The Chinese University of Hong Kong
Academic Honesty Declaration Statement

Submission Details

Student Name	CHIM, Ka Long (1155094482)		
Year and Term	2019-2020 Term 2		
Course	CSCI-4190-- Introduction to Social Networks		
Assignment Marker	Professor CHAN Lai Wan		
Submitted File Name	Project_1155094482.docx		
Submission Type	Individual		
Assignment Number	1	**Due Date (provided by student)**	2020-05-04
Submission Reference Number	2605957	**Submission Time**	2020-05-04 16:53:01

Agreement and Declaration on Student's Work Submitted to VeriGuide

VeriGuide is intended to help the University to assure that works submitted by students as part of course requirement are original, and that students receive the proper recognition and grades for doing so. The student, in submitting his/her work ("this Work") to VeriGuide, warrants that he/she is the lawful owner of the copyright of this Work. The student hereby grants a worldwide irrevocable non-exclusive perpetual licence in respect of the copyright in this Work to the University. The University will use this Work for the following purposes.

(a) Checking that this Work is original
The University needs to establish with reasonable confidence that this Work is original, before this Work can be marked or graded. For this purpose, VeriGuide will produce comparison reports showing any apparent similarities between this Work and other works, in order to provide data for teachers to decide, in the context of the particular subjects, course and assignment. However, any such reports that show the author's identity will only be made available to teachers, administrators and relevant committees in the University with a legitimate responsibility for marking, grading, examining, degree and other awards, quality assurance, and where necessary, for student discipline.

(b) Anonymous archive for reference in checking that future works submitted by other students of the University are original
The University will store this Work anonymously in an archive, to serve as one of the bases for comparison with future works submitted by other students of the University, in order to establish that the latter are original. For this purpose, every effort will be made to ensure this Work will be stored in a manner that would not reveal the author's identity, and that in exhibiting any comparison with other work, only relevant sentences/ parts of this Work with apparent similarities will be cited. In order to help the University to achieve anonymity, this Work submitted should not contain any reference to the student's name or identity except in designated places on the front page of this Work (which will allow this information to be removed before archival).

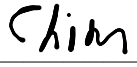
(c) Research and statistical reports
The University will also use the material for research on the methodology of textual comparisons and evaluations, on teaching and learning, and for the compilation of statistical reports. For this purpose, only the anonymously archived material will be used, so that student identity is not revealed.

I confirm that the above submission details are correct. I am submitting the assignment for:

[X] an individual project.

I have read the above and in submitting this Work fully agree to all the terms. I declare that: (i) the assignment here submitted is original except for source material explicitly acknowledged; (ii) the piece of work, or a part of the piece of work has not been submitted for more than one purpose (e.g. to satisfy the requirements in two different courses) without declaration; and (iii) the submitted soft copy with details listed in the <Submission Details> is identical to the hard copy(ies), if any, which has(have) been / is(are) going to be submitted. I also acknowledge that I am aware of the University's policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the University website http://www.cuhk.edu.hk/policy/academichonesty/.

I also understand that assignments without a properly signed declaration by the student concerned will not be graded by the teacher(s).



Signature (CHIM, Ka Long, 1155094482)

2020-5-4

Date

Instruction for Submitting Hard Copy / Soft Copy of the Assignment

This signed declaration statement should be attached to the hard copy assignment or submission to the course teacher, according to the instructions as stipulated by the course teacher. If you are required to submit your assignment in soft copy only, please print out a copy of this signed declaration statement and hand it in separately to your course teacher.

CSCI 4190 Introduction to Social Networks

Project Report

Task 3: Simulate cascading behaviors in networks

Chim Ka Long 1155094482

1 Abstract

This project involves a simulation of cascading behaviors in networks. It is about how an innovation or idea spread through the network. The dataset used is a real-world dataset of a technology-related news website. By controlling a few factors and parameters, observing how an idea spread through the network. This social network analysis is implemented by Python langrage and Stanford Network Analysis Platform (SNAP).

2 Objective

Cascading behaviors in networks simulates how an idea spread in a social network. For example, suppose all users in network are using social media App A in the beginning, a new social media App B is developed, and number of initial adopters start using it. If other users whether using App B is determined by fraction of their friends using App B, how many users will use App B eventually. Cascading behavior in network is mainly affected by payoff (threshold), number of initial adopters, method of selecting initial adopters. We want to analysis how these factors affecting the cascading process and result.

3 Methodology

3.1 Data source

A Slashdot dataset is selected to be analyzed in this project. Slashdot is a technology-related news website which allow users tag other as friends. The detail of dataset can be viewed and downloaded from SNAP website (http://snap.stanford.edu/data/soc-Slashdot0902.html). The original data set is directed graph. Considering the nature of sharing idea is bi-directional process, we import the dataset as undirected graph. All single direction edge will be imported as bi-direction edge. Also, the edge pointing to self-nodes will be eliminated.

3.2 Tool

SNAP 5.0 for Python is used to analysis the dataset. Also, the related program is developed by python 3.7 in Windows 10 platform. Matplotlib 3.2.1 is used to plot graph. The following is the list of related package versions.

```
C:\Users\jackc\Desktop\lab report\CSCI4190>pip list
Package                Version
---------------------- -------
cycler                 0.10.0
kiwisolver             1.2.0
matplotlib             3.2.1
mysql-connector-python 8.0.19
numpy                  1.18.3
pip                    20.0.2
pyparsing              2.4.7
python-dateutil        2.8.1
setuptools             39.0.1
six                    1.14.0
snap-stanford          5.0.0

C:\Users\jackc\Desktop\lab report\CSCI4190>python -V
Python 3.7.0
```

3.3 Experiment Design

Suppose all nodes are adopting action A, except some nodes are adopting action B in the beginning. The node will adopt action B, if the fraction of his friends (neighbor nodes connected by edge) equal or exceed the threshold. The experiment is implemented with homogenous threshold, such that every node has same threshold to determine whether adopt action B.

The general algorithm of cascading is below:

1. Randomly choose some nodes as initial adopters (adopt action B in the beginning). Put their neighbors who still adopt action A into a queue (First In First Out)
2. Pop a node from queue. Calculate the fraction of his friend using action B. If the fraction equals or exceed the certain threshold, this node will adopt action B, put all his friends who still adopt action A into queue.
3. Repeat process 2 until the queue is empty.

The above algorithm is like breadth first search, such that the order of processing nodes depending on the distance.

We will do 3 experiments:

1. Threshold vs cascading
2. Adopting key nodes vs cascading
3. Complete cascade vs clustering

The detail of experiment and result will be shown on part 5 experiment and result.

4 Data Statistics

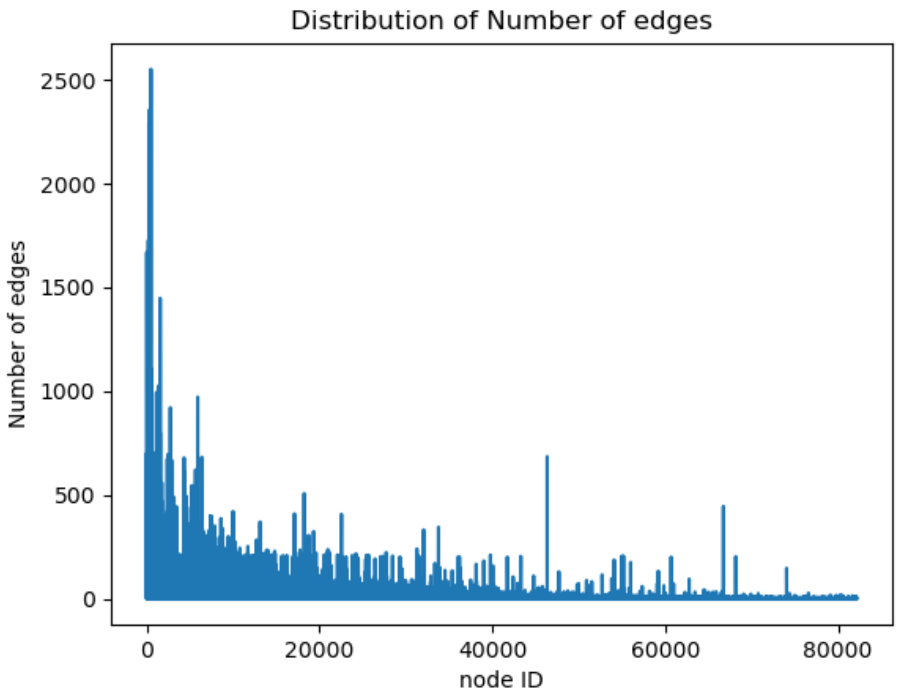
Below data can be generated again by p0.py.

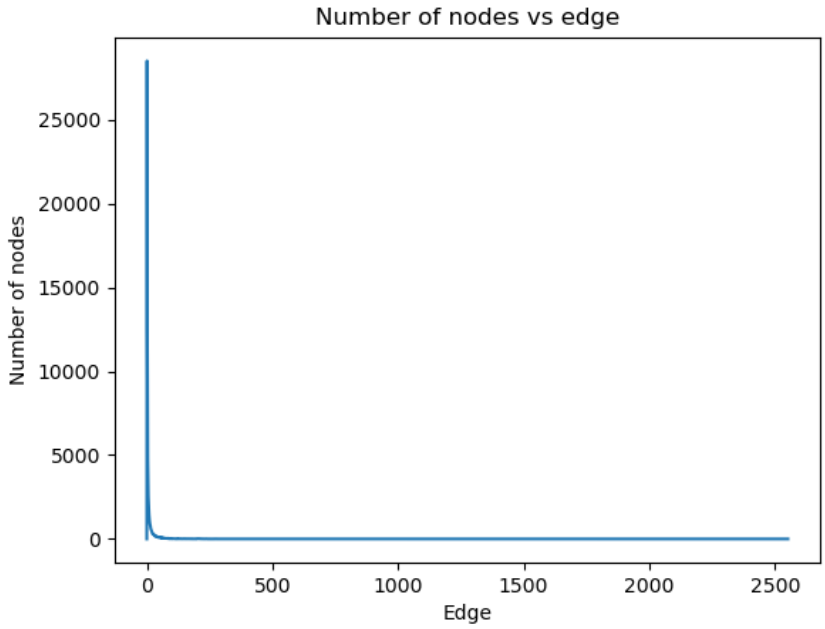
We use function of snap.PrintInfo to produce following result (info.txt):

```
stats:
  Nodes:                    82168
  Edges:                    504230
  Zero Deg Nodes:           0
  Zero InDeg Nodes:         0
  Zero OutDeg Nodes:        0
  NonZero In-Out Deg Nodes: 82168
  Unique directed edges:    1008460
  Unique undirected edges:  504230
  Self Edges:               0
  BiDir Edges:              1008460
  Closed triangles:         602592
  Open triangles:           73175813
  Frac. of closed triads:   0.008168
  Connected component size: 1.000000
  Strong conn. comp. size:  1.000000
  Approx. full diameter:    11
  90% effective diameter:   4.775920
```

The graph is a connected network with connected component size 1, such that complete cascade is must with enough low threshold.

Edge distribution (p0_result.png) (p0_result2.png)(p0_result.txt):






```
Minimun number of edges: 1
Maximun number of edges: 2552
Avergae number of edges: 12.273147697400448
Median number of edges: 2
```

Obviously, most of nodes have edges below than 10. There are a few nodes have extremely high number of edges. Be noticed that, there are total of 82168 nodes, including 28499(34.68%) nodes have only one edge and 12615(15.35%) nods have only two edges. It means that they are seriously affected by a few nodes.

5 Experiment and result

All the below experiment can be implemented again by run “python p?.py”. However, the initial nodes are chosen randomly, so that the result of every time may be different. Each experiment may be run for 15 mins.

Cascading percentage = (total adopters – number of initial adopters)/(total nodes – number of initial adopters) * 100%

Cascading percentage is in range [0%,100%]

5.1 Threshold vs cascading

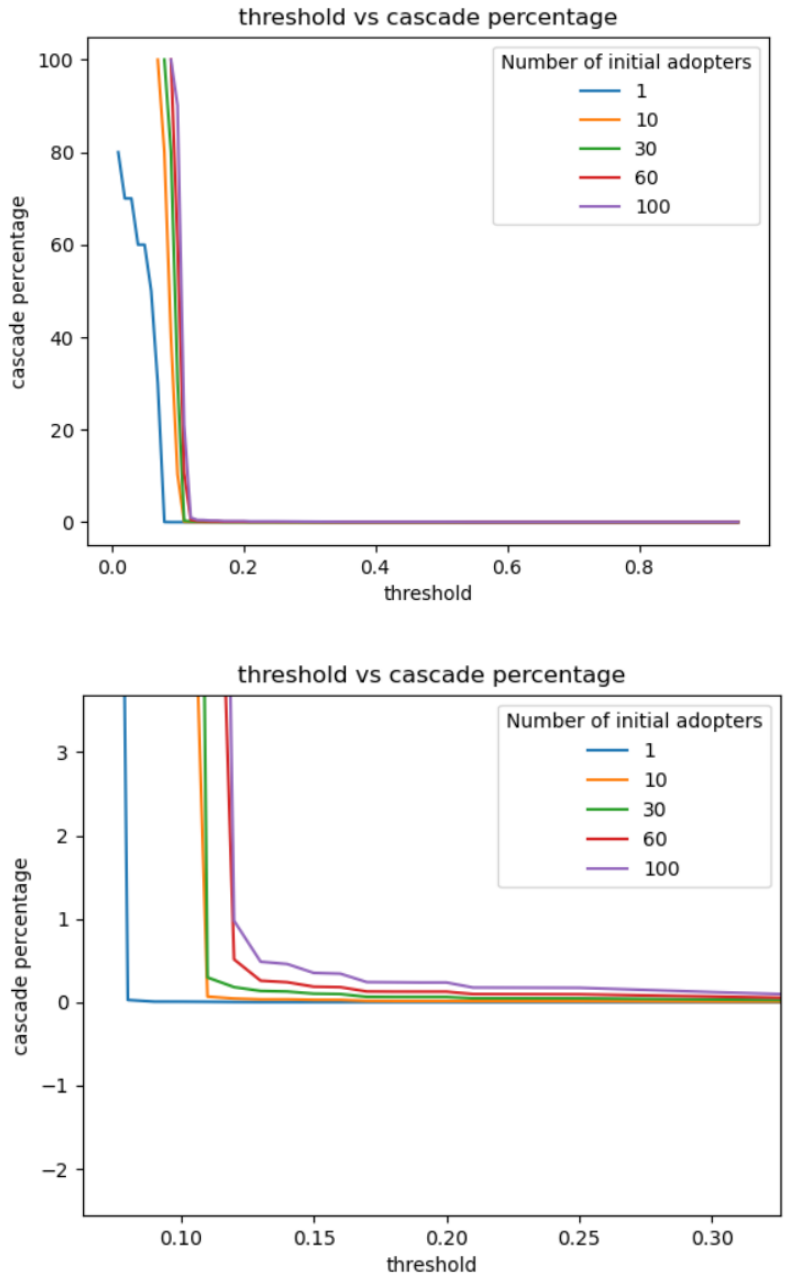
In this experiment, we want to know the relationship between threshold and cascading percentage under different number of initial adopters.

Step 1: For each number of initial adopters (1,10,30,60,100), choose 10 sets of initial adopters.

Step 2: For each number of initial adopters and with decreasing threshold, run cascading and record the average of cascading percentage for 10 sets of initial adopters.

The reason of averaging 10 sets is to decrease the impact of a certain set of initial adopters. For same threshold, same number of initial adopters, different set of initial adopters, the cascading percentage can be very different.

Result (p1_result.png):



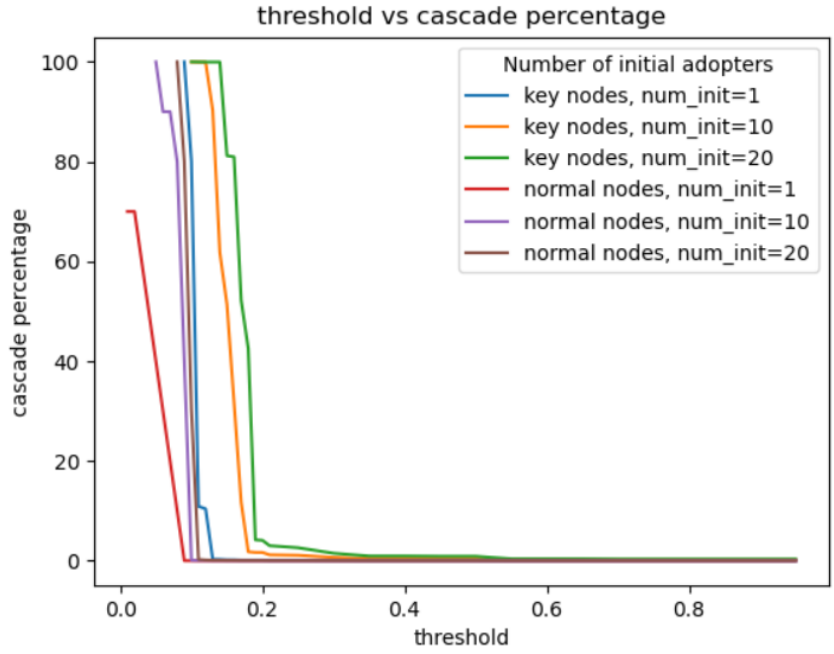
Detail of the value is in p1_result.txt.

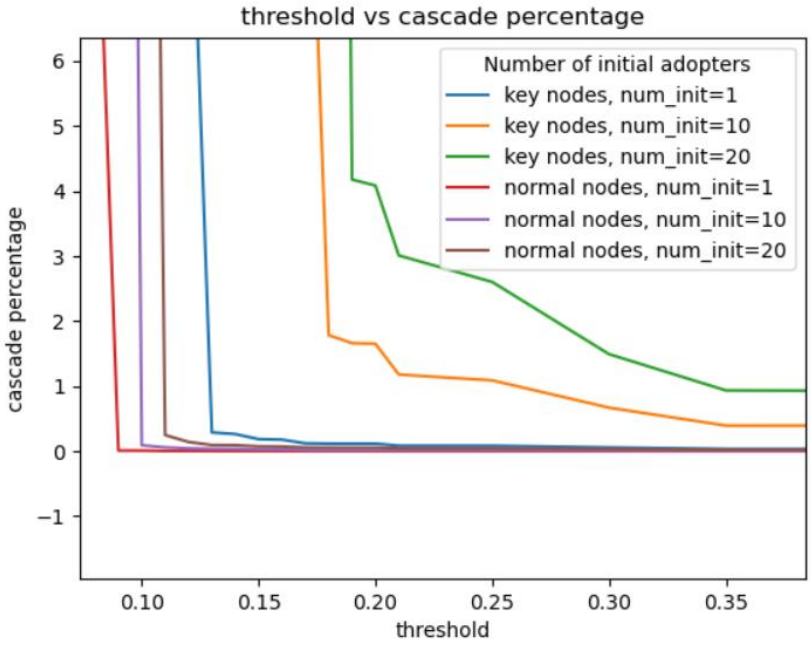
5.2 Adopting key nodes vs cascading

In this experiment, we want to test the difference between randomly choosing initial adopters (normal nodes) and choosing initial adopters with high page-rank value (key nodes) under different number of initial adopters.

Step 1: Calculate the page rank value of all nodes and sort them in descending order.
Step 2: For each number of initial adopters (1,10,20), randomly choose 10 sets key nodes as initial adopters from top 100 page-rank value nodes without replacement. Also, we randomly choose 10 sets of normal nodes from all nodes without replacement.
Step 3: For different threshold value, run the cascading process for the above 30 sets of key nodes and 30 sets of normal nodes. Then average the cascading percentage group by initial number of adopters.

Result(p3_result.png):





Detail of the value is in p3_result.txt.

5.3 Complete cascade vs clustering

In lecture notes, we understood the relationship between complete cascade and clustering. Although this claim “Whenever a set of initial adopters does not cause a complete cascade with threshold q, the remaining network must contain a cluster of density greater than 1 − q” is easily to prove, we want to verify this claim again by real experiment.

Step 1: Calculate the page rank value of all nodes and sort them in descending order.
Step 2: Randomly select 5 nodes from top 100 page-rank value nodes as initial adopters and implement cascading with threshold 0.2. Likely, it cannot lead to a complete cascade.
Step 3: We partition the remaining network into clusters by local bridge and calculate the cluster density of each cluster.
Step 4: Search for a cluster with density higher than 0.8 (1-0.2).

Remark: cluster containing only one node will not print into result file

Result(p4_result.txt):

```
cluster density: 0.6666666666666666
cluster contains 3 nodes
cluster density: 0.5
cluster contains 3 nodes
cluster density: 0.3333333333333333
cluster contains 9 nodes
cluster density: 0.8888888888888888
cluster contains 3 nodes
cluster density: 0.25
```

```
current threshold: 0.2   current cascade percentage: 0.9310760317904653
770 Nodes take action B
cluster contains 25816 nodes
cluster density: 0.10526315789473684
cluster contains 3 nodes
cluster density: 0.25
cluster contains 3 nodes
cluster density: 0.2
cluster contains 3 nodes
cluster density: 0.4
cluster contains 3 nodes
cluster density: 0.25
cluster contains 3 nodes
cluster density: 0.4
cluster contains 8 nodes
cluster density: 0.25
cluster contains 3 nodes
cluster density: 0.3333333333333333
cluster contains 3 nodes
cluster density: 0.2857142857142857
cluster contains 3 nodes
cluster density: 0.4
cluster contains 4 nodes
cluster density: 0.125
cluster contains 3 nodes
cluster density: 0.4
```

6 Discussion

6.1 Threshold vs cascading

From 5.1 result, we can see a few characteristics.

1. With decreasing threshold, cascading percentage must remain unchanged or increase. Cascading behavior is always blocked by some local bridge. If the threshold is not low enough to break through those local bridge, cascading percentage is likely to remain unchanged.
2. With higher number of initial adopters, cascading percentage is always higher. Also, complete cascade is easily to done with a higher threshold.
3. There is an interesting point that the cascading percentage rise sharply when the threshold is low enough.

Focusing on point 3, I think there are two main reasons.
In part 4 data statistic, I mention there are 50.03% of nodes have only one or two edges. From 5.3 result, there is a large cluster with 25816 nodes and many small clusters with few nodes.

Firstly, there are many nodes have only one or two edge. When these nodes are chosen as initial adopters, they are hard to break through local bridge to affect large cluster with high cluster coefficient. When the threshold is low enough to break through local bridge and make some nodes in large cluster adopt action B, the adoption spread fast through the large cluster. Those small clusters connected to large cluster will quickly adopt action B too.

Secondly, if those nodes in that large cluster are chosen as initial adopters, they are still hard to spread the action B through the large cluster because the number of

initial adoptions is too small. They may easily affect those small cluster connecting to them, but it cannot rise cascading percentage too much.

To conclude, the cascading percentage rise sharply is because there is only one large cluster containing 25816 nodes (31.42%), and too much cluster containing a few nodes (always lower than 10). The critical point to increase the cascading percentage is whether the threshold is low enough to spread through that large cluster.

6.2 Adopting key nodes vs cascading

From 5.2 result, we can see that the cascading percentage of adopting key nodes as initial adopters is far higher than one of adopting normal nodes as initial adopters. Even adopting one key node is much better than adopting 20 normal nodes. To understand why key node is easy to spread the adoption, we need to understand how page rank value is calculated. In undirected graph, page rank value is statistically close to the degree distribution. (from wiki)

PageRank of an undirected graph [edit]

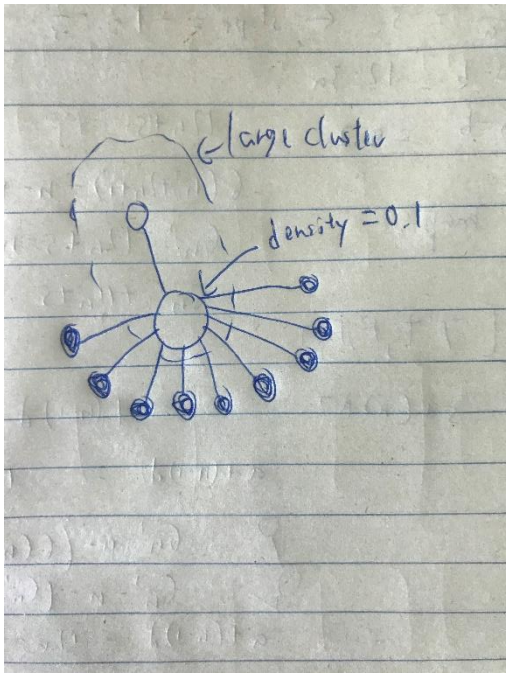
The PageRank of an undirected graph G is statistically close to the degree distribution of the graph G,[36] but they are generally not identical: If R is the PageRank vector defined above, and D is the degree distribution vector

Those high page-rank value nodes are in the large cluster and they have higher probability to connected with each other. These nodes have higher probability of having common friends which are easily to adopt action B. Also, key nodes have many edges where some of them are local bridge connected to small clusters, so that they are easily to make small cluster with low density adopt action B.

6.3 Complete cascade vs clustering

In lecture notes p.33, there is a claim “Whenever a set of initial adopters does not cause a complete cascade with threshold q, the remaining network must contain a cluster of density greater than 1 − q”. In experiment 5.2, we set threshold as 0.2, we can see there is really a cluster with density 0.8889 which is higher than 0.8 (1-0.2).

Although the large cluster have with density 0.105, it is still hard to spread through the large cluster. From the below graph, we can see it is possible to spread through the local bridge if the threshold is 0.9. The large cluster has too many local bridges. Although one of the local bridges may be weak to pass through, it has a small probability that the initial adopters are in clusters connected to this bridge. Moreover, if the below 9 black nodes are not in same cluster, we need all initial nodes are in that 9 clusters to pass through this local bridge.



In lecture notes, we learn it is impossible to spread into cluster which has a density higher than 1-q. In this experiment I understand it may be hard to spread into large cluster even though it is possible with current threshold.

7 Conclusion

After the 3 experiment, we understand the relation between threshold and cascading are highly depends on network structure. Also, we have known how threshold, number of initial adopters and method of selecting adopters affect the cascading result.

8 Reference

SNAP: Stanford network analysis platform
https://snap.stanford.edu/snappy/index.html#download
matplotlib
https://matplotlib.org/