

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**Homophily Evolutionary model using networkX**

A Report

*submitted by*

**VISHAKHA KUMARESAN 19BCE2678**

**AKSHITA KHALORIA 19BCE2305**

**VASHISHTH GAJJAR 19BCE2286**

**NAVDEEP SUREKA 19BCE2679**

*CSE3021 – Social and Information Networks – J Component*

*E1 + E2 Slot*

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| 1 | **Introduction** | 3 |
| 2 | **Literature Survey** | 4 |
| 3 | **Code & Implementation** | 6 |
| 4 | **Results & Discussion** | 10 |
| 5 | **Conclusion** | 11 |
| 6 | **Future work** | 11 |
| 7 | **References** | 11 |

1. **INTRODUCTION**

Homophily is a well-established phenomenon that has been observed to occur frequently in social networks, where users with similar contexts have a nature of connecting with one another constantly. The study of homophily can provide eminent insights into the flow of information and behaviors within a society and this has been extremely useful in analyzing the formations of online communities. The main driving forces for initiating these networks are social influence and homophily Two main concepts of Homophily include Selection and Social Influence.

Selection:  
It is the tendency in which people make friends with similar interests i.e people select other people having similar habits or interests. In selection, people select similar nodes and connects with them.

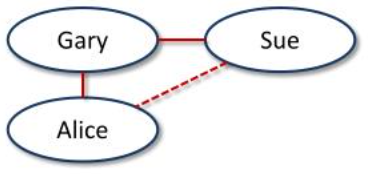
Social Influence:  
It is the tendency in which people change their attitude or behavior to meet the social environment by getting influenced by other people is called Social Influence.Social Influence makes connected nodes similar.

The 3 properties which are going to help us determine the connections to be made/updated are:

* Triadic Closure
* Foci Closure
* Membership Closure

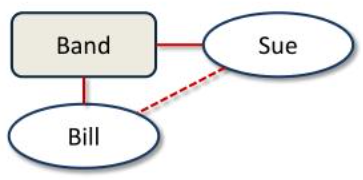
Triadic Closure:

Triadic closure is the property among three nodes A, B, and C (representing people, for instance), that if the connections A-B and A-C exist, there is a tendency for the new connection B-C to be formed.



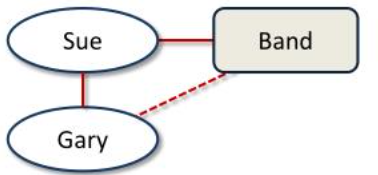
Foci Closure:

If B and C represent people, but A represents a focus, then this is something different: it is the tendency of two people to form a link when they have a focus in common. This is foci closure.



Membership Closure:

If A and B are people, and C is a focus, then we have the formation of a new affiliation: B takes part in a focus that her friend A is already involved in. This is membership closure.



1. **LITERATURE SURVEY**

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Paper Title | Author | Key Points |
| 1 | Birds of a Feather: Homophily in Social Networks | Miller McPherson, Lynn Smith-Lovin and James M. Cook | Principle—the homophily principle—structures network ties of every type, including marriage, friendship, work, advice, support, information transfer, exchange, comembership, and other types of relationship. The result is that people’s personal networks are homogeneous with regard to many sociodemographic, behavioral, and intrapersonal characteristics. |
| 2 | Purity homophily in social networks. | Morteza Dehghani, Kate Johnson, Joe Hoover, Justin Garten, Niki Jitendra Parmar, Rumen Iliev, Eyal Sagi, Stephen Vaisey, Jesse Graham | Analysis of a corpus of over 700,000 tweets revealed that the distance between 2 people in a social-network can be predicted based on differences in the moral purity content— but not other moral content— of their messages |
| 3 | A community detection algorithm based on triadic closure and membership closure. | XU Yunfeng; ZHAO Ning; HAO Xuejun; LI Bing; LIU Huijuan | While using this algorithm, the number of Triadic closure, the probability of vertex belonged to a community and the difference in expansion degree are the keys to value vertices in complex network. This method combines the characteristics of the global importance of vertices. Namely, in complex networks, the greater the number of Triadic closures is, the greater the likelihood of them in a community will be. The greater the vertex membership closures are, the priorer the vertex will be divided. The difference in expansion degree is to determine whether the i community is divided completely or not. |
| 4 | A simple model of homophily in social networks | Sergio Currarini. Jesse Matheson, Fernando Vega Redondo | Threshold equilibrium generates patterns of in-group and cross-group ties that are consistent with empirical evidence of homophily in two paradigmatic instances: high school friendships and interethnic marriages. |
| 5 | Origins of Homophily in an Evolving Social Network | Gueorgi Kossinets and Duncan J. Watts | The analysis indicates that highly similar pairs do show greater than average propensity to form new ties; however, it also finds that tie formation is heavily biased by triadic closure and focal closure, which effectively constrain the opportunities among which individuals may select. |

1. **CODE & IMPLEMENTATION**

import networkx as nx

import matplotlib.pyplot as plt

import random

import math

import time

def create\_graph():

G=nx.Graph()

G.add\_nodes\_from(range(1,101))

return G

plt.show()

nx.draw(G)

def assign\_bmi(G):

for each in G.nodes():

G.nodes[each]['name']=random.randint(15,40)

G.nodes[each]['type']='person'

def visualize(G,t):

time.sleep(1)

labeldict=get\_labels(G)

nodesize=get\_size(G)

color=get\_colors(G)

nx.draw(G,labels=labeldict,node\_size=nodesize,node\_color=color)

plt.savefig('evolution.jpg')

plt.clf()

plt.cla()

nx.write\_gml(G,'evolution\_'+str(t)+'.gml')

#plt.show()

def get\_labels(G):

dict1={}

for each in G.nodes():

dict1[each]=G.nodes[each]['name']

#print(dict1[each])

#print("labels")

#print(dict1)

return dict1

def get\_size(G):

array1=[]

for each in G.nodes():

if(G.nodes[each]['type']=='person'):

array1.append(G.nodes[each]['name']\*10)

else:

array1.append(500)

return array1

def add\_foci\_nodes(G):

n=G.number\_of\_nodes()

i=n+1

foci\_nodes=['gym','eatout','movie\_club','karate\_club','yoga\_club']

for j in range(5):

G.add\_node(i)

G.nodes[i]['name']=foci\_nodes[j]

G.nodes[i]['type']='foci'

i+=1

def get\_colors(G):

c=[]

for i in G.nodes():

if(G.nodes[i]['type']=='person'):

if(G.nodes[i]['name']==15):

c.append('yellow')

elif(G.nodes[i]['name']==40):

c.append('green')

else:

c.append('blue')

else:

c.append('red')

return c

def get\_person\_nodes(G):

p=[]

for i in G.nodes():

if(G.nodes[i]['type']=='person'):

p.append(i)

return p

def get\_foci\_nodes(G):

f=[]

for i in G.nodes():

if(G.nodes[i]['type']=='foci'):

f.append(i)

return f

def add\_foci\_edges(G):

foci\_nodes=get\_foci\_nodes(G)

person\_nodes=get\_person\_nodes(G)

print("foci nodes")

print(foci\_nodes)

print("person nodes")

print(person\_nodes)

for i in person\_nodes:

r=random.choice(foci\_nodes)

G.add\_edge(i,r)

def homophily(G):

pnodes=get\_person\_nodes(G)

for u in pnodes:

for v in pnodes:

if(u!=v):

diff=abs(G.nodes[u]['name']-G.nodes[v]['name'])

p=1/(diff+1000)

r=random.uniform(0,1)

if(r<p):

G.add\_edge(u,v)

#common neighbours

def cmn(u,v,G):

nu=set(G.neighbors(u))

nv=set(G.neighbors(v))

return(len(nu & nv))

def closure(G):

array1=[]

for u in G.nodes():

for v in G.nodes():

if(u!=v and G.nodes[u]['type']=='person' or G.nodes[v]['type']=='person' ):

k=cmn(u,v,G)

p=1-math.pow(1-0.01,k)

tmp=[]

tmp.append(u)

tmp.append(v)

tmp.append(p)

array1.append(tmp)

#print(array1)

for i in array1:

u=i[0]

v=i[1]

p=i[2]

r=random.uniform(0,1)

if r<p:

G.add\_edge(u,v)

def change\_bmi(G):

fnodes=get\_foci\_nodes(G)

for i in fnodes:

if(G.nodes[i]['name']=='eatout'):

for j in G.neighbors(i):

if(G.nodes[j]['name']!=40):

G.nodes[j]['name']=G.nodes[j]['name'] + 1

if(G.nodes[i]['name']=='gym'):

for j in G.neighbors(i):

if(G.nodes[j]['name']!=15):

G.nodes[j]['name']=G.nodes[j]['name'] – 1

G = create\_graph()

assign\_bmi(G)

add\_foci\_nodes(G)#to add interest field in the nodes

add\_foci\_edges(G)

time.sleep(10)

visualize(G,0)

#nx.write\_gml(G,'evolution\_0.gml')

for t in range(1,10):

homophily(G)

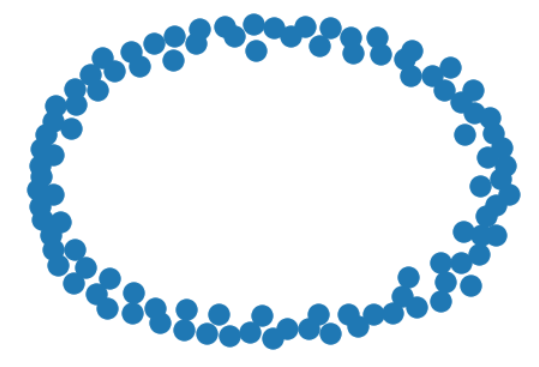
closure(G)

change\_bmi(G)

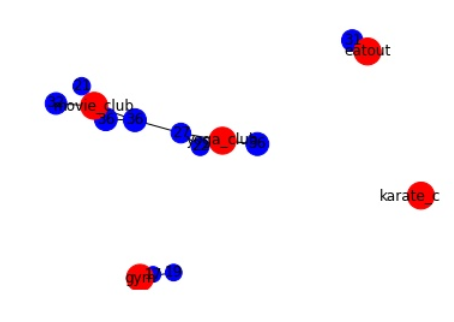
visualize(G,t+1)

1. **RESULTS & DISCUSSION**

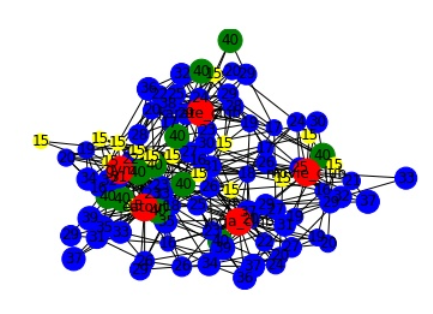
Graph before connecting the edges and adding the foci nodes:



For 10 nodes:



For 100 nodes:



1. **CONCLUSION**

The model we created is a simple depiction of how homophily is represented in real life scenarios. The result shows the random edges which were added depending on the probability that they will undergo either one of the 3 closures mentioned.

This randomness depended on how they were connected after adding them into the network.

The final network graph shows the visualization of the focal nodes and person nodes were connected after undergoing the factors of selection and social influence.

1. **FUTURE WORK**

The dataset we have used for the present implementation is a randomly generated value dataset. In the future we would like to test the model with some already existing real-world values.

We would also like to potentially see the deployment and storage aspect of the Homophily model.

1. **REFERENCES**

* [**https://www.sciencedirect.com/science/article/abs/pii/S0014292116300642**](https://www.sciencedirect.com/science/article/abs/pii/S0014292116300642)
* [**https://www.jstor.org/stable/2678628**](https://www.jstor.org/stable/2678628)
* [**https://ieeexplore.ieee.org/abstract/document/5616320**](https://ieeexplore.ieee.org/abstract/document/5616320)
* [**https://psycnet.apa.org/record/2015-58947-001**](https://psycnet.apa.org/record/2015-58947-001)
* [**https://www.sciencedirect.com/science/article/abs/pii/S0047272716301001**](https://www.sciencedirect.com/science/article/abs/pii/S0047272716301001)
* [**https://journals.sagepub.com/doi/abs/10.1177/001440299606200504**](https://journals.sagepub.com/doi/abs/10.1177/001440299606200504)