**Exercise four: Influence maximization**

In the lecture, we discussed a few heuristics for the influence maximization problem in social network. Apply degree heuristics and betweenness heuristics to the IC model you have developed in Question 11 (! Please change the initially infected node to S107!). Answer the following questions (Question 17, 5 points):

1. You can immunize 3 nodes in the network, which after immunization, will never spread the virus to other connected nodes. According to degree heuristics and betweenness heuristics, which 3 nodes should be immunized in order to contain the virus?

Answer:

* according to degree heuristic: S54, S20, S110
* according to betweenness heuristic: S37, S4, S96

1. Immunize the 3 nodes suggested by degree heuristics and betweenness heuristics, respectively, which heuristic provides the better outcome regarding a) the final activated number of people and b) flattening the daily infection curve (please provide figure in your answer)?

*Answer:*

**Table \_ Average final number of activated nodes in High School data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Probability of contagion** | **No** immunity | **Degree Heuristic** | **Betweenness Heuristic** |
| Final number of activated nodes | | |
| **0.5** | 122 | 119 | 119 |
| **0.15** | 122 | 119 | 119 |
| **0.1** | 120 | 113 | 108 |
| **0.06** | 91 | 55 | 50 |
| **0.01** | 9 | 7 | 4 |

Chart, line chart

Description automatically generated

So, from both numbers and the picture we see that betweenness centrality heuristic prevents the spread at least several nodes better in terms of final number of activations. However, IC model has a random character meaning that our results for final activation numbers are not representative, which made us to perform the IC model 100 times for the daily infection curve. From the results it is evident that if the probability of contagion is high enough, the whole network will be activated, but in a case of 0.1 and smallerheuristics may prevent the contagion of the whole network.

While the betweenness heuristic slows down the virus spread better, both heuristics flatten the curve. The performance of betweenness heuristic may be explained with its trait more global characteristic: while the degree considers the properties of a given node separately (i.e. the number of its connections), the betweenness takes into account the shortest paths from the whole graph.

1. Do you think the observation in 2) (i.e., degree heuristic preforms better than betweenness heuristics, or the opposite) is sensitive to a) the network structure and b) parameter in the IC model? And Why?

With regard to the contagion mechanism, as already mentioned in the previous question, two heuristics do not vary in terms of final activation number, however they differ in terms of the time, needed to cover every possible node (i.e. left skewness of the curve).

With regard to the dependency of the heuristic effectiveness on the network structure, from Table \_ and \_ it is evident that betweenness heuristic constrains the spread more effectively in Barabasi models but have little difference in Small model and in Highschool model. The reason behind this we have partially covered in the previous question, claiming that betweenness considers more global traits of the network than degree. None the less, the more rigorous research in this topic of F. Morone, H. Makse claim that betweenness heuristic does not outperform other metrics (2015).

For small world model as well as for Barabasi model, we increased the number of days to figure out at what probability the whole network will be covered, and the bigger time period needed as the networks taken from the previous exercises are bigger than Highschool data.

Morone, F., & Makse, H. A. (2015). Influence maximization in complex networks through optimal percolation. Nature, 524(7563), 65-68.

|  |  |
| --- | --- |
| Chart, line chart  Description automatically generated | Chart, line chart  Description automatically generated |

**Table \_ Average final number of activated nodes in Barabasi model with superliner probability dependency**

***(power 1.5, size 300, seed node randomly selected every iteration)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Probability of contagion** | **No immunity** | **Degree Heuristic:** | **Betweenness Heuristic:** |
| Final number of activated nodes | | |
| **0.5** | 299 | 288 | 159 |
| **0.15** | 276 | 266 | 157 |
| **0.1** | 233 | 227 | 135 |
| **0.06** | 49 | 50 | 36 |
| **0.01** | 3 | 3 | 1 |

**Table \_ Average final number of activated nodes in Small world model**

***(size 300, seed node randomly selected every iteration)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Probability of contagion** | **No immunity** | **Degree Heuristic:** | **Betweenness Heuristic:** |
| Final number of activated nodes | | |
| **0.5** | 300 | 297 | 297 |
| **0.15** | 300 | 297 | 297 |
| **0.1** | 294 | 291 | 292 |
| **0.06** | 213 | 217 | 206 |
| **0.01** | 3 | 3 | 1 |

(Important note: In Question 12, the initially infected node is S5. Please change it to S107 to answer Question 18. In other words, at Day 1, an infected node (N0, node ID= S107) is introduced to the network.)

In addition to heuristics, we also introduced the greedy algorithm in the lecture. Develop a greedy algorithm to the IC model you have developed in Question 11 (! Please change the initially infected node to S107!). Answer the following questions (Question 18, 5 points):

1. You can immunize 3 nodes in the network, which after immunization, will never spread the virus to other connected nodes. According to greedy algorithm, which 3 nodes should be immunized in order to contain the virus?

Answer: S37, S21, S9

1. Compared to the result from greedy algorithm to those from degree heuristic and betweenness heuristic. Regarding a) the final activated number of people and b) flattening the daily infection curve (please provide figure in your answer), does greedy algorithm provide the best result? And explain the reason.

*Answer:* As well as the betweenness heuristic, the greedy algorithm suggests immunizing node number 37. However, betweenness degree heuristic still outperforms the greedy in terms of peak reduction and curve flattening. This implies that with greedy we only found a local extremum in the objective function of IMP problem, meaning that by adding those nodes one by one and constraining the number of iterated combinations, the greedy algorithm is restricted by those nodes, which are added in the earlier iterations.

However it is proven that greedy algorithm approximates the optimum solution[], in our case it has not reached it, as the set of immunized nodes chosen with betweenness heuristic performs better. With greedy algorithm we add nodes one by one, achieving the Influence maximization with linear combinations of nodes ordered by their degree of influence. But the desired combination of node may produce its effect due to a relational nature but not due to a cumulative nature. So independently from each other, or better say dependent not fully on each other but only on the preceding node, the set of { S37, S21, S9} is less influential then set of { S37, S4, S96}.

Kempe, D., Kleinberg, J., & Tardos, É. (2003, August). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 137-146).

Fig. \_

Chart, line chart

Description automatically generated

**Table \_ Average final number of activated nodes   
for Greedy on Highschool data**

|  |  |  |
| --- | --- | --- |
| **Probability of contagion** | **Greedy** | **Betweenness Heuristic:** |
| Final activated number | |
| 0.5 | 119 | 119 |
| 0.15 | 117 | 119 |
| 0.1 | 99 | 108 |
| 0.1 | 56 | 50 |
| 0.01 | 4 | 4 |

(Important note: In Question 12, the initially infected node is S5. Please change it to S107 to answer Question 18. In other words, at Day 1, an infected node (N0, node ID= S107) is introduced to the network. Please submit the codes of this question along with your answer.)

# Import packages

%matplotlib inline

import matplotlib.pyplot as plt

from random import uniform, seed

import numpy as np

import pandas as pd

import csv

import time

from igraph import \*

import cairocffi

import cairo

with open('Highschool\_network\_edge.csv') as csvfile:

reader = csv.reader(csvfile)

edges = [(int(row[0][1:]), int(row[1][1:])) for row in reader]

g= Graph(edges, directed=False)

NodeID = []

Gender = []

Hall = []

Threshold = []

with open('Highschool\_network\_att.csv') as csvfile:

reader = csv.reader(csvfile)

next(reader)

for row in reader:

NodeID.append(int(row[0][1:]))

Gender.append(row[1])

Hall.append(row[2])

Threshold.append(row[3])

g.vs["NodeID"] = NodeID

g.vs["Gender"] = Gender

g.vs["Hall"] = Hall

g.vs["Threshold"] = Threshold

def IC(g,S,p=0.5,mc=1000, timestamps = 28):

"""

Input:

g - graph object,

S - set of seed nodes(dtype list)

p - propagation probability

mc - the number of Monte-Carlo simulations

timestamps - the number of timestamps

Output:

- average number of nodes activated in each Monte-Carlo simulation

- average number of nodes influenced by the seed nodes in each timestamp

"""

# Loop over the Monte-Carlo Simulations

spread = []

for i in range(mc):

# Simulate propagation process

new\_active, A = S[:], S[:]

sum\_spread = 0

#while new\_active:

for \_ in range(timestamps):

# For each newly active node, find its neighbors that become activated

new\_ones = []

for node in new\_active:

# Determine neighbors that become infected

np.random.seed(i)

success = np.random.uniform(0,1,len(g.neighbors(node,

mode="out"))) < p

new\_ones += list(np.extract(success, g.neighbors(node,mode="out")))

new\_active = list(set(new\_ones).union(set(A)))

sum\_spread += len(new\_active) - len(A)

# in case the network is fully activated

if new\_active == A:

break

# Add newly activated nodes to the set of activated nodes

A = new\_active

spread.append(len(A))

return(np.mean(spread), np.mean(sum\_spread))

def greedy(g, k, p=0.1,mc=1000, timestamps = 28):

"""

Input:

g - graph object

k - number of seed nodes

Output: optimal seed set, resulting spread, time for each iteration

"""

S, spread, timelapse, start\_time = [], [], [], time.time()

list = [i for i in range(g.vcount())]

list.remove(107)

list.insert(0, 107)

# Find k nodes with largest marginal gain

for \_ in range(k):

# Loop over nodes that are not yet in seed set to find biggest marginal gain

best\_spread = 0

#for j in set(range(g.vcount())) - set(S):

for j in (set(list) - set(S)):

# Get the spread

s = IC(g, S + [j], p, mc, timestamps)

# Update the winning node and spread so far

if s[1] > best\_spread:

best\_spread, node = s[1], j

# Add the selected node to the seed set

S.append(node)

# Add estimated spread and elapsed time

spread.append(best\_spread)

timelapse.append(time.time() - start\_time)

return(S,spread, timelapse)

Next you will study the influence maximization problem in the threshold model. Following below steps to answer Question 19 (8 points):

1. Use the threshold model for the “once-a-beef” campaign you build in Question 14, reset the seed nodes to a null set;
2. According to degree heuristics, which nodes should be included in the seed set in order to maximize the spread of the campaign? The size of seed set is 7, i.e., you can choose 7 nodes to activate to kick off the contagion process.
3. According to betweenness heuristics, which nodes should be included in the seed set in order to maximize the spread of the campaign? The size of seed set is 7, i.e., you can choose 7 nodes to activate to kick off the contagion process.
4. According to greedy algorithm, which nodes should be included in the seed set in order to maximize the spread of the campaign? The size of seed set is 7, i.e., you can choose 7 nodes to activate to kick off the contagion process.
5. Compared the results from degree heuristics, betweenness heuristics and greedy algorithm, which method provides the best outcome? Please show the results of three methods in a figure.

(Please submit the codes of this question along with your answer.)

Question 20 (7 points): You have compared the performance of degree heuristics, betweenness heuristics and greedy algorithm. Can you propose an even more efficient algorithm (i.e., achieve even higher diffusion rate with even less percentage of nodes using as seeds)? Answer this question by the following steps:

1. Description of your algorithm and the reason why you think it will be more effective;
2. Test your algorithm in the threshold model for the “once-a-beef” campaign used in Question 19; Comments on its effectiveness;
3. Test your algorithm in a large real-world network (e.g., n≥1000) using threshold model. For the larger network, you can choose one from the [database](https://snap.stanford.edu/data/), and make a subgraph from it (e.g., choose only 1000 nodes). Describe the network you choose, your setting about threshold, and comment on the effectiveness of your algorithm on this network.

(Note: The proposed algorithm does not need to excel in all the settings, but should outperform at least some of the existing heuristics or greedy algorithm in some scenarios. And please submit the codes of this question along with your answer.)