P5 Virus Propagation

CSC 591 Graph Data Mining

Group Members –

Name	Unity Id	Student Id
Mrunmayi Deshpande	mdeshpa4	200263235
Alisha Shahane	asshahan	200311941

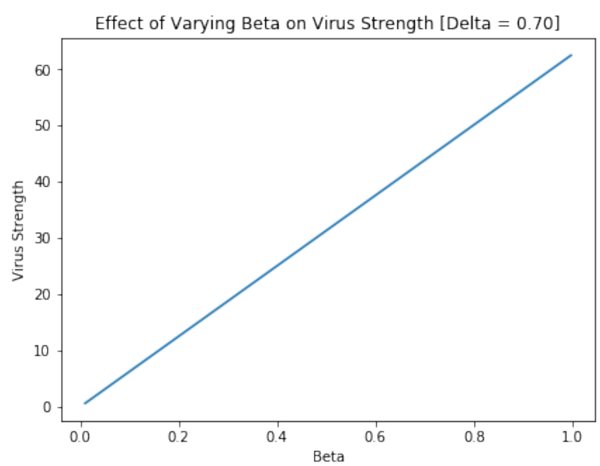
Q1 SIS Model

a) Will the infection spread across the network (i.e., result on an epidemic), or will it die quickly?

Since the effective strength is 12.53 ie greater than 1, the virus will result in an epidemic.

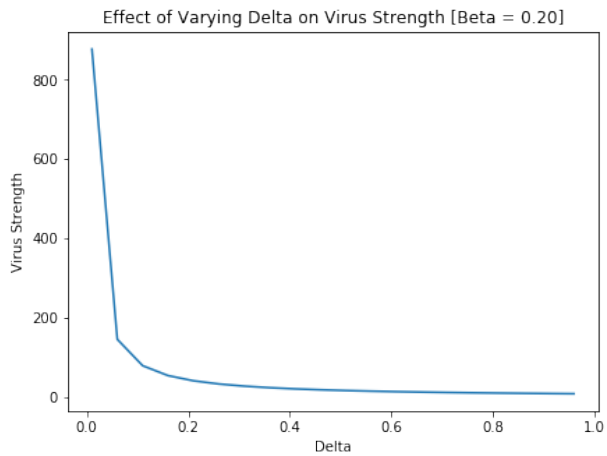
b) Keeping δ fixed, analyze how the value of β affects the effective strength of the virus. What is the minimum transmission probability (β) that results in a network-wide epidemic?

When $\delta_1 = 0.70$ minimum beta for which virus strength > 1 and therefore results in an epidemic is : 0.0175.



c) Keeping β fixed, analyze how the value of δ affects the effective strength of the virus. What is the maximum healing probability (δ) that results in a networkwide epidemic?

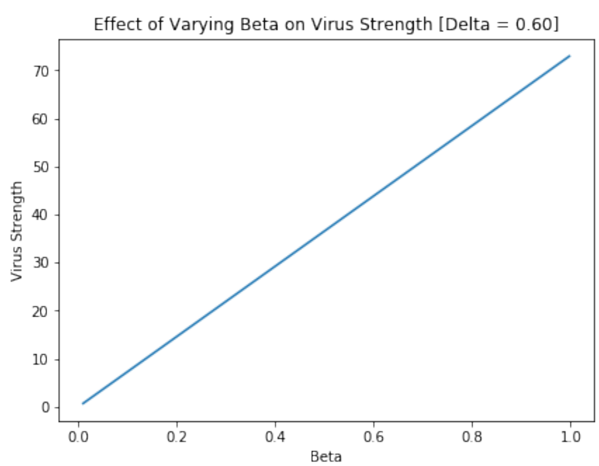
When $\beta 1 = 0.20$ maximum delta for which virus strength > 1 and therefore results in an epidemic is : 0.96



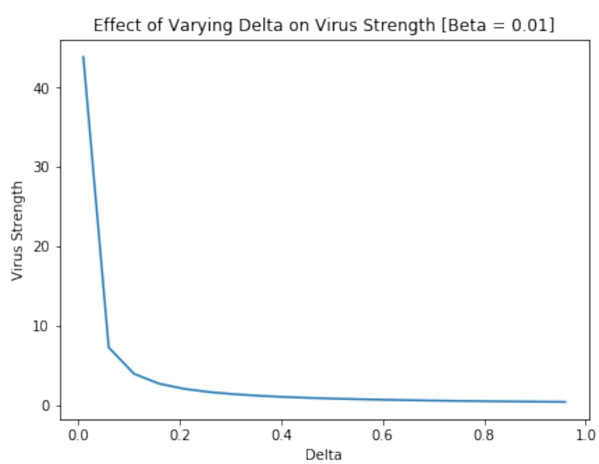
d) Repeat (1), (1a), (1b) and (1c) with $\beta = \beta_2$, and $\delta = \delta_2$.

i. The effective strength is 0.73 ie less than 1, so the epidemic is prevented.

ii. When $\delta 2 = 0.60$ minimum beta for which virus strength > 1 and therefore results in an epidemic is : 0.015.



iii. When $\beta 2 = 0.01$ maximum delta for which virus strength > 1 and therefore results in an epidemic is : 0.41

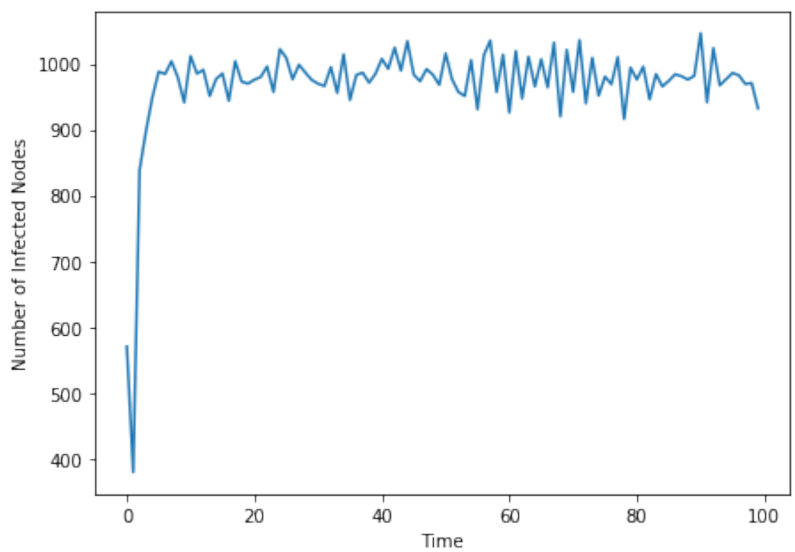


Q2 Simulation

a) Run the simulation program 10 times for the static contact network provided (*static.network*), with $\beta = \beta_1$, $\delta = \delta_1$, $c = n/10$ (n is the number of nodes in the network), and $t = 100$.

```
Running Virus Propagation Simulation

PARAMETERS : β =  0.2        δ =  0.7       k =  200       t = 100

Maximum Eigen Value of Graph :  43.85

Effective Virus Strength :  12.53

Average Number of Nodes Initially Infected :  571.0

Average Number of Nodes Finally Infected :  932.3

RESULT : Virus has caused an epidemic
```

b) Plot the average fraction of infected nodes at each time step. Did the infection spread across the network, or did it die quickly? Do the results of the simulation agree with your conclusions in (1a)?



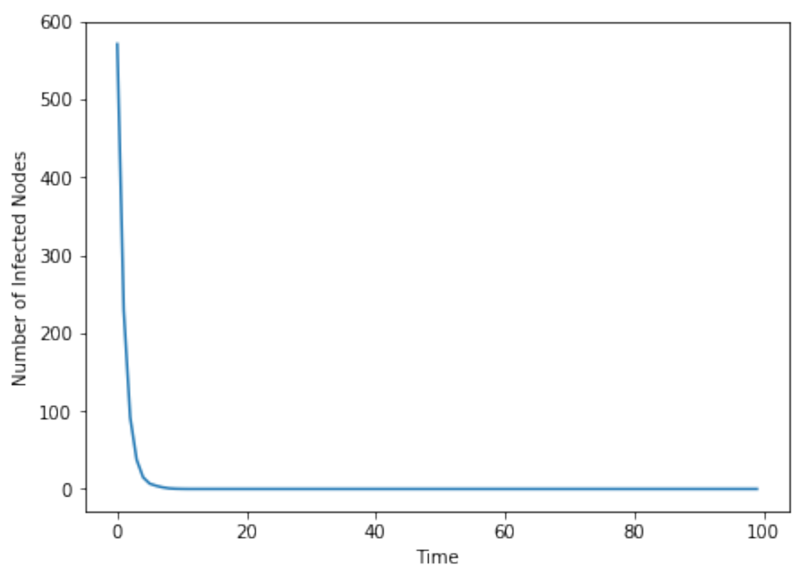
The virus spread throughout the network resulting in an epidemic, which is the same we expected as in 1 a.

c) Repeat (2a) and (2b) with $\beta = \beta_2$, and $\delta = \delta_2$.

i.

```
Running Virus Propagation Simulation

PARAMETERS : β =  0.01       δ =  0.6       k =  200       t = 100

Maximum Eigen Value of Graph :  43.85

Effective Virus Strength :  0.731

Average Number of Nodes Initially Infected :  571.0

Average Number of Nodes Finally Infected :  0.0

RESULT : Virus epidemic was prevented
```

ii.



The infection in the network quickly died out, which is the same we expected as in 1 d.

Q3 Immunization

a) What do you think would be the optimal immunization policy? What would be its time complexity? Would it be reasonable to implement this policy? Justify.

The most optimal immunization policy is immunizing the k nodes whose removal causes largest drop in highest eigen value of the graph ($\lambda 1$). However, this would require calculating the $\lambda 1$ drop for each possible subset of k nodes, resulting in a time complexity of $O(n^k)$. This would be computationally infeasible.

b) Intuition behind heuristic

Policy A

This immunization policy randomly selects k nodes. The only intuition here is that if we run it several times, it will succeed in containing the epidemic at least few times.

Policy B

This immunization policy selects the k highest degree nodes to immunize. The intuition here is that the nodes with higher degree are connected to maximum nodes and have higher chance of spreading the infection so immunizing these nodes should help contain the epidemic.

Policy C

This immunization policy is a modification of policy B. Here too we select the k highest degree nodes to immunize. But the change is that we iteratively remove the highest degree node and re-calculate the adjacency matrix after each removal. This is repeated k times. The intuition here is that each removal results in changes to the graph so it is necessary to remove them one at a time.

Policy D

This immunization policy selects k nodes corresponding to the largest values in the eigen vector of the highest eigen value to immunize. The intuition here is that removing these nodes will result in decrease of largest eigen value and therefore cause decrease in virus strength.

c) Pseudocode and Time Complexity of Immunization Policy

Policy A

```
procedure policy_a(graph, k):
    n <- get_nodes(graph)
    immunize_nodes <- choose_random(n, k)
    remove_from_graph(immunize_nodes)
```

Time Complexity -
O(k) for selecting and removing k nodes; $O(kn^2)$ for updating adjacency matrix. Therefore, overall time complexity is **$O(kn^2)$.**

Policy B

```
procedure policy_a(graph, k):
    n <- get_nodes(graph)
    degree <- get_degrees(graph)
    sorted_nodes <- sort(degrees)
    immunize_nodes <- sorted_nodes(1:k)
    remove_from_graph(immunize_nodes)
```

Time Complexity -
$O(n^2)$ for calculatig degree matrix; O(nlogn) for sorting degrees; $O(kn^2)$ for updating adjacency matrix. Therefore overall time complexity is **$O(kn^2)$.**

Policy C

```
procedure: policy_a(graph, k):
       for i : 1 -> k times:
          n <- get_nodes(graph)
          degree <- get_degrees(graph)
          sorted_nodes <- sort(degrees)
          immunize_node <- sorted_nodes[0] # highest degree node
          remove_from_graph(immunize_node)
       end for
```

Time Complexity -
$O(kn^2)$ for calculatig degree matrix; O(knlogn) for sorting degrees; $O(k^2n^2)$ for updating adjacency matrix. Therefore time complexity is **$O(k^2n^2)$.**

Policy D

```
procedure: policy_a(graph, k):
       n <- get_nodes(graph)
       eigen_val, eigen_vec <- get_top_eigen_value_vector(graph)
       sorted_node_index <- sort(absolute_values(eigen_vec))
       immunize_node <- sorted_nodes[1:k]
       remove_from_graph(immunize_node)
```

Time Complexity -
$O(n^2)$ for calculatig eigen value/vector; $O(n\log n)$ for sorting eigen vector; $O(kn^2)$ for removing nodes and updating adjacency matrix. Therefore overall time complexity is **$O(kn^2)$.**

d) Calculate the effective strength (s) of the virus on the immunized contact network. Did the immunization policy prevent a networkwide epidemic?

Policy A
Maximum Eigen Value = 43.20
The effective strength of the virus is 12.34
Result - The virus will result in a network-wide epidemic.

Policy B
Maximum Eigen Value = 3.78
The effective strength of the virus is 1.08
Result - The epidemic will be prevented.

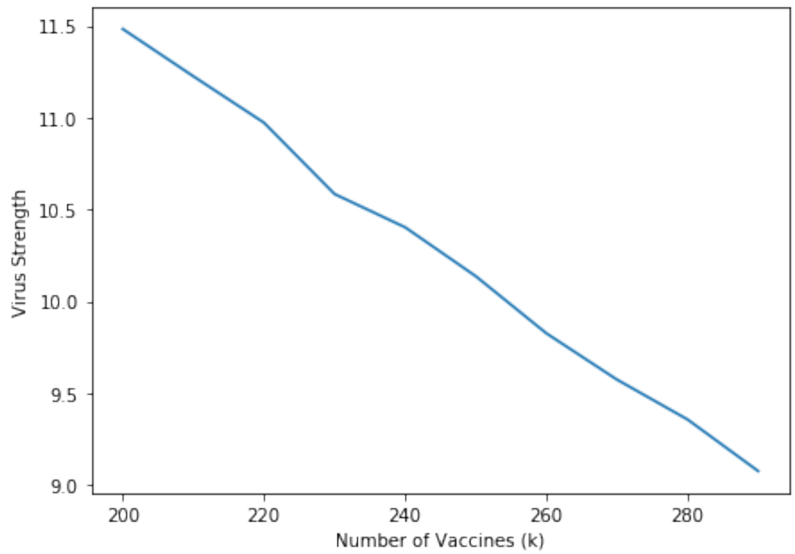
Policy C
Maximum Eigen Value = 3.80
The effective strength of the virus is 1.08
Result - The epidemic will be prevented.

Policy D
Maximum Eigen Value = 10.74
The effective strength of the virus is 3.07
Result - The epidemic will be prevented.

e) Keeping β and δ fixed, analyze how the value of *k* affects the effective strength of the virus on the immunized contact network. Estimate the minimum number of vaccines necessary to prevent a networkwide epidemic.

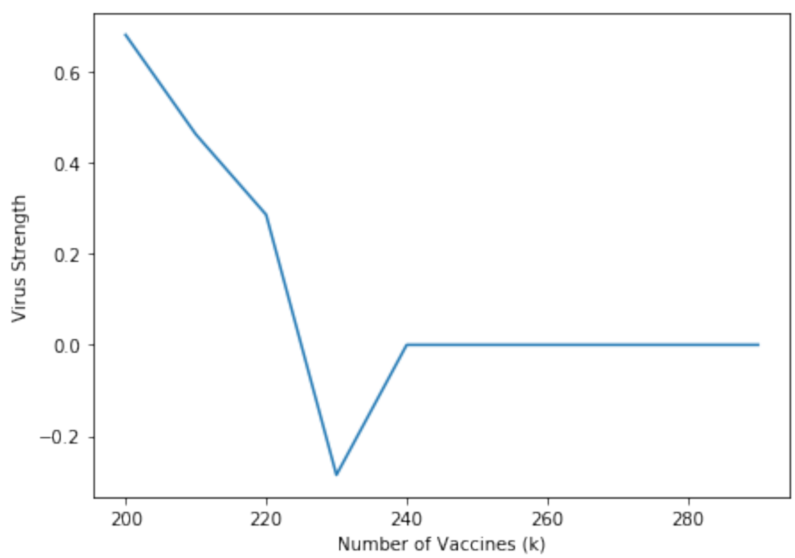
Policy A

This number is very large. The policy is unable to prevent an epidemic even when the number of vaccines is as high as 300, the virus strength is around 9.



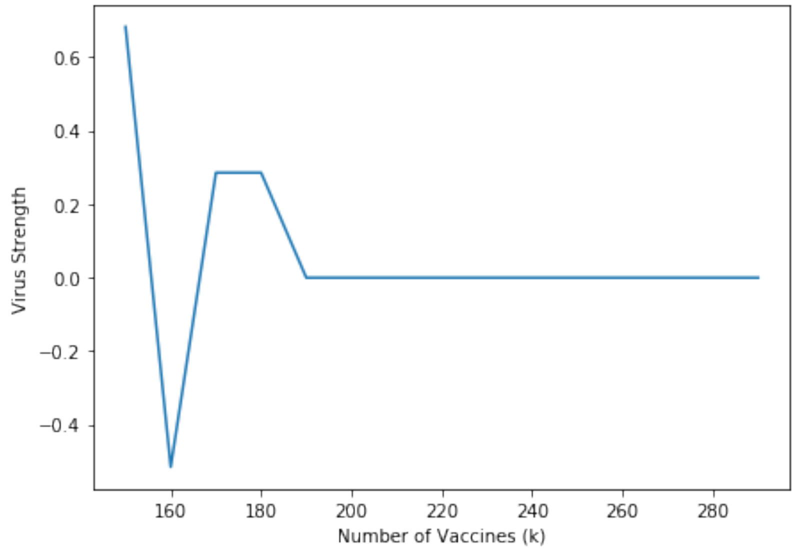
Policy B

Around 240 vacciness are sufficient to prevent the epidemic.



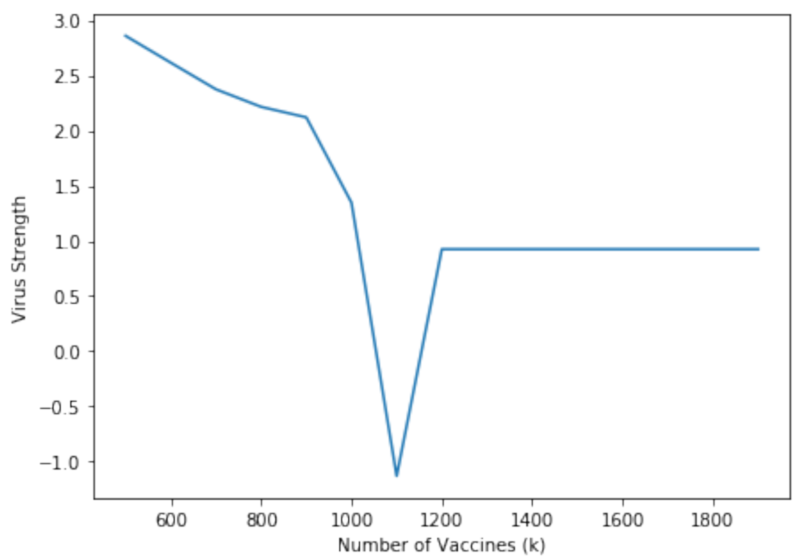
Policy C

Around 190 vacciness are sufficient to prevent the epidemic. This is fewer than policy B which is expected as this policy is an improvement on B.



Policy D

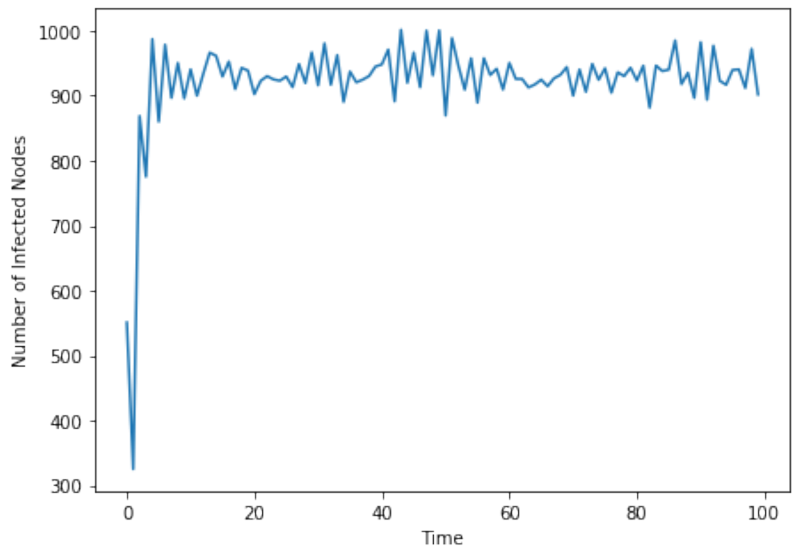
A large number of vaccines, around 1300 vaccines are required to prevent the epidemic.



f) Given $k=k_1$, $\beta=\beta_1$,$\delta=\delta_1$, $c=n/10$, and $t=100$, run the simulation from problem (2) for the immunized contact network 10 times. Plot the average fraction of infected nodes at each time step. Do the results of the simulation agree with your conclusions in (3d)?

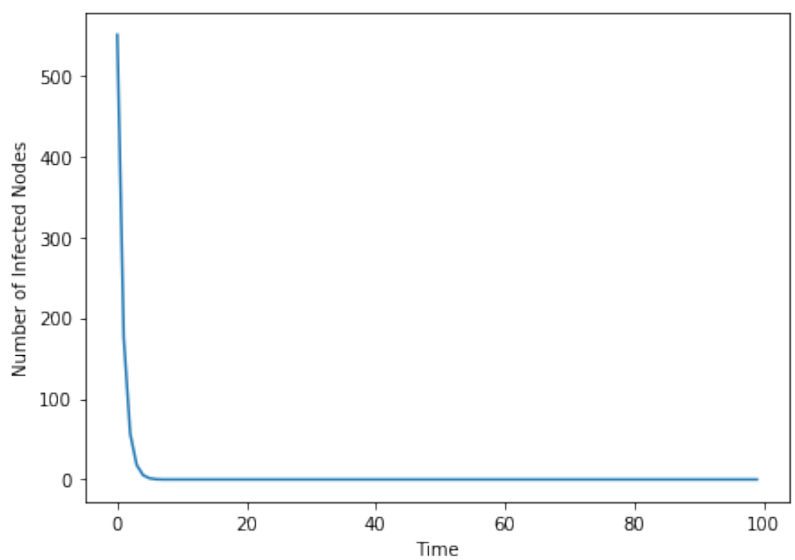
Policy A

The simulation plot shows that the virus always persists and results in an epidemic, which is consistent with the result in 3d.



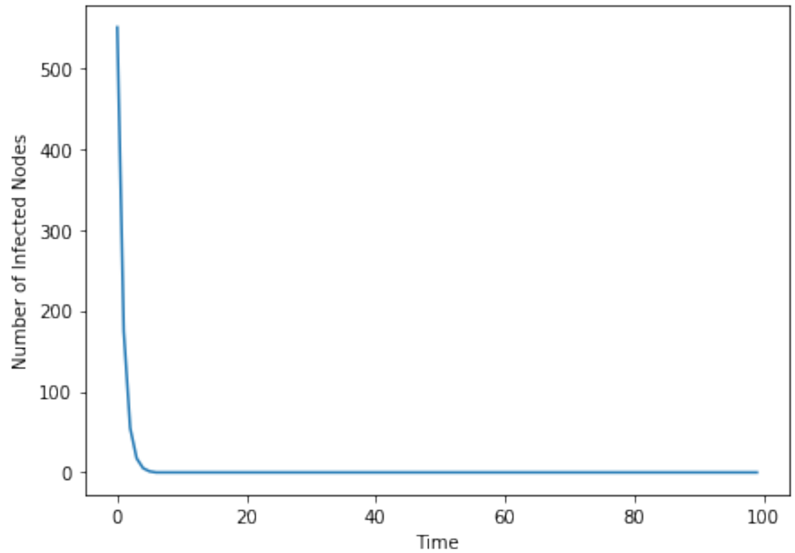
Policy B

The simulation plot shows that the epidemic is contained, which is consistent with the result in 3d.



Policy C

The simulation plot shows that the epidemic is contained, which is consistent with the result in 3d.



Policy D

The simulation plot shows that the epidemic is contained, which is consistent with the result in 3d.