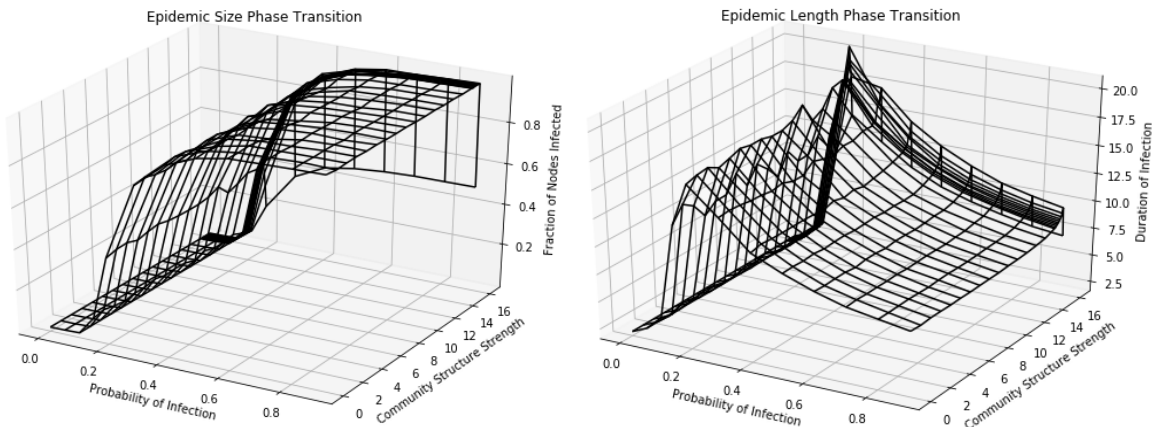


The epidemic size and length are similar to my expectation. For smaller probabilities of the infections spreading, the size remains small and dies out in fewer time steps. As the probability grows, more people begin to get infected and more time steps are required before it dies out. This is because with each time step some people might be infected and they could infect others, but the probability isn't high enough to spread to a bunch of nodes at every timestamp. After the critical point, the size grows quickly causing a majority of the network to be infected. For these higher probabilities, the time steps begin to decrease since so many people are being infected at each time step that the network gets covered much faster. Thus, the time decreases to the log of the network size.

The critical point occurs around 0.13. There is an average of 8 edges per node. At this point the probability of not infecting any of the neighbors is 0.3. Thus, even if the nodes have fewer edges it's still likely that the node affects its neighbors. Given this probability, for each 3 infected nodes, 2 of them will infect at least one neighbor. Thus, the infection is likely to spread at the beginning and once it begins spreading, it will continue even through nodes with fewer neighbors.

(c) (30 pts) Now use the planted partition model of part (1a) to investigate whether the strength of community structure enhances, limits, or has no effect on epidemic size s/n and/or epidemic length ℓ . Let $n = 200$ and $c = 8$, and consider various combinations of the two parameters: $p \in [0, 1]$ and $\epsilon \in [0, 2c]$.

- Present your characterization of how community structure strength ϵ impacts epidemic size s/n and/or epidemic length ℓ using one or more figures that show the relationship clearly (smooth functions).
- Discuss the qualitative shape of these functions, and how they contrast, if at all, with your results from part (1b).
- Provide a brief intuitive explanation for why community structure strength does or does not impact the shape of the epidemic.

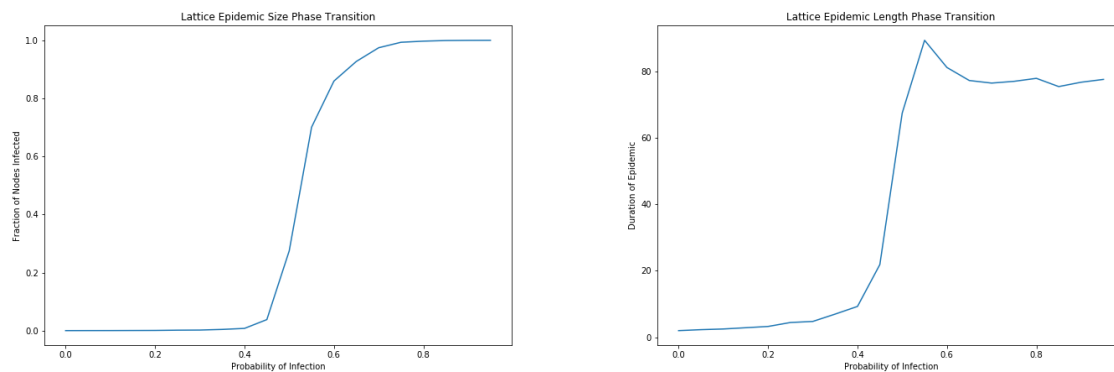


For most the values of ϵ , the structure of the epidemic size is the same as that of from part b. Thus, for any value of $\epsilon \in [0, 2 * c)$. The community structure doesn't impact the epidemic size. However, once ϵ is equal to or extremely close to $2 * c$, the graph is two disconnected subgraphs or the two groups have very few connections. With 2 disconnected groups, only the group that contains the first infected node can get infected regardless of the probability of infection. This means that at most half the nodes are infected. If the groups have only a few connections then there are limited opportunities for the infection to spread to the other group. However with more connections this becomes more probable. There is a steep drop off for the epidemic size near $\epsilon = 2 * c$.

Similarly, the length of the epidemic is very similar to the graph from part b for all values of ϵ except those near $2 * c$. Once again the lack of connections between the groups results in half the nodes not getting infected. Thus, it takes fewer iterations for the infection to die out. There is a steep drop off in the infection length for $\epsilon = 2 * c$. Thus, unless there is an extremely strong community structure, it has little effect on the epidemic length and size. However, even when $\epsilon = 2 * c$, both graphs have the same shape as those from part b, just with smaller values.

2. (30 pts total) The impact of long-range edges on spreading processes.

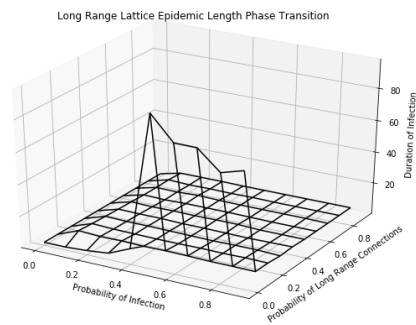
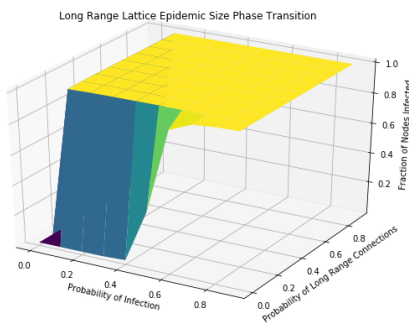
- (a) An $n \times n$ grid or lattice is a very simple network, in which each node, except for those on the boundary, connects to its neighbors above, below, left, and right of itself. Using the same SI model from question (1), investigate how the epidemic size s and epidemic length ℓ vary as a function of p , for $n = 50$. Present two figures showing these relationships and provide a brief interpretation of their shape, based on the way the SI model works and the shape of the network.



The phase transition for the size of the epidemic has a critical point around 0.45. With the lattice structure, each node has 4 neighbors on average. Thus, at this point, the probability of have none of the neighbors infected is about 0.09. Thus, after this point, the probability of having at least one neighbor infected is high. This results in the infection spreading across the network.

The length of infection similarly peaks shortly after the critical point. Before the critical point there are very few iterations since the infection is not spreading. However, one the critical point is reached in infection time quickly rises to the maximum time taken. At this point, the infection is likely to spread throughout most of the network. However, the probability is low enough that not all the neighbors are being infected at once resulting in a longer duration of infection. After this point the number of iterations decreases to the log of the number of nodes.

- (b) If a network can be embedded in a metric space, as can a grid or lattice, a “long-range” connection is one that connects two vertices that are separated by many steps on the lattice, i.e., two nodes that are “far” apart. Let q represent the probability that some pair of nodes i, j are connected by a long range link. Investigate how the epidemic size s and epidemic length ℓ depend on the variables q and p . Present your results clearly, and include a brief interpretation of how the long-range links change the epidemic dynamics relative to your results in part (2a).



When the probability of having long-range links is close to zero, then the graphs have the same structure as that of part a. However, when the probability of long-range links increases, the size of the infection will only be a fraction of the nodes for low probabilities of spreading the infection. Otherwise, once the probability of having the infection is greater than 0.05, the entire network gets infected. This is because the long-range links connect nodes that are further away. This causes the probability of the infection spreading to nodes that are further away to increase. This leads to different parts of the lattice being infected and then spreading it to their surrounding neighbors.

Due to this, the length of the infection is also the same as the previous for the probability of long-range links is close to zero. As the probability of long-range links increases, the infection begins to spread faster. This is because the distance of the network has decreased due to the long-range links. It becomes easier to reach any given node in the network from the initial infected node. For a lower probability of infection, the epidemic still dies out quickly. However, as the probability of getting infected increases, the epidemic length increases slightly but remains relatively small.

Algorithm 1: Python Code for Problems

```

import numpy as np
import matplotlib.pyplot as plt
import os
import networkx as nx
from scipy.stats import gaussian_kde

#created planted partition graph
def createPlantedPartitionGraph(n,c,eps):
    G = nx.Graph()
    p_in = (2*c+eps)/(2*n)
    p_out = (2*c-eps)/(2*n)
    k = n//2
    for i in range(k):
        G.add_node(i)
        G.add_node(i+k)
        for j in range(k):
            if j>i:
                #connection between group 1
                if random.random()<p_in:
                    G.add_edge(i,j)
                #connection between group 2
                if random.random()<p_in:
                    G.add_edge(i+k,j+k)
            #connections between the 2 groups
            if random.random()<p_out:
                G.add_edge(i,j+k)
    return G

#PROBLEM 1a - visualizations
n = 50
c = 5
eps = [0,4,8]
for e in eps:
    G = createPlantedPartitionGraph(n, c, e)
    pos = nx.spring_layout(G)
    plt.figure()
    nx.draw_spring(G)
    plt.draw()
    plt.axis('off')
    plt.savefig("eps"+str(e)+".png")

#spreading the infection across the graph
def spreadingProcess(G, p):
    #randomly infect one vertex
    infected = [random.choice(list(G.nodes))]
    index = 0
    t = 1

```

```
#while new people are being infected
while index < len(Infected):
    #loop over the new infected people
    num_inf = len(Infected)
    while index < num_inf:
        #infect their uninfected neighbors with prob p
        for node in G.neighbors(Infected[index]):
            if not node in Infected and random.random() <= p:
                Infected.append(node)
        index += 1
    t += 1
return (len(Infected)/len(G), t)

#PROBLEM 1b - spreading visualization
n = 1000
c = 8
eps = 0
reps = 500
epid_size = []
epid_len = []
probs = np.arange(0,1,0.001)
#run repetitions and get data
for p in probs:
    size = 0
    length = 0
    for i in range(reps):
        G = createPlantedPartitionGraph(n,c,eps)
        (s,l) = spreadingProcess(G, p)
        size += s
        length += l
    epid_size.append(size/reps)
    epid_len.append(length/reps)

#plot size
plt.figure(figsize=(10,7))
plt.plot(probs, epid_size)
plt.axvline(0.13)
plt.title("Epidemic_Size_Phase_Transition")
plt.xlabel("Probability_of_Infection")
plt.ylabel("Fraction_of_Nodes_Infected")
plt.savefig("size.png")

#plot length
plt.figure(figsize=(10,7))
plt.plot(probs, epid_len)
plt.axhline(math.log(n))
plt.axvline(0.13)
plt.title("Epidemic_Length_Phase_Transition")
plt.xlabel("Probability_of_Infection")
plt.ylabel("Duration_of_Epidemic")
plt.savefig("length.png")
```



```
#PROBLEM 1c - spreading visualization with community structure
n = 1000
c = 8
reps = 500
epid_size_2d = []
epid_len_2d = []
probs = np.arange(0,1,0.05)
eps = np.append(np.arange(0,2*c,1), np.arange(2*c-0.9,2*c+0.01,0.1))
#run repetitions and get data
for e in eps:
    epid_size = []
    epid_len = []
    for p in probs:
        size = 0
        length = 0
        for i in range(reps):
            G = createPlantedPartitionGraph(n,c,e)
            (s,l) = spreadingProcess(G, p)
            size += s
            length += l
        epid_size.append(size/reps)
        epid_len.append(length/reps)
    epid_size_2d.append(epid_size)
    epid_len_2d.append(epid_len)
#plot epidemic size
X, Y = np.meshgrid(probs,eps)
fig = plt.figure(figsize=(10,7))
ax = plt.axes(projection='3d')
ax.plot_wireframe(X, Y, np.asarray(epid_size_final), color='black')
ax.set_title("Epidemic_Size_Phase_Transition")
ax.set_xlabel("Probability_of_Infection")
ax.set_ylabel("Community_Structure_Strength")
ax.set_zlabel("Fraction_of_Nodes_Infected")
plt.savefig("size_2d.png")

#plot epidemic length
X, Y = np.meshgrid(probs,eps)
fig = plt.figure(figsize=(10,7))
ax = plt.axes(projection='3d')
ax.plot_wireframe(X, Y, np.asarray(epid_len_final), color='black')
ax.set_title("Epidemic_Length_Phase_Transition")
ax.set_xlabel("Probability_of_Infection")
ax.set_ylabel("Community_Structure_Strength")
ax.set_zlabel("Duration_of_Infection")
plt.savefig("length_2d.png")

#create lattice graph
def latticeGraph(n):
    G = nx.Graph()
```

```

    for r in range(n):
        for c in range(n):
            if r>0:
                G.add_edge(n*(r-1)+c, n*r+c)
            if r<n-1:
                G.add_edge(n*(r+1)+c, n*r+c)
            if c>0:
                G.add_edge(n*r+c-1, n*r+c)
            if c<n-1:
                G.add_edge(n*r+c+1, n*r+c)
    return G

#PROBLEM 2a - spreading visualization lattice
n = 50
G = latticeGraph(n)
reps = 100
epid_size = []
epid_len = []
probs = np.arange(0,1,0.005)
#run repetitions and get data
for p in probs:
    size = 0
    length = 0
    for i in range(reps):
        (s,l) = spreadingProcess(G, p)
        size += s
        length += l
    epid_size.append(size/reps)
    epid_len.append(length/reps)

#plot size
plt.figure(figsize=(10,7))
plt.plot(probs, epid_size)
plt.title("Lattice_Epidemic_Size_Phase_Transition")
plt.xlabel("Probability_of_Infection")
plt.ylabel("Fraction_of_Nodes_Infected")
plt.savefig("lattice_size.png")

#plot length
plt.figure(figsize=(10,7))
plt.plot(probs, epid_len)
plt.title("Lattice_Epidemic_Length_Phase_Transition")
plt.xlabel("Probability_of_Infection")
plt.ylabel("Duration_of_Epidemic")
plt.savefig("lattice_length.png")

#create lattice graph with long range links
def longRangeLatticeGraph(n,q):
    G = nx.Graph()
    for r in range(n):
        for c in range(n):

```

```

        if r>0:
            G.add_edge(n*(r-1)+c, n*r+c)
        if r<n-1:
            G.add_edge(n*(r+1)+c, n*r+c)
        if c>0:
            G.add_edge(n*r+c-1, n*r+c)
        if c<n-1:
            G.add_edge(n*r+c+1, n*r+c)
        #long range edges
        for r2 in range(r+1,n):
            for c2 in range(c+1,n):
                if r2-r + c2-c >5 and random.random()<q:
                    G.add_edge(n*r+c, n*r2+c2)

    return G

#PROBLEM 2b - spreading visualization with long range lattice
n = 50
reps = 100
epid_size_2d = []
epid_len_2d = []
q = np.append(np.arange(0,0.1,0.01),np.arange(0.1,1,0.1))
probs = np.append(np.arange(0,0.1,0.01),np.arange(0.1,1,0.1))
#run iterations and get data
for q_val in q:
    epid_size = []
    epid_len = []
    for p in probs:
        size = 0
        length = 0
        for i in range(reps):
            G = longRangeLatticeGraph(n, q_val)
            (s,l) = spreadingProcess(G, p)
            size += s
            length += l
        epid_size.append(size/reps)
        epid_len.append(length/reps)
    epid_size_2d.append(epid_size)
    epid_len_2d.append(epid_len)

#plot epidemic size
X, Y = np.meshgrid(probs,q)
fig = plt.figure(figsize=(10,7))
ax = plt.axes(projection='3d')
ax.plot_wireframe(X, Y, np.asarray(epid_size_2d), color='black')
ax.set_title("Long_Range_Lattice_Epidemic_Size_Phase_Transition")
ax.set_xlabel("Probability_of_Infection")
ax.set_ylabel("Probability_of_Long_Range_Connections")
ax.set_zlabel("Fraction_of_Nodes_Infected")
plt.savefig("lattice_size_2d.png")

#plot epidemic length

```

```
X, Y = np.meshgrid(probs,q)
fig = plt.figure(figsize=(10,7))
ax = plt.axes(projection='3d')
ax.plot_wireframe(X, Y, np.asarray(epid.len_2d), color='black')
ax.set_title("Long_Range_Lattice_Epidemic_Length_Phase_Transition")
ax.set_xlabel("Probability_of_Infection")
ax.set_ylabel("Probability_of_Long_Range_Connections")
ax.set_zlabel("Duration_of_Infection")
plt.savefig("lattice_length_2d.png")
```