Final Project: Part 2

Karros Huang, Girish Narayanswamy, and Clint Olsen

ECEN 5322 Higher-Dimensional Datasets

University of Colorado
Boulder

Introduction

This is an exploration of the spread of and vaccination for a disease in a network. The premise is to understand how disease spread from an initial infected node, and which factors correlate to the scale of an epidemic. Additionally this paper explores the value of vaccination in a network, where certain nodes are immune and cannot transmit a disease.

In the previous part of the project, we analyzed multiple datasets that described the face-to-face and co-presence interaction amongst communities. These datasets were transformed into adjacency matrices, and their structural properties were evaluated. Furthermore, we down-sampled the graphs using various sampling algorithms, compared the structural properties of the down sampled co-presence network to the face-to-face network, and came to the conclusion that it is a feasible idea to represent the data-rich and fine-grained representation of connections between individuals, with a sparser and more coarsely sampled structures.

The exploration of the spread of a disease is conducted on both face-to-face data as well as co-presence data (discussed in the first part of this project). The goal is to show that the evolution of the disease in a face-to-face network, can also be represented by the co-presence network.

All parts of this project use the Susceptible-Invective-Recovered (SIR) graph model. The model categorizes each node as either susceptible to infection, already infected, or recovered. Susceptible nodes are not infected, but are also not immune to infection if a neighbor is infected. Infected nodes may infect neighboring nodes, and require some amount of time to recover. Recovered nodes, are previously infected nodes which are now immune, meaning that cannot be infected and cannot infect neighboring nodes.
For each epidemic (SIR) simulation a μ was chosen, where μ is the recovery rate, how quickly a node recovers from infection. During the time that a node is infected, it is able to infect its neighbors with a probability proportional to the edge weight between those nodes. Thus nodes with a higher number of "interactions" (degree) are more likely to spread infection. μ is defined as the following:

$\mu = \beta/k \times 100$
$k \in \{2, 4, 6, 8, 10\}$
$\beta = 4 \times 10^{-4}$

Before diving any further, a few notes on μ and k:
The initial value of μ was multiplied by 100. The previous value of μ was too small, and thus did not allow nodes to recover in a reasonable time frame. The increase of this rate in reality correlates to the idea that different groups of people, dependent on demographic, age and other parameters, have different recovery rates associated with them. With the given weights and β value, the probability of infection was far greater than the probability of recovery, thus showing the need for scaling μ. The initial values of $k \in \{1, 4, 16, 64, 256\}$ were altered to $k \in \{2, 4, 6, 8, 10\}$. The original values of k were so large that for value of $k > 4$ all of the nodes became infected due to the exponential growth of the reproduction rate. These values were adjusted as a epidemic where 20% of the nodes are infected is considered quite large. Thus using values of large values of k such as $k > 10$ resulted in all the nodes being infected which is a rather unhelpful case.

Part 7, 8, 9

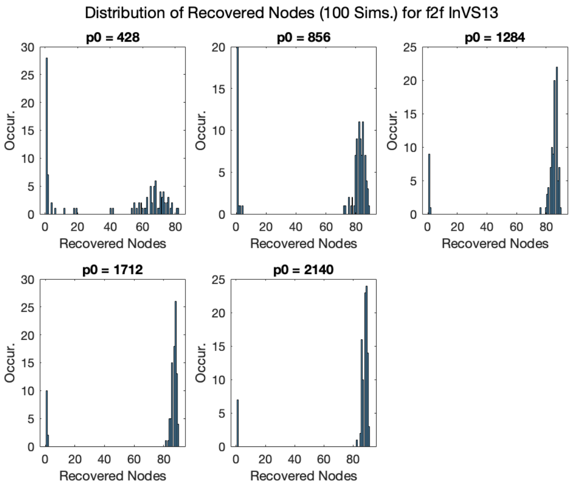
Parts 7, 8, and 9 of this project are based around the spread of infection in a graph. The infection begins at a randomly infected node, and continues until all nodes are either recovered, or there are no longer infected nodes in the graph.

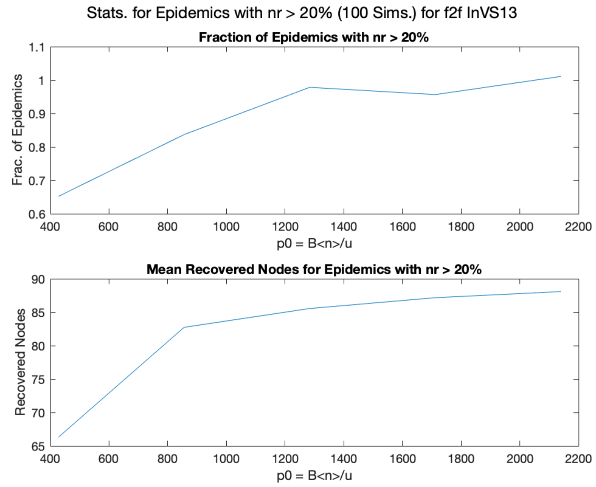
For parts 7, 8, 9 the SIR simulation was run 100 times per value of μ, for each dataset's face-to-face adjacency matrix. For each simulation the number of recovered nodes n_r was recorded. The distribution of n_r, the fractions of epidemics with the fraction of n_r over 20%, and the average number of n_r in epidemics with over 20% recovered nodes, are plotted below. All these plots are shown as a function of ρ_0 which is defined as the following:

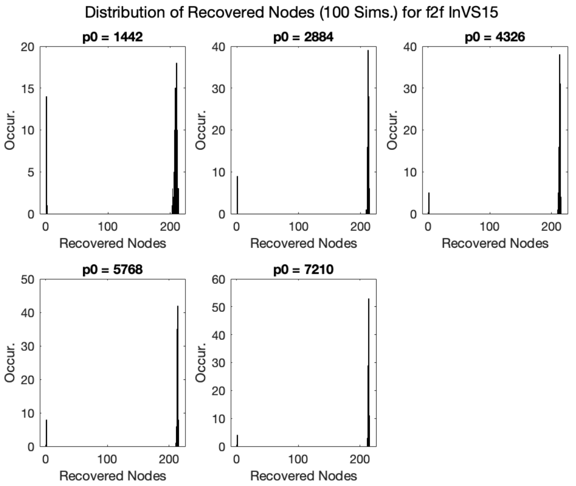
$\rho_0 = \beta < d > /\mu$

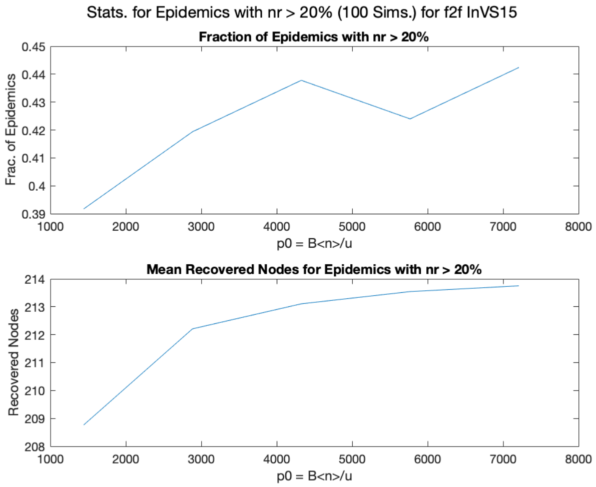
Where $< d >$ is the average degree of the graph.

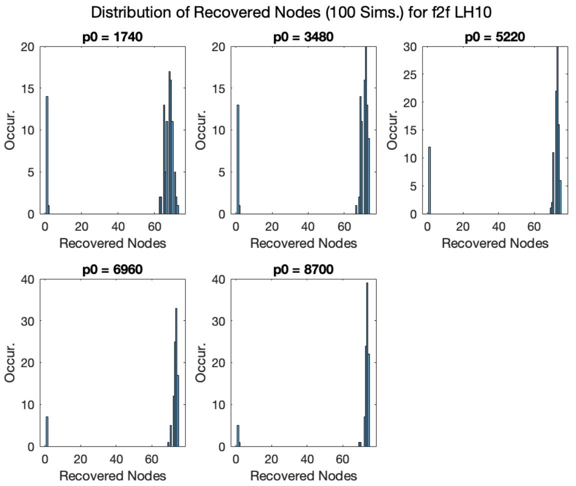
The following are figures for the 6 datasets:

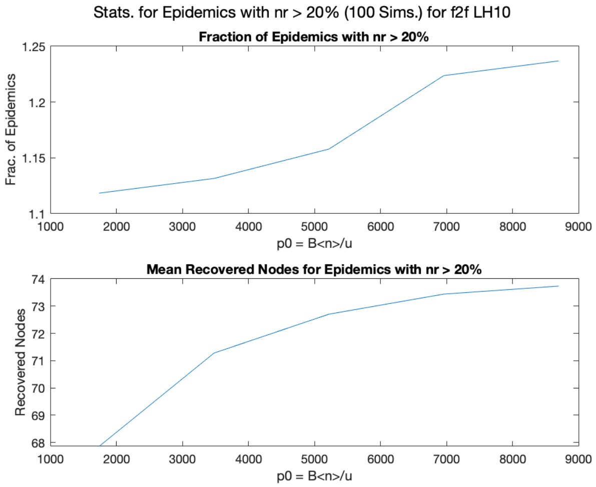

Distribution of Recovered Nodes (100 Sims.) for f2f InVS13
p0 = 428
p0 = 856
p0 = 1284
p0 = 1712
p0 = 2140
Occur.
Recovered Nodes
0
20
40
60
80
0
5
10
15
20
25
30

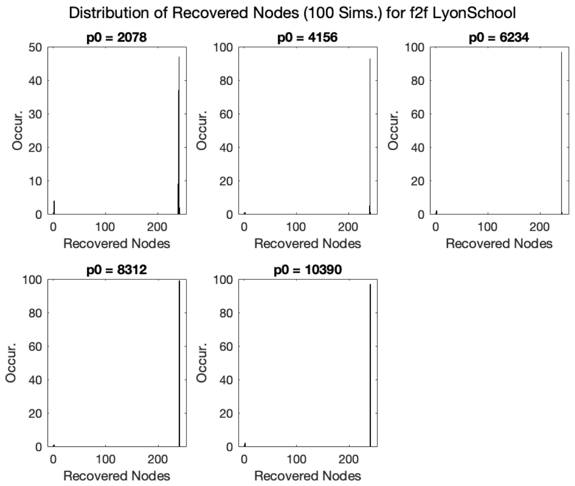

Stats. for Epidemics with nr > 20% (100 Sims.) for f2f InVS13
Fraction of Epidemics with nr > 20%
Frac. of Epidemics
0.6
0.7
0.8
0.9
1
1.1
Mean Recovered Nodes for Epidemics with nr > 20%
Recovered Nodes
65
70
75
80
85
90
400
600
800
1000
1200
1400
1600
1800
2000
2200
p0 = B<n>/u

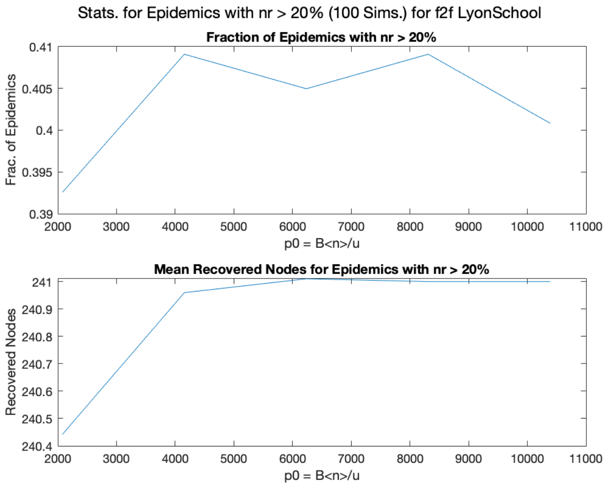

Distribution of Recovered Nodes (100 Sims.) for f2f InVS15
p0 = 1442
p0 = 2884
p0 = 4326
p0 = 5768
p0 = 7210
Occur.
Recovered Nodes
0
100
200

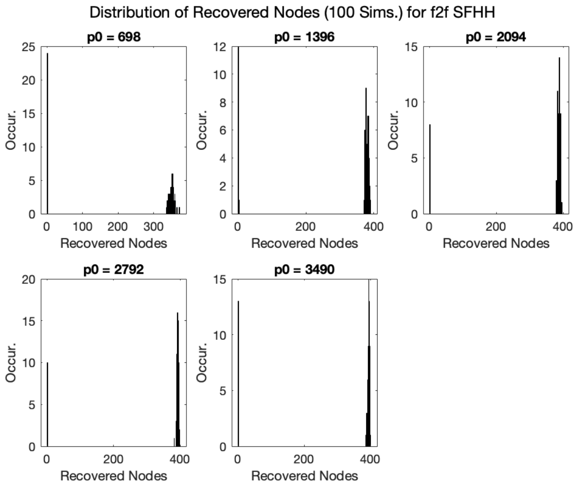

Stats. for Epidemics with nr > 20% (100 Sims.) for f2f InVS15
Fraction of Epidemics with nr > 20%
Frac. of Epidemics
0.39
0.4
0.41
0.42
0.43
0.44
0.45
Mean Recovered Nodes for Epidemics with nr > 20%
Recovered Nodes
208
209
210
211
212
213
214
1000
2000
3000
4000
5000
6000
7000
8000
p0 = B<n>/u

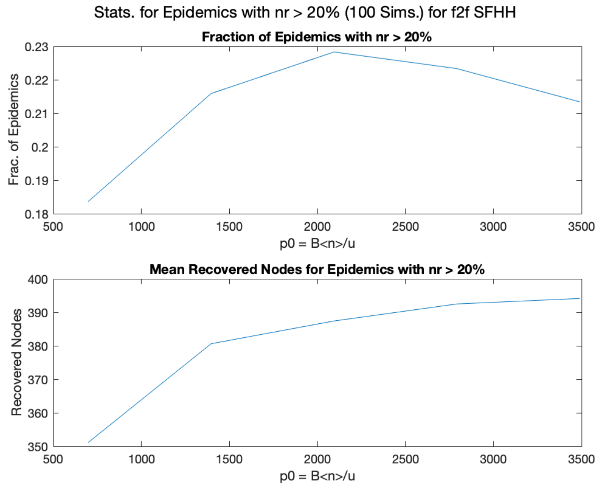

Distribution of Recovered Nodes (100 Sims.) for f2f LH10
p0 = 1740
p0 = 3480
p0 = 5220
p0 = 6960
p0 = 8700
Occur.
Recovered Nodes
0
20
40
60

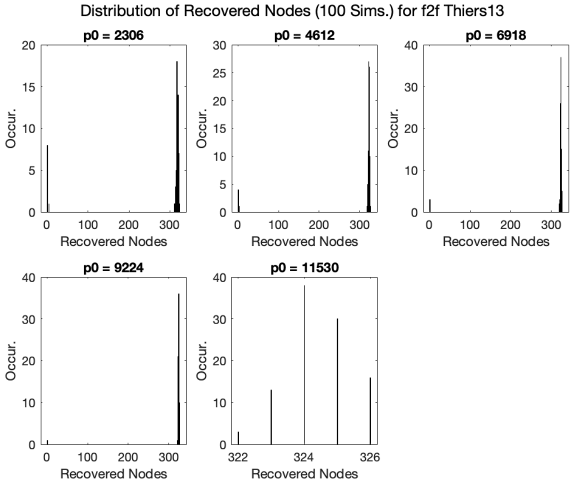

Stats. for Epidemics with nr > 20% (100 Sims.) for f2f LH10
Fraction of Epidemics with nr > 20%
Frac. of Epidemics
1.1
1.15
1.2
1.25
Mean Recovered Nodes for Epidemics with nr > 20%
Recovered Nodes
68
69
70
71
72
73
74
1000
2000
3000
4000
5000
6000
7000
8000
9000
p0 = B<n>/u

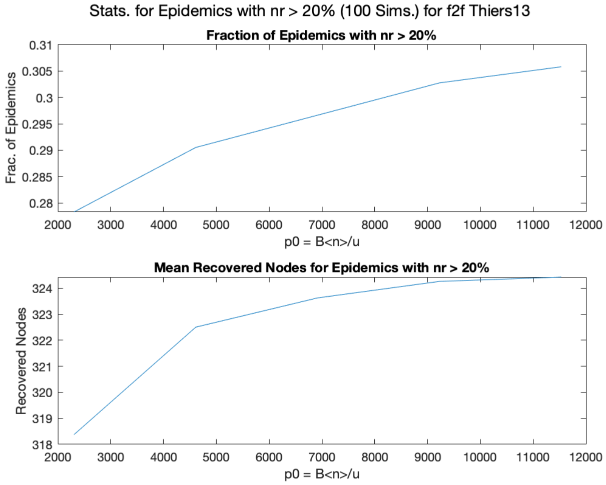












The histograms above **(part 7)** represent the distribution of number of recovered nodes over 100 simulations for the given value of ρ_0. It is clear that as the value of ρ_0 increases the distribution of recovered (once infected) nodes becomes tighter, with a higher mean and a lower standard deviation. This is because higher values of ρ_0 correspond to lower values of the recovery rate μ. With a lower recovery rate a node has more time to infect its neighbors. Thus for larger values of ρ_0 the vast majority of nodes in the graph become infected.

This phenomenon is most obvious for relatively sparse f2f graphs, like that of InVS13. As ρ_0 increases it is visually apparent that the distribution of number of recovered nodes becomes tighter with a higher mean. It is less obvious for extremely dense f2f graphs, such as that of the LyonSchool dataset. In this dataset, the graph is dense enough that even with relatively high values of μ the vast majority of nodes become infected because the average degree of a node is very high.

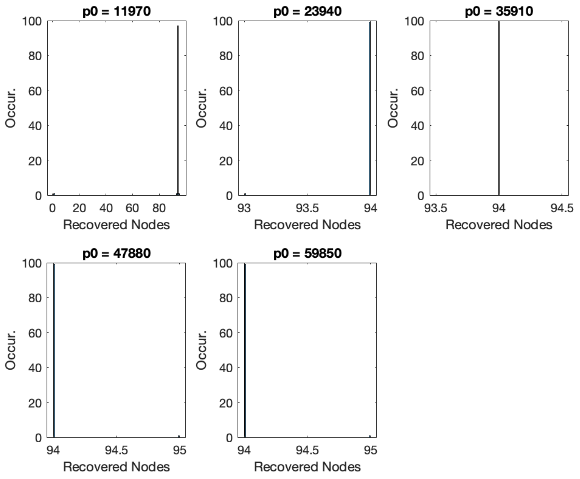
For each dataset the first of the two plots **(part 8)** shows the fraction of epidemics, out of 100 simulations, where the fraction of recovered nodes is greater than 20%. These situations are considered large epidemics. The plots show that as ρ_0 increases the average number of "large" epidemics increases. After a certain value of ρ_0 the trend begins to plateau. This suggests there is a critical value of ρ_0 after which point the a significant portion of nodes become infected.

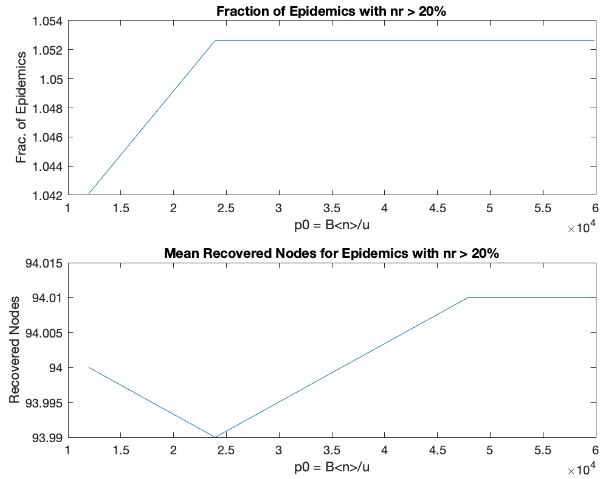
A similar trend can be observed for the second of the two plots **(part 9)** for each dataset. These plots show the trend of mean number of recovered nodes over the 100 simulations. As the value of ρ_0 increases the mean number of infected nodes increases as well. The values begins to plateau after a sufficiently high value of ρ_0. This suggests as ρ_0 increases the mean number of infected/recovered nodes increases until almost all of the nodes are at some point infected.

The histogram and the two trend plots per dataset provide some insight based off how infections spread based off the structure of the graph. As ρ_0 increases the distribution of infected nodes increases, the fraction of simulations with recovered nodes greater than 20% increases, and the mean number of recovered nodes increases. This suggests that as the value of μ, the recovery rate, decreases more and more nodes are infected. Additionally as the average degree, $< d >$, increases the more likely nodes are to become infected in a graph. Thus networks with high average degree, and with low recovery rate are more susceptible to large epidemics.

Part 10

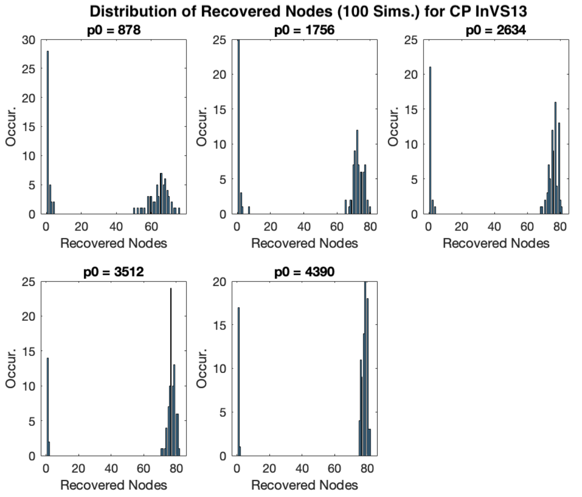
Part 10 compares the distribution of recovered nodes and the stats for the subsampled co-presence networks to what we discovered in the face-to-face networks in Part 7, 8, 9. For this part and part 14, the Metropolis Hastings algorithm was used due to its consistency in previous experimentation. We found that with $f = 0.8$ the infection was taking over the recovery and infecting the entire network as a result of the average degree density of the co-presence network being significantly greater than the face-to-face. The few plots below show that our value of $p0$ is drastically greater than of those in the face-to-face, and as a result the distribution of recovered nodes over 100 simulations all fell onto the max number of nodes of the graph. Therefore setting $f = 0.8$ is insufficient and the density needs to be reduced.

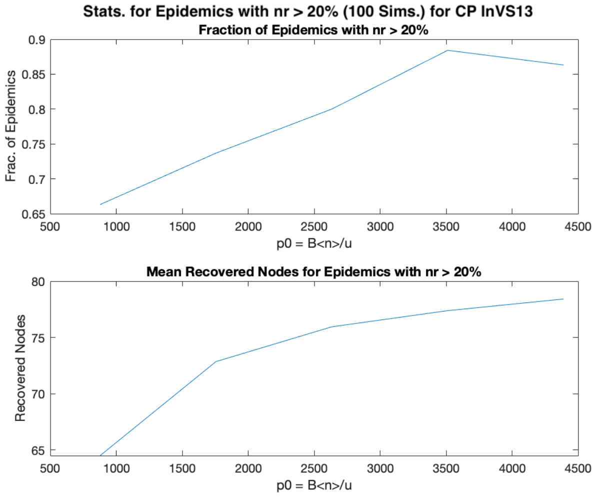


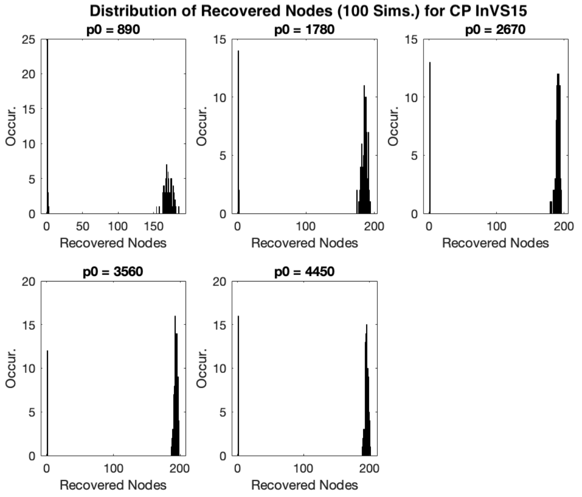


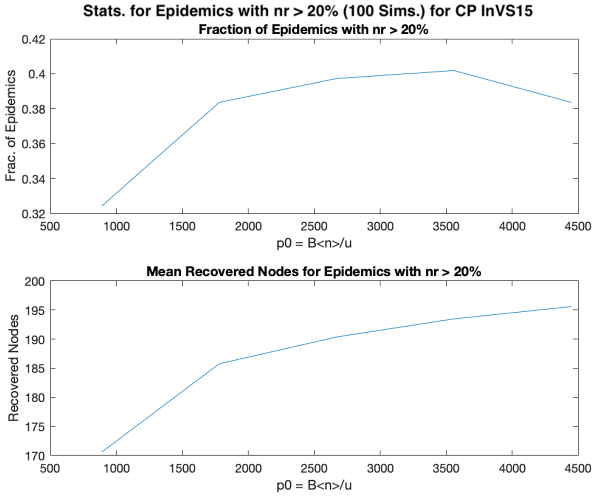
From the first part of the project, we learned that down-sampling the frequency of the average degrees retained the structure of the co-presence network because the clustering coefficient and degree distribution were the same. So in order to resolve the over-saturation of degrees, we set f so that the average degree density of the co-presence graph is normalized to the face-to-face graph $f = \frac{f2f\_avg\_degree}{cp\_avg\_degree}$

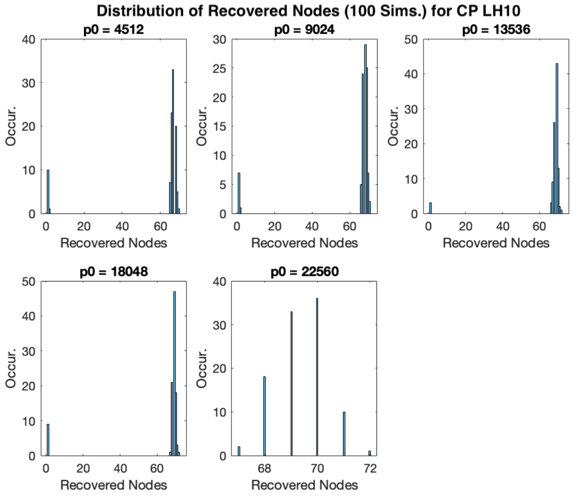
The following are figures for the 6 datasets:

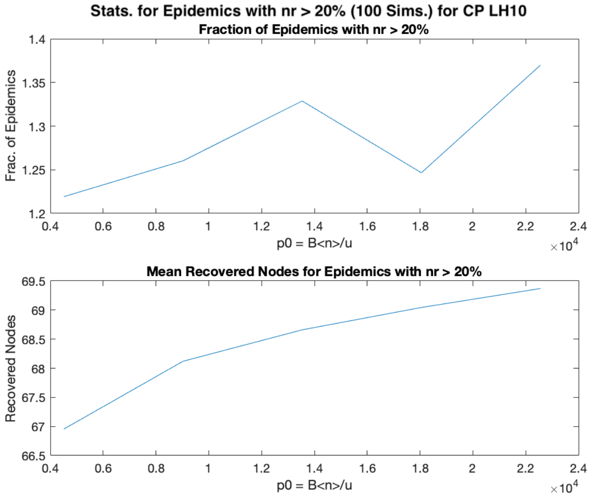

Distribution of Recovered Nodes (100 Sims.) for CP InVS13
p0 = 878
p0 = 1756
p0 = 2634
p0 = 3512
p0 = 4390
Occur.
Recovered Nodes
0
20
40
60
80

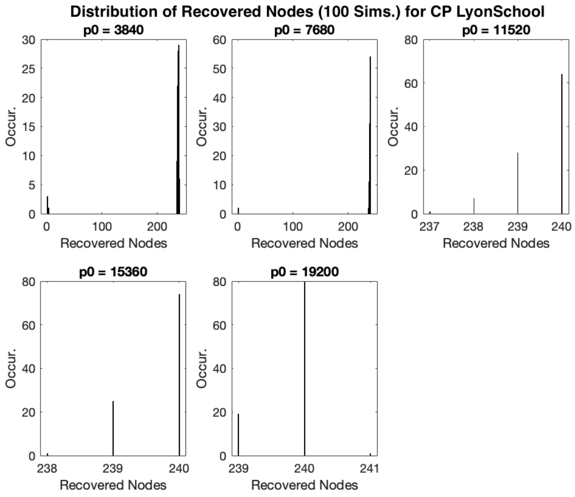

Stats. for Epidemics with nr > 20% (100 Sims.) for CP InVS13
Fraction of Epidemics with nr > 20%
Frac. of Epidemics
0.65
0.7
0.75
0.8
0.85
0.9
Mean Recovered Nodes for Epidemics with nr > 20%
Recovered Nodes
65
70
75
80
500
1000
1500
2000
2500
3000
3500
4000
4500
p0 = B<n>/u

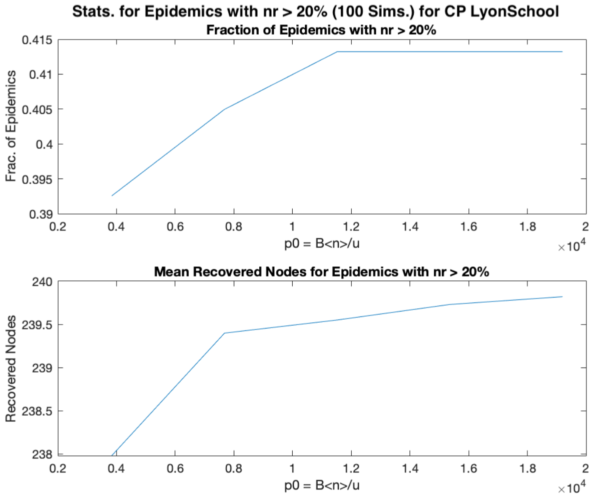

Distribution of Recovered Nodes (100 Sims.) for CP InVS15
p0 = 890
p0 = 1780
p0 = 2670
p0 = 3560
p0 = 4450
Occur.
Recovered Nodes
0
50
100
150
200

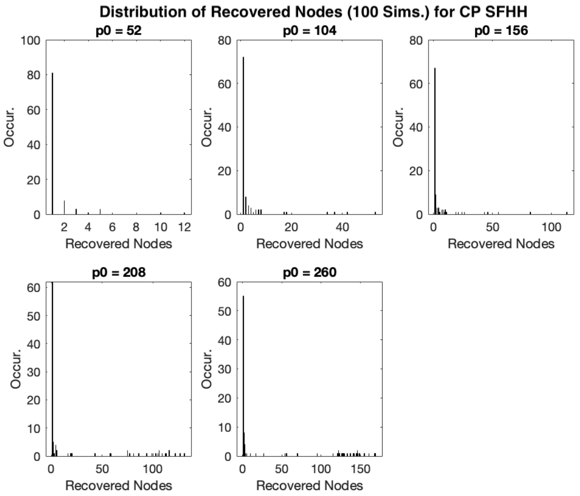

Stats. for Epidemics with nr > 20% (100 Sims.) for CP InVS15
Fraction of Epidemics with nr > 20%
Frac. of Epidemics
0.32
0.34
0.36
0.38
0.4
0.42
Mean Recovered Nodes for Epidemics with nr > 20%
Recovered Nodes
170
175
180
185
190
195
200
500
1000
1500
2000
2500
3000
3500
4000
4500
p0 = B<n>/u

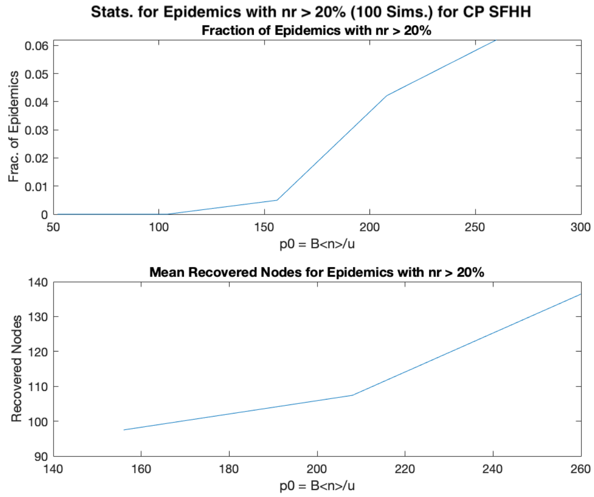

Distribution of Recovered Nodes (100 Sims.) for CP LH10
p0 = 4512
p0 = 9024
p0 = 13536
p0 = 18048
p0 = 22560
Occur.
Recovered Nodes
0
20
40
60
68
70
72

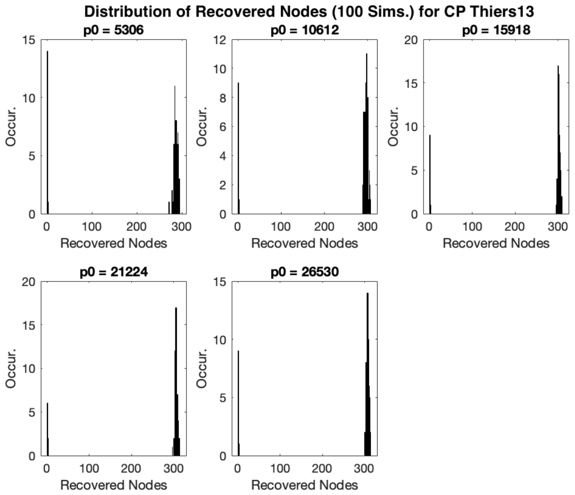

Stats. for Epidemics with nr > 20% (100 Sims.) for CP LH10
Fraction of Epidemics with nr > 20%
Frac. of Epidemics
1.2
1.25
1.3
1.35
1.4
Mean Recovered Nodes for Epidemics with nr > 20%
Recovered Nodes
66.5
67
67.5
68
68.5
69
69.5
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
2.2
2.4
p0 = B<n>/u
×10^4

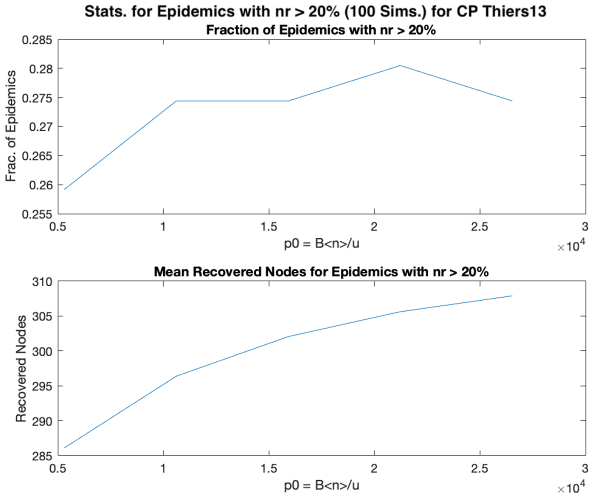








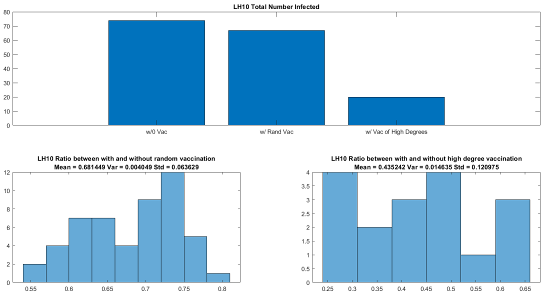


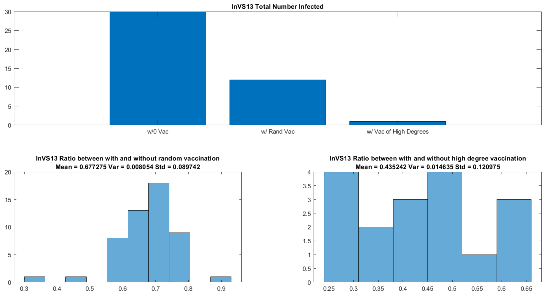


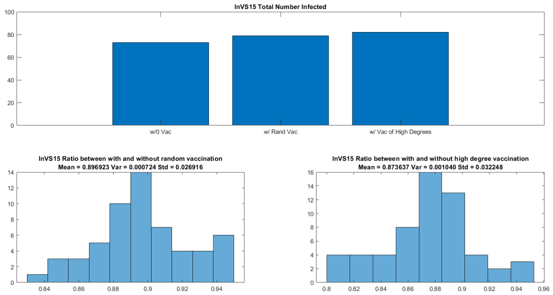
For most of the networks this solution appears to work. InVS13, InVS15, LH10, LyonSchool, and Thiers13 all seem to have comparable results between their face-to-face and co-presence graphs. The only exception is SFHH, where the normalization technique broke the graph and only very few simulations were able to recover from the disease. Otherwise all other simulations had 0 recovered nodes.

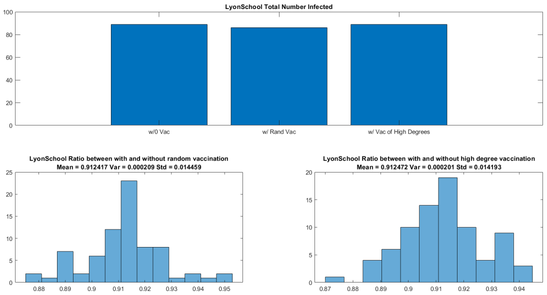
Part 11, 12, 13

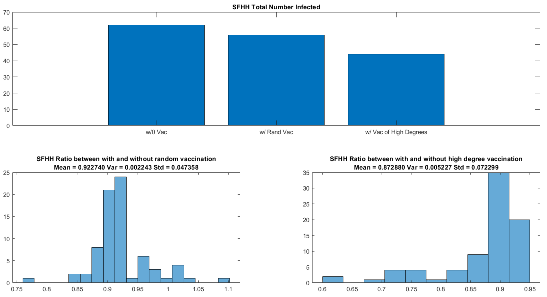
As a further extension of the experiment and the correlation between a network in reality, the idea of vaccination was implemented within each of the 6 datasets with two different methods of vaccination. The first method consists of uniformly choosing 20 random nodes to vaccinate in the network while the other is choosing the 20 nodes with the highest degree. 20 vaccinated nodes were used in all experiments. Below shows plots that correlate to the results of running the SIR model with the inclusion of these vaccinated nodes on the face to face networks:

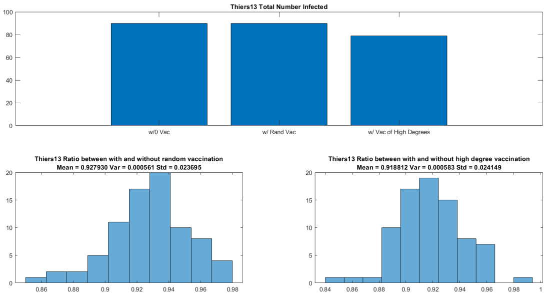












Several things are shown within these plots. The first (top bar graph in each set of 3 plots) is the total number of simulations that resulted in a infection of over 20% of the nodes being infected in three cases: without vaccination, with random vaccination, and with high degree vaccination (the plot specifies that it is number of nodes infected, but it is in fact the number of outbreaks of 20% or greater). The other two histograms in these groupings show the ratio distributions over the 100 simulations over the number of nodes infected with vaccination vs. without for both vaccination schemes in cases where the outbreak was greater than 20% of the population. The first interesting thing to note here is that the success of the vaccination was dependent on the structure of the graph, as well as the size of the graph.In most cases, the total number of epidemics was in fact reduced, but the difference between the scenarios with and without vaccination were minimal. For example, with the LH10 and invs13 data sets, the impact of vaccination was substantial where the mean of the ratio distributions shows anywhere from a 30% to 50% decrease in number of infected nodes when vaccination was implemented. However, in looking at the LyonSchool dataset, which is much more dense due to the structure of the graph and social environment. The effects of vaccination were less prevalent. This also leads to the idea that selecting a constant number of nodes for all datasets is perhaps not the best scheme, as varying the vaccinated nodes based on the total number of nodes in the graph is a better scheme. It seems logical that especially in the case for the school, the effects of vaccination would be very substantial, so the number of vaccinated nodes was re-scaled based on the number of nodes in the network to see if the expected trend occurs. Below shows the resulting comparison:
The impact here is substantial and gives a result that is more expected from the scenario and graph structure from the school environment, so in future experiments it would likely be a much better scheme to vaccinate this way in practice.

Another possible explanation for seeing little effect in difference between scenarios with and without vaccination could be dependent on the structure of the graph. Looking back at the visualization of the Thiers adjacency matrix from part 1 of the experiment, it looks like the areas of contact are very compartmentalized, which would mean interleaving vaccinated nodes would only stop the spread of infection in small local areas and not across the entire graph. A final interesting note is

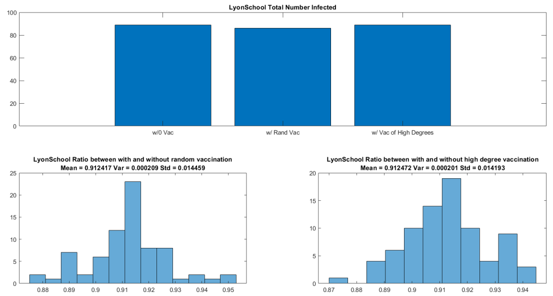


Figure 1: Lyon School with 20 vaccinated nodes

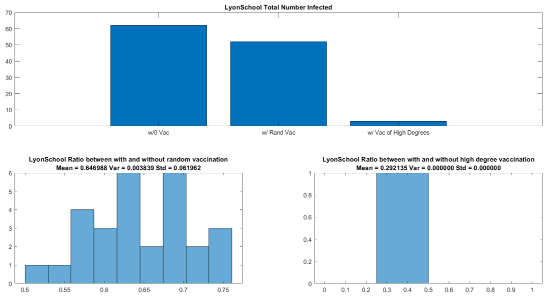
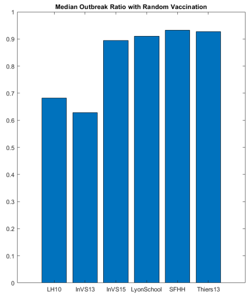
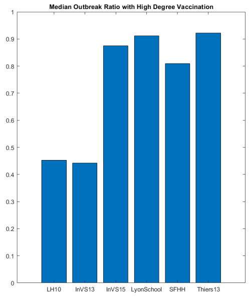


Figure 2: Lyon School with 25% of the nodes vaccinated

looking at the ratio distributions between the random vaccination distributions and the high degree vaccination distributions. In general the results show a greater variance when applying the high degree vaccination in terms of the percentage of nodes infected in comparison to the non-vaccinated scenario. One possible explanation for this is that high degree vaccination results in a fixed set of vaccinated nodes for each random infection spread, which in turn causes large variation in how the spread interacts with these nodes each simulation. In the random case, both the vaccinated nodes selection and the infection spread are random, so it is possible that over a large number of simulations the distribution converges to a similar value.

The other plot that was generated here shows the ratio of the median value of infected nodes for both vaccination methods, which gives insight into how these methods can be compared.



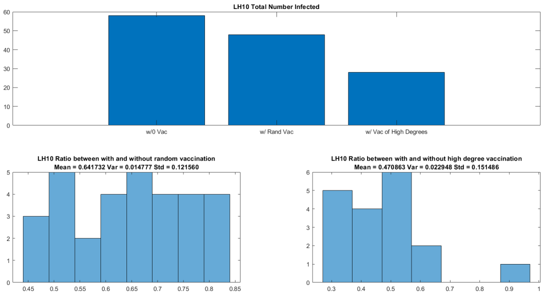


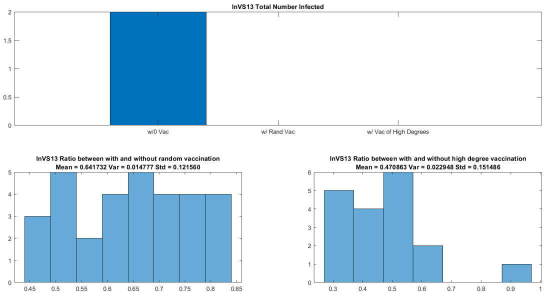
From the median plot it can be seen that on average, the method of vaccinating the highest degree results in a lower median ration between non-vaccinated and vaccinated nodes, which is to be expected. In practice, this correlates to vaccinating the social epicenters within a social group (such as healthcare providers or teachers) that have a lot of contact with the rest of the graph over the course of time. In practice this seems like the most logical and simple solution to the problem. Practically speaking, this can also be correlated to high bandwidth network routers or switches that could be highly susceptible to transmitting virus across the internet. Another possible scheme for vaccination would be to analyze the community structure of the graph, and attempt to vaccinate the highest degree nodes in each community. This would attempt to prevent both spread of infection within each community, and also the transmission of infection from one community to another.

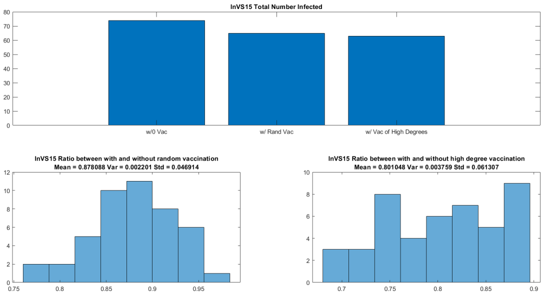
Part 14

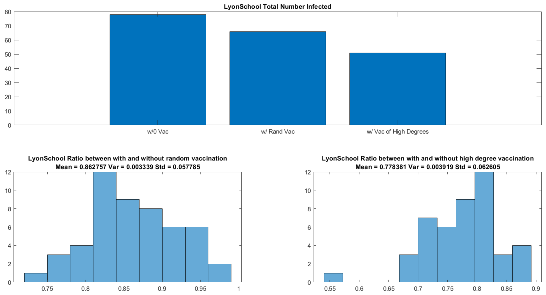
We now use the co-presence data and run the SIR model with vaccinated nodes. Our value of f for the down-sampling will continue following the formula $f = \frac{f2f\_avg\_degree}{cp\_avg\_degree}$ for we have shown that this gives us a better representation of the disease spreading through the network than using a constant value for all cases.

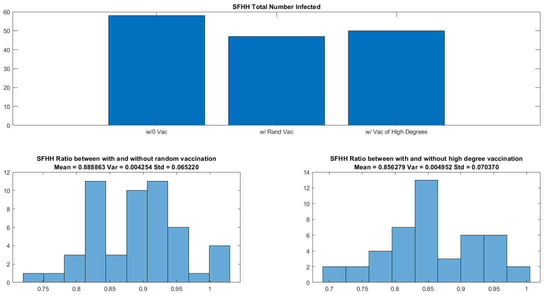
The following are the histograms and statistics showing the effect of vaccinated nodes in the network for the co-presence data.

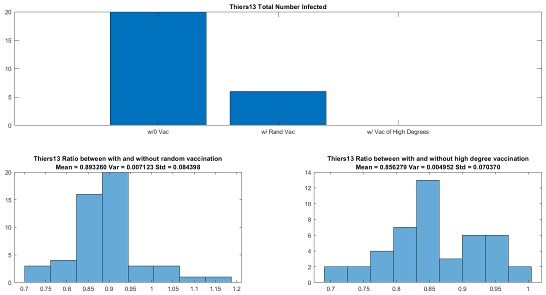


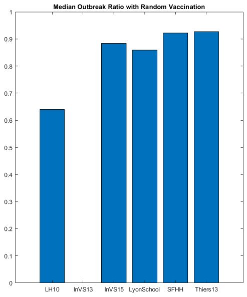


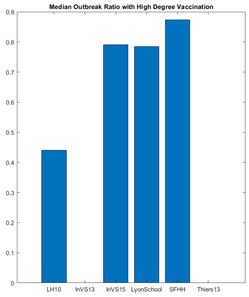




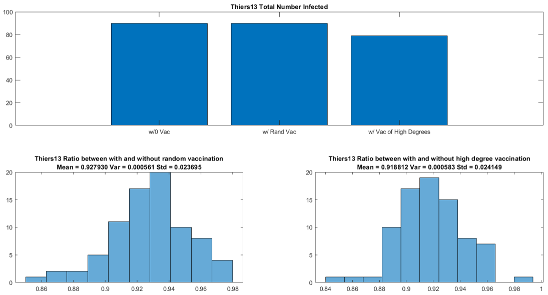






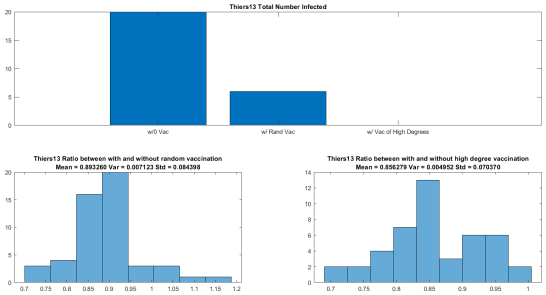


The trends among the co-presence plots differs from the face-to-face plots that were derived in parts 11, 12, and 13. The total number of infections through the course of 100 simulations that reach over 20% of the population is slightly less throughout every co-presence dataset compared to the face-to-face dataset. The mean and standard deviation of the histograms are relatively similar and differ by ±10% . We presume the reason why the number of infections among the co-presence graph is uniformly less is because the effects of the vaccinations are much more prevalent in a co-presence simulation. If the nodes with the highest degrees are vaccinated in a co-presence dataset, then the disease will fail to infect a population greater than 20%, for the highest degree nodes were the primary sources to propagate the disease. However, the results of the subampling for some of the datasets greatly alter the networks themselves (see InVS13), which distorts the spread of the epidemic seen in the face to face case. Overall, the method of subsampling and using the co-presence data for this scenario was not reliable enough to be deemed feasible. Below shows an example that exemplifies this difference in the Thiers dataset:


Thiers13 Total Number Infected

Thiers13 Ratio between with and without random vaccination
Mean = 0.927930 Var = 0.000561 Std = 0.023695

Thiers13 Ratio between with and without high degree vaccination
Mean = 0.918812 Var = 0.000583 Std = 0.024149


Thiers13 Total Number Infected

Thiers13 Ratio between with and without random vaccination
Mean = 0.893260 Var = 0.007123 Std = 0.084398

Thiers13 Ratio between with and without high degree vaccination
Mean = 0.856279 Var = 0.004952 Std = 0.070370

Conclusion

The process of analyzing and correlating the data in this experiment back to reality was a very interesting and insightful learning experience. It really showed that making simple assumptions regarding small details within this experiment can lead to large impacts in the results due to the complexity of the problem in general. The results showed that perhaps the co-presence data is too dense and coarse to accurately depict the spread of the infection through the network, especially due to the fact that the SIR model seems optimized to work with face to face data, purely in its nature. The only solution trialed in making the simulation accurately model the spread of the disease using the co-presence data is by scaling the average degree to what the face-to-face data had. However, its hard to say that this is a viable solution because the co-presence data alone did not create an accurate model as it is still dependent on the face-to-face data. Despite the difficulties with the subsampled co-presence, the access to face to face data, although perhaps not practical in general, allowed us to visualize and confirm suspected hypotheses regarding how vaccination would affect the graph and how infection would spread in different graph structures. Overall, these experiments showed not only how important defining characteristics of algorithms is, but also how important it is to understand the data that is used within the algorithm, and if the design is built for harmony between the two.

Code Appendix

SIR Model Without Vaccination

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Clint Olsen
% ECEN 5322: Higher-Dimensional Datasets
% Final Project
% SIR Model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% - Clint_SIR.m: Runs the SIR model on a given adjacency matrix
%
% - Inputs: Adjacency matrix (A), Infection rate (Beta), Recovery rate (mu)
%
% - Outputs: Number of recovered nodes and count distributions for
% infected, susceptible and recovered nodes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [numRecovered, infectedDistribution, susceptibleDistribution, recoveredDistribution] = Clint_SIR(A, Beta, mu)

    % Time step
    % Determine so probability is normalized between 0 and 1 for
    med = median(A(find(A~=0)));
    dt = (1/Beta)*(1/med)*.001;

    %Generate random seedm for 1st node to be infected
    numInfected = 1;
    totalInfected = 1;
    infected = zeros(1,size(A,1));
    infected(1,randi([1 size(A,1)],1,1)) = 1;

    numRecovered = 0;
    plotIndex = 1;

    numSusceptible = size(A,1) - 1;
    susceptible = ones(1,size(A,1));
    susceptible(1,find(infected == 1)) = 0;
    while(numInfected > 0) % This is when the infection dies out

        infectedList = find(infected == 1); % Pick an infected node
        infectedIndex = infectedList(1);
           for j=2:size(A,1)    % Loop through neighbors of that infected node
               if(A(infectedIndex,j) > 0 && susceptible(j) == 1) % If index is a neighbor and is susceptible
                   if(binornd(1,Beta*A(infectedIndex,j)*dt) >= 1) % Node will become infected
                        numInfected = numInfected + 1;
                        infected(j) = 1; % Infect the Node
                        susceptible(j) = 0; % Infected implies no longer susceptible
                        numSusceptible = numSusceptible - 1;

                        % Collect counts for returned data sets (for plots)
                        susceptibleDistribution(plotIndex) = numSusceptible;
                        infectedDistribution(plotIndex) = numInfected;
                        recoveredDistribution(plotIndex) = numRecovered;
                        plotIndex = plotIndex + 1;

                   end
               end
           end

            %Probability current infected node will transition to recovered
            % The current calculation for dt makes this value very small,
            % so the likelihood of all nodes getting infected is much
            % higher due to mu decreasing with increased k (increasing the
            % strength of the spread) rho becomes larger and larger than
            % 1
            if(binornd(1,mu*dt) >= 1) % Node will become recovered
                infected(infectedIndex) = 0;
                numRecovered = numRecovered + 1; % Node Recovers
                numInfected = numInfected - 1;
                susceptible(infectedIndex) = 0;

                % Collect counts for returned data sets (for plots)
                infectedDistribution(plotIndex) = numInfected;
                recoveredDistribution(plotIndex) = numRecovered;
                susceptibleDistribution(plotIndex) = numSusceptible;
                plotIndex = plotIndex + 1;

            end

    end

end
```

SIR Model With Vaccination

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Clint Olsen
% ECEN 5322: Higher-Dimensional Datasets
% Final Project
% SIR Model with Vaccination
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% - Clint_SIR_Vac.m: Runs the SIR model on a given adjacency matrix
%
% - Inputs: Adjacency matrix (A), Infection rate (Beta), Recovery rate (mu)
%
% - Outputs: Number of recovered nodes and count distributions for
% infected, susceptible and recovered nodes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [numRecovered, infectedDistribution, susceptibleDistribution, recoveredDistribution] = Clint_SIR_Vac(A, Beta, mu, rand)

    % Time step
    % Determine so probability is normalized between 0 and 1 for
    med = median(A(find(A~=0)));
    dt = (1/Beta)*(1/med)*.001;

    %Generate random seedm for 1st node to be infected
    numInfected = 1;
    totalInfected = 1;

    infected = zeros(1,size(A,1));
    infected(1,randi([1 size(A,1)],1,1)) = 1;
    numRecovered = 0;
    plotIndex = 1;

    % Compute degrees
    for i=1:size(A,1)

        degrees(i) = sum(A(i,:));

    end

    numSusceptible = size(A,1) - 21;
    susceptible = ones(1,size(A,1));
    if(rand == "true")
        %vaccinated = randperm(size(A,1), 20);
        vaccinated = randperm(size(A,1), floor(.25*size(A,1)));
    else
        [v,index] = sort(degrees, 'descend');
        %vaccinated = index(1:20);
        vaccinated = index(1:floor(.25*size(A,1)));
    end
    susceptible(vaccinated) = 0;
    susceptible(1,find(infected == 1)) = 0;

    while(numInfected > 0) % This is when the infection dies out

        infectedList = find(infected == 1); % Pick an infected node
        infectedIndex = infectedList(1);
           for j=2:size(A,1)    % Loop through neighbors of that infected node
               if(A(infectedIndex,j) > 0 && susceptible(j) == 1) % If index is a neighbor and is susceptible
                   if(binornd(1,Beta*A(infectedIndex,j)*dt) >= 1) % Node will become infected
                        numInfected = numInfected + 1;
                        infected(j) = 1; % Infect the Node
                        susceptible(j) = 0; % Infected implies no longer susceptible
                        numSusceptible = numSusceptible - 1;

                        % Collect counts for returned data sets (for plots)
                        susceptibleDistribution(plotIndex) = numSusceptible;
                        infectedDistribution(plotIndex) = numInfected;
                        recoveredDistribution(plotIndex) = numRecovered;
                        plotIndex = plotIndex + 1;

                    end
               end
           end

            %Probability current infected node will transition to recovered
            % The current calculation for dt makes this value very small,
            % so the likelihood of all nodes getting infected is much
            % higher due to mu decreasing with increased k (increasing the
            % strength of the spread) rho becomes larger and larger than
            % 1
            if(binornd(1,mu*dt) >= 1) % Node will become recovered
                infected(infectedIndex) = 0;
                numRecovered = numRecovered + 1; % Node Recovers
                numInfected = numInfected - 1;
                susceptible(infectedIndex) = 0;

                % Collect counts for returned data sets (for plots)
                infectedDistribution(plotIndex) = numInfected;
                recoveredDistribution(plotIndex) = numRecovered;
                susceptibleDistribution(plotIndex) = numSusceptible;
                plotIndex = plotIndex + 1;

            end

    end

end
```