Network Software Modelling

Yifei Zhang

The real-world phenomenon

In the real world, we humans interact with each other at various fronts, in person, through a phone call, text messages, social media, emails, and many other ways. Through the impact of the conversations, everybody accepts influence from their friends in the network, to a degree. Let us also make an assumption that the information that our agents will be exchanging is about a single real number between 0 and 1.

Based on the influence rule above, we can make a further investigation, about how individual influence other people in a friend circle network, and what potential impact factors in a social network can decide individual's influential ability? The influencer always has a solid attitude and other's attitude influenced by the people they know.

Graph model corresponds to the real-world phenomenon.

We will be using undirected graph for modelling with edge weights equal for everyone. We consider this because of the assumption that all agents, here people hold similar relationship with each other. Here we are describing a network of people, a social network.

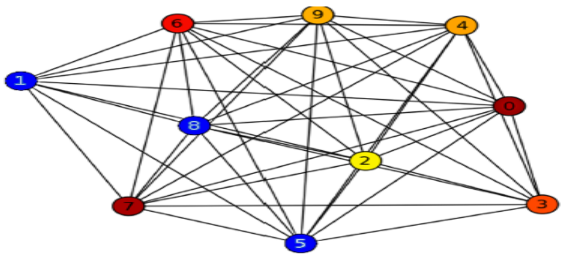
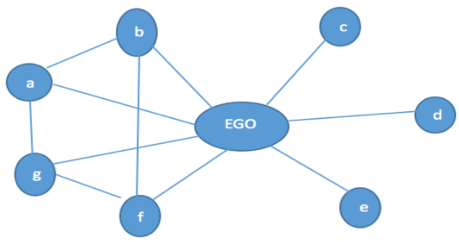


Image 1



Ego network

Consider image 1, it denotes a social network of a person in real world. From the above graph we can see that, a line between two people show that they are connected.

The line denotes "know each other" relation between the connected nodes, the relationship is two ways, meaning both know each other. We signify an edge connecting node ***'u'*** *and node* ***'v'*** by pair(u,v). Since the 'know each other' relationship goes both ways, (u,v) is similar to (v,u), which is the case of undirected graph. Hence, this is undirected graph.

Also as a person cannot be friends with self, we do not have self-loops in the graph.

We have considered a classic application of undirected graph, ego network. Ego network comprises of a central node called (EGO) and the nodes which are directly connected to it, these are called alter and the ties among the alters. Furthermore, each alter, has their own ego network. As result, there is an interlock which forms this social network. Hence, these individual ego networks are sub-graph of the social network graph.

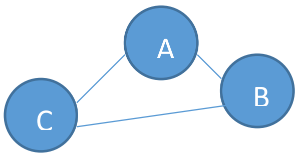
Simulation rules

Initial state:
Normal node:
I_state: the initial attitude for each person that generated by the uniform distribution (0,1)
A_state: the accepted attitude which set to I_state at the initial state, the state changes at each time step.
alpha factor: the person's gullibility factor alpha, set to 0.8 by default.
Influencer node:
I_state: the initial attitude for each person that generated by the uniform distribution (0,1)
A_state: the accepted attitude which set to 1 at the initial state, the state does not change at each time step.
alpha factor: the person's gullibility factor alpha, set to 0.8 by default.

How agents change state:
The changed state is accepted state, which is calculated by the formula below:

```
#all nodes in the list of neighbors are equally weighted, including self
w=1/float((len(neighbors)+1))
s=w*self.a
for node in neighbors:
    s+=w*node.a

# update my beliefs = initial belief plus sum of all influences
self.a=(1-self.alpha)*self.i + self.alpha*s
```



Example:
Calculate the A_state at Time Step = 1for Node B.

Time Step = 0	Node A: Influencer	Node B	Node C
I_state	0.3	0.4	0.8
A_state	1	0.4	0.8
alpha	0.8	0.8	0.8

Node B:

	Time Step = 0	Time Step = 1
I_state	0.4	0.4
A_state	0.4	0.6608

Wb = 1/3
S = 1/3*1 + 1/3*0.8 + 1/3*0.4 = 0.726
A_state = (1 – alpha)*I_sate + alpha*S = (1-0.8)*0.4 + 0.8*0.726 = 0.6608

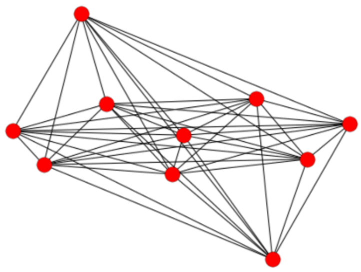
Simulation properties and graph properties

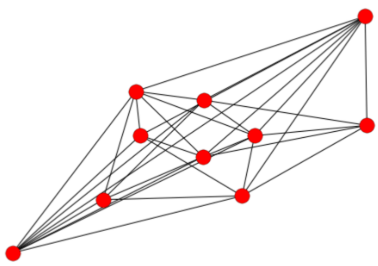
1. The changes in a person's influential ability as the level of **centrality** of different nodes changes.

   Influential ability: as we set our influencer's accept attitude to 1, so the influential ability should be measured by the proportion of the attitude greater than 0.5. and the changes in the proportion shows the influencer's influential ability. For example, if the influencer is the most influential person in the network, other's attitude is more like to be influenced by this influencer – the proportion increase over the time step.
   Centrality: the degree of the node, how many connections this person has in the friend's circle.

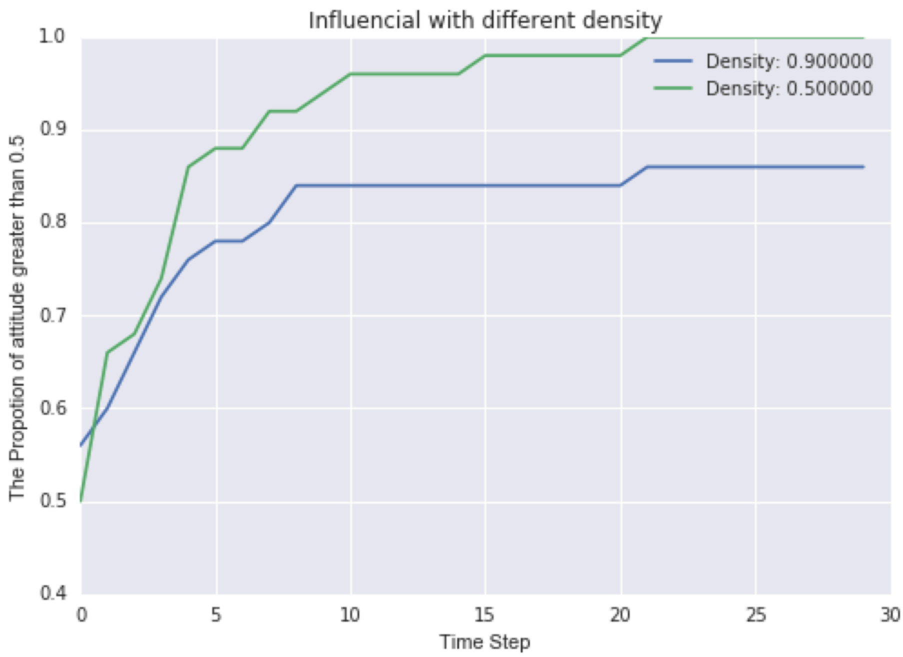
   The simulation result can be found in the Facebook circle real life graph: the node having higher centrality that has higher influence in the network.

2. For the person who has the most connections in the network, we try to simulate influential ability changes over the change in the **density** of the network. In the real world example: the one who has the most connections has a better influence in his friend's circle if the network is less well connected.





   the two graphs above show the different density (left: 0.9 and right: 0.5)



The graph above generated by the python file:
NSM_Assignment2_Density_Graph.py
This meets our common sense that in the less density network, the one who connects with most of people is easier to influence others opinion, since the rest people in the group are less influenced by others compared to the one who has most connection – they are more likely to influenced by the influencer.

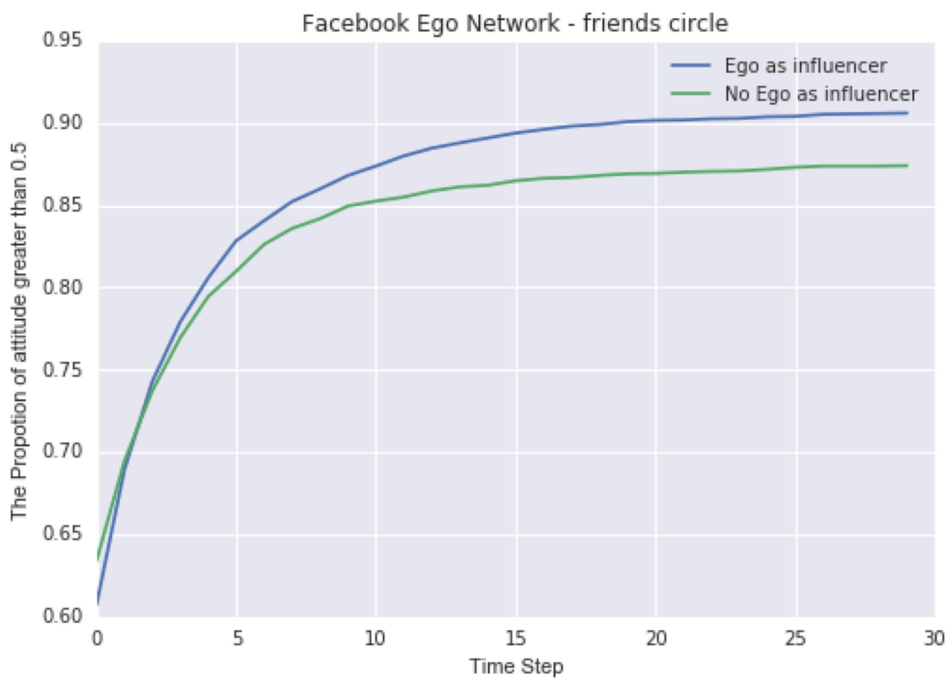
Properties measurement based on the experiments in real world graph and scalable graph

Real life graph: Facebook ego-network
Source: https://snap.stanford.edu/data/egonets-Facebook.html
Python file: *NSM_Assignment2_Real_life_Graph.py*

Graph statistics: nodes: 4039, edges: 88234

The setting for two lines in the graph:
Ego as influencer: ego node's accept attitude = 1 and can not be influencer by other people in the network – fixed during the simulation.
No ego as influencer: a not ego node's accept attitude = 1 and can not be influencer by other people in the network – fixed during the simulation.

The question: in the ego's friends circle, what is the difference of the ego and no ego's influential ability. The answer is obvious: the ego is the most influential person in this ego's network. And the result shows below it meets our expectation – the ego influencer influences more people than a no ego influencer in the ego's network.



Scalable graph: barabasi_albert_graph(n, m)
Python file: *NSM_Assignment2_Scalable_Graph.py*

Parameter setting for barabasi_albert_graph:
N = the number pf nodes
M = number of edges to attach from a new node to existing node

In our case we set the N = 5*M for M from 10 to 190, then N from 50 to 950. In this way, the density of the network does not change. In the real world, N means the number of friends that we could have on the Facebook or other social media.

We generate the sub-graph – an ego graph that the ego is the node has the most degree in the network. And set this ego node as our influencer – the accepted attitude is 1 and can not be influenced by other people in the network.

The question we want to discuss is: how does the ego node's influential ability changes as the network scale with the same density. In another words, as my Facebook friends number grows, how does I influence the friends in my network.

There are two changes we need to focus in the graph: First, how much proportion of attitude (>0.5) changes over the time step for each n. Second, how long it takes for each line to reach the steady state.

Following the two points above, the first finding is: the proportion of attitude changes decreases as the n increase. In the real world meaning is that even our social network growth but the influential of individual "I", the ego, does not relate with the number of my friends, there is certain limitation of ego's influential ability.

The second finding: as n increase, it takes longer time to reach the steady state, which means as the number of friends growth, it takes longer time to spread the ego's influence to the whole network/friends circle.