**“Kuch bhi”**

**PROJECT REPORT**

Social and Information Networks (CSE3021)

By

**Tajinder Singh Sondhi (15BCE0040)**

**Shivam Ghildiyal(15BCE0216)**

**Slot: A2+TA2**

**Name of faculty**: **Prof. Vasantha W.B.**

**(SCHOOL OF COMPUTER SCIENCE AND ENGINEERING)**

WINTER SEMESTER 2016-17



**CERTIFICATE**

This is to certify that the project work entitled “***Kuch bhi***” that is being submitted by “***Tajinder Singh Sondhi, and Shivam Ghildiyal***” for Social and Information Networks (CSE3021) is a record of bonafide work done under my supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other CAL course.

Place: Vellore

Date: 03/11/2017

**Signature of Students**:

Tajinder Singh Sondhi (15BCE0040)

Shivam Ghildiyal (15BCE0216)

**Signature of Faculty: Vasantha W.B.**

**Abstract**

All sociologists agree to the fact that power is a fundamental property of social structures. There is much less agreement about what power is, and how one can describe and analyse its causes and consequences. The global scene as of today is more fluid than ever before in human history. All these changes have a profound impact on foreign relations in the decades to come and makes international relations inspiring, intriguing and important. In this project, we analyse the relational network of top 20 countries of the world; which is a list of countries ordered by Military Strength Index based on the Credit Suisse report in September 2015. Our analysis will be focused on shifting the power centre of the top militaries of the world to India.

*Keywords* – Military Strength Index, power centre, betweenness centrality, closeness centrality, eigen value centrality.

**1. Introduction**

Thinking from a network aspect has contributed several important insights about social power. Most importantly, the network approach emphasizes that power is inherently relational. An individual does not have power in the abstract, they have power because they can dominate others (ego's power is alter's dependence). Because power is a consequence of patterns of relations, the amount of power in social structures can vary to a wide extent. If a system is very loosely coupled (low density) not much power can be exerted whereas in high density systems there is the potential for greater power. The amount of power in a system and its distribution across actors are related, but are not the same thing. Two systems can have the same amount of power, but it can be equally distributed in one and unequally distributed in another. Thus, power is both a systemic (macro) and relational (micro) property. Power in social networks may be viewed either as a micro property (i.e. it describes relations between actors) or as a macro property (i.e. one that describes the entire population). Just as with other key sociological concepts, the macro and micro are closely connected in social network thinking.

Network analysts often describe the way that an actor is embedded in a relational network as imposing constraints on the actor, and offering the actor opportunities. Actors that face fewer constraints, and have more opportunities than others are in favourable structural positions. Having a favoured position means that an actor may extract better bargains in exchanges, have greater influence, and that the actor will be a focus for deference and attention from those in less favoured positions. In our project we are trying to change relational ties for India in such a way that the node of India in our graph would be in a favourable structural position after making ties with some non-adjacent nodes.

**2. Literature Review Summary Table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Authors and Year (Reference)* | *Title (Study)* | *Concept / Theoretical model/ Framework* | *Methodology used/ Implementation* | *Dataset details/ Analysis* | *Relevant Finding* | *Limitations/ Future Research/ Gaps identified* |
|  |  |  |  |  |  |  |

**3. Objective of the project**

The aim of our project is to shift the power centre of the top militaries of the world to India. We propose that which country would be best for India to form military alliance with in order to attain maximum average gain in its betweenness centrality, closeness centrality and eigen vector centrality.

**4. Innovative component in the project**

The experimental component of our project is to propose the best possible way to attain maximum power (centrality) amongst the top 20 militaries of the world. The methodology adopted by us, which is also discussed later in this report, is greedy in nature. Instead of comparing relations with all possible countries in order to shift the power centre to India, we have proposed to select a locally optimum choice at each with the hope of finding a global optimum. This methodology has significantly reduced the time complexity of our approach from O(n!) to O(n2) by greedy approach.

**5. Work done and implementation**

**5.1. Methodology adapted:**

In our methodology, we have implemented greedy algorithm, which is an algorithmic paradigm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. We find the best possible country amongst the possible country, with which if an alliance is formed, will result in maximum increase in India’s power (centrality). The gain in India’s centrality is calculated as an average of its betweenness centrality, closeness centrality, and eigen vector centrality; and that country is then assumed to be an alliance of India. This process is repeated till India becomes the power centre of the world based upon any one of the parameters: betweenness centrality, closeness centrality, and eigen vector centrality.

**5.2. Dataset used:**

The dataset has been collected and narrowed down as per a Full Network Research Design (FNRD). In FNRD, we begin with a set of nodes and then measure all of the ties of a given type among those nodes.

1. The main source of our data our two articles – a report by Credit Sussie, which is a list of countries by Military Strength Index based on the Credit Suisse report in September 2015 [ https://en.wikipedia.org/wiki/List\_of\_countries\_by\_Military\_Strength\_Index]; and the list of current security alliances of the world. We made a sociomatrix where these 20 countries are the nodes and their relation would exist if there exists any military alliance between them. [https://en.wikipedia.org/wiki/List\_of\_military\_alliances]

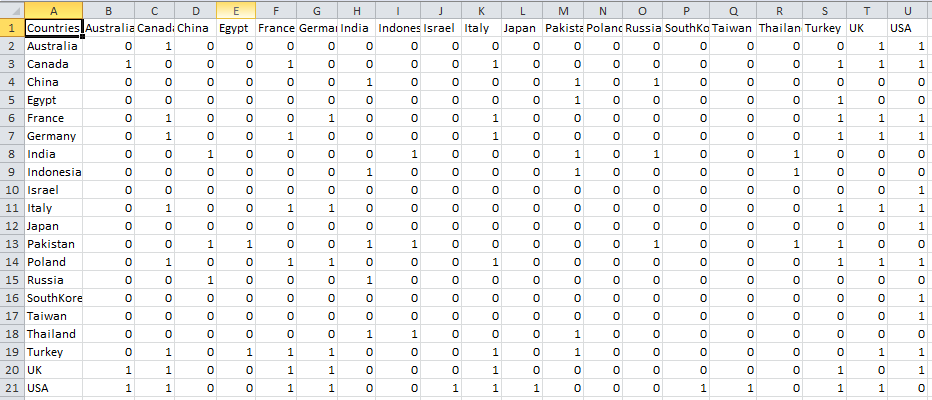


Fig: Sociomatrix

1. Our project is not based on or referred from any other project.
2. Our project is unique from all the previous projects as it suggests the best possible military alliance for India amongst the top 20 militaries of the world.

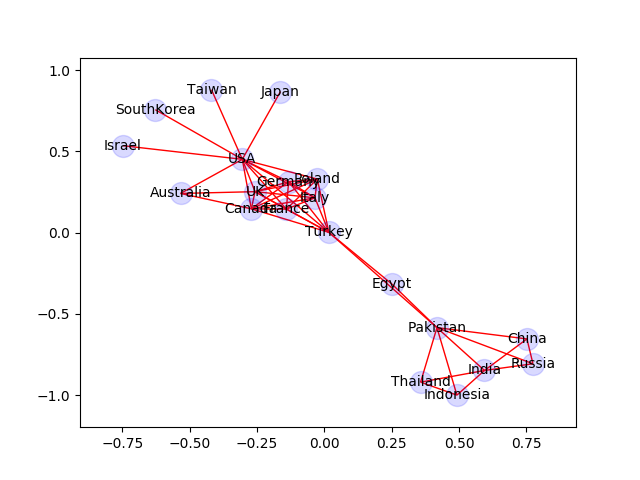
**5.3. Tools used:**

Tools used to implement the methodology:

* Programming language for Data Analysis: Python 3.5
* Pandas – pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools.
* Pylab from Matplotlib – Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.
* Networkx – NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.
* Spreadsheet softwares: LibreCalc, MS-Excel

**5.4. Screenshot and Demo:**

1. Initial Graph



1. Source Code (nx.py)

import networkx as nx

import pandas

import pylab as plt #import Matplotlib plotting interface

def create\_adjlist():

f = open('myfile.adjlist', 'w')

for i in range(0,20):

#write row country in same line

row = df.index[i]

f.write(row)

for j in range(0,20):

if df.iloc[i,j] == 1:

#write column country in same line

col = df.columns[j]

f.write(" "+col)

#new line

if(i != 19):

f.write("\n")

f.close()

def draw\_graph\_G(name):

graph\_pos = nx.spring\_layout(G)

# draw nodes, edges and labels

nx.draw\_networkx\_nodes(G, graph\_pos, node\_size=250 , node\_color='blue', alpha=0.15)

nx.draw\_networkx\_edges(G, graph\_pos, edge\_color='red')

nx.draw\_networkx\_labels(G, graph\_pos, font\_size=10, font\_family='sans-serif', font\_color='black')

plt.savefig("graphs/"+name+".png")

plt.clf()

def draw\_graph\_my\_graph(name):

graph\_pos = nx.spring\_layout(my\_graph)

# draw nodes, edges and labels

nx.draw\_networkx\_nodes(my\_graph, graph\_pos, node\_size=250 , node\_color='blue', alpha=0.15)

nx.draw\_networkx\_edges(my\_graph, graph\_pos, edge\_color='red')

nx.draw\_networkx\_labels(my\_graph, graph\_pos, font\_size=10, font\_family='sans-serif', font\_color='black')

plt.savefig("graphs/"+name+".png")

plt.clf()

#show graph

#plt.show()

def calculate\_power():

bet\_cen = nx.betweenness\_centrality(G)

clo\_cen = nx.closeness\_centrality(G)

eig\_cen = nx.eigenvector\_centrality(G)

max\_bet\_cen = highest\_centrality(bet\_cen)

max\_clo\_cen = highest\_centrality(clo\_cen)

max\_eig\_cen = highest\_centrality(eig\_cen)

return (max\_bet\_cen,max\_clo\_cen,max\_eig\_cen)

def highest\_centrality(cent\_dict):

cent\_items=[(b,a) for (a,b) in cent\_dict.items()]

cent\_items.sort()

cent\_items.reverse()

return tuple(reversed(cent\_items[0]))

def link\_prediction():

global india

leng = 20

for col in range(0,leng):

if col == india:

continue

if df.iloc[india,col]==1:

continue

else:

df.iloc[india,col]=1

df.iloc[col,india]=1

find\_new\_powers(india,col)

df.iloc[india,col]=0

df.iloc[col,india]=0

def find\_new\_powers(india, col):

global G,my\_graph,max\_cen,country

create\_adjlist()

fh=open("myfile.adjlist", 'rb')

G=nx.read\_adjlist(fh)

fh.close()

#draw\_graph()

powers = calculate\_power()

#print("\nPowers if India forms a link with: "+df.columns[col])

#print(str(powers[0])+"\n"+str(powers[1])+"\n"+str(powers[2])+"\n")

ind\_cen = indian\_power()

avg\_cen = (ind\_cen[0]+ind\_cen[1]+ind\_cen[2])/3

if avg\_cen >= max\_cen:

max\_cen = avg\_cen

country = col

my\_graph = G

a["new\_max\_centralities"] = powers

a["new\_indian\_centralities"] = ind\_cen

def indian\_power():

bet\_cen = nx.betweenness\_centrality(G)['India']

clo\_cen = nx.closeness\_centrality(G)['India']

eig\_cen = nx.eigenvector\_centrality(G)['India']

#print("New centralities of India:")

#print('bet\_cen: '+str(bet\_cen)+'\t\tchange: '+str(bet\_cen-a[0]))

#print('clo\_cen: '+str(clo\_cen)+'\t\tchange: '+str(clo\_cen-a[1]))

#print('eig\_cen: '+str(eig\_cen)+'\t\tchange: '+str(eig\_cen-a[2]))

return (bet\_cen, clo\_cen, eig\_cen)

def print\_link():

if country == -1:

print("\nNo such country found!")

else:

a["link"] = df.columns[country]

print("\n\tThe country with whom, India would get maximum increase in its centrality is: "+a["link"])

print("\n\tNew Indian centralities:\n\t"+str(a["new\_indian\_centralities"]))

print("\n\tNew\_max\_centralities:\n\t"+str(a["new\_max\_centralities"]))

draw\_graph\_my\_graph(str(relation)+"-"+a["link"])

def is\_india\_power\_center():

for item in a["new\_max\_centralities"]:

if str(item[0]) == 'India':

return True

return False

################################################################################################################

############################## MAIN ###################################################

################################################################################################################

a = {

"original\_max\_centralities":"",

"original\_indian\_centralities":"",

"link":"",

"new\_indian\_cetralities":"",

"new\_max\_centralities":""

}

india = 6

df = pandas.read\_csv("sin\_dataset.csv")

df = df.set\_index("Countries")

create\_adjlist()

fh=open("myfile.adjlist", 'rb')

G=nx.read\_adjlist(fh)

fh.close()

powers = calculate\_power()

a["original\_max\_centralities"] = powers

#print("\nCurrent Powers:\n"+str(powers[0])+"\n"+str(powers[1])+"\n"+str(powers[2])+"\n")

a["original\_indian\_centralities"] = (nx.betweenness\_centrality(G)['India'], nx.closeness\_centrality(G)['India'], nx.eigenvector\_centrality(G)['India'])

print("\nOriginal\_max\_centralities:\n"+str(a["original\_max\_centralities"]))

print("\nOriginal\_indian\_centralities:\n"+str(a["original\_indian\_centralities"]))

draw\_graph\_G("0-initial")

df.to\_csv("stochiomatrices/0-initial.csv")

max\_cen = (a["original\_indian\_centralities"][0]+a["original\_indian\_centralities"][1]+a["original\_indian\_centralities"][2])/3

country = -1

my\_graph = nx.Graph()

relation = 1

while(not is\_india\_power\_center()):

country = -1

print("\nNew Relation: "+str(relation))

link\_prediction()

print\_link()

df.iloc[india,country] = 1

df.iloc[country,india] = 1

df.to\_csv("stochiomatrices/"+str(relation)+"-"+a["link"]+".csv")

relation +=1

print("\n\*\*\*INDIA HAS BECOME THE POWER CENTER\*\*\*")

print("Total new relations (edges) made: "+str(relation-1))

print("The Grpahs have been printed in 'graphs' directory.")

print("\n\*\*\*THANK YOU\*\*\*\n")

def draw\_graph\_G(name):

graph\_pos = nx.spring\_layout(G)

# draw nodes, edges and labels

nx.draw\_networkx\_nodes(G, graph\_pos, node\_size=250 , node\_color='blue', alpha=0.15)

nx.draw\_networkx\_edges(G, graph\_pos, edge\_color='red')

nx.draw\_networkx\_labels(G, graph\_pos, font\_size=10, font\_family='sans-serif', font\_color='black')

plt.savefig("graphs/"+name+".png")

plt.clf()

def draw\_graph\_my\_graph(name):

graph\_pos = nx.spring\_layout(my\_graph)

# draw nodes, edges and labels

nx.draw\_networkx\_nodes(my\_graph, graph\_pos, node\_size=250 , node\_color='blue', alpha=0.15)

nx.draw\_networkx\_edges(my\_graph, graph\_pos, edge\_color='red')

nx.draw\_networkx\_labels(my\_graph, graph\_pos, font\_size=10, font\_family='sans-serif', font\_color='black')

plt.savefig("graphs/"+name+".png")

plt.clf()

#show graph

#plt.show()

def calculate\_power():

bet\_cen = nx.betweenness\_centrality(G)

clo\_cen = nx.closeness\_centrality(G)

eig\_cen = nx.eigenvector\_centrality(G)

max\_bet\_cen = highest\_centrality(bet\_cen)

max\_clo\_cen = highest\_centrality(clo\_cen)

max\_eig\_cen = highest\_centrality(eig\_cen)

return (max\_bet\_cen,max\_clo\_cen,max\_eig\_cen)

def highest\_centrality(cent\_dict):

cent\_items=[(b,a) for (a,b) in cent\_dict.items()]

cent\_items.sort()

cent\_items.reverse()

return tuple(reversed(cent\_items[0]))

def link\_prediction():

global india

leng = 20

for col in range(0,leng):

if col == india:

continue

if df.iloc[india,col]==1:

continue

else:

df.iloc[india,col]=1

df.iloc[col,india]=1

find\_new\_powers(india,col)

df.iloc[india,col]=0

df.iloc[col,india]=0

def find\_new\_powers(india, col):

global G,my\_graph,max\_cen,country

create\_adjlist()

fh=open("myfile.adjlist", 'rb')

G=nx.read\_adjlist(fh)

fh.close()

#draw\_graph()

powers = calculate\_power()

#print("\nPowers if India forms a link with: "+df.columns[col])

#print(str(powers[0])+"\n"+str(powers[1])+"\n"+str(powers[2])+"\n")

ind\_cen = indian\_power()

avg\_cen = (ind\_cen[0]+ind\_cen[1]+ind\_cen[2])/3

if avg\_cen >= max\_cen:

max\_cen = avg\_cen

country = col

my\_graph = G

a["new\_max\_centralities"] = powers

a["new\_indian\_centralities"] = ind\_cen

def indian\_power():

bet\_cen = nx.betweenness\_centrality(G)['India']

clo\_cen = nx.closeness\_centrality(G)['India']

eig\_cen = nx.eigenvector\_centrality(G)['India']

#print("New centralities of India:")

#print('bet\_cen: '+str(bet\_cen)+'\t\tchange: '+str(bet\_cen-a[0]))

#print('clo\_cen: '+str(clo\_cen)+'\t\tchange: '+str(clo\_cen-a[1]))

#print('eig\_cen: '+str(eig\_cen)+'\t\tchange: '+str(eig\_cen-a[2]))

return (bet\_cen, clo\_cen, eig\_cen)

def print\_link():

if country == -1:

print("\nNo such country found!")

else:

a["link"] = df.columns[country]

print("\n\tThe country with whom, India would get maximum increase in its centrality is: "+a["link"])

print("\n\tNew Indian centralities:\n\t"+str(a["new\_indian\_centralities"]))

print("\n\tNew\_max\_centralities:\n\t"+str(a["new\_max\_centralities"]))

draw\_graph\_my\_graph(str(relation)+"-"+a["link"])

def is\_india\_power\_center():

for item in a["new\_max\_centralities"]:

if str(item[0]) == 'India':

return True

return False

################################################################################################################

############################## MAIN ###################################################

################################################################################################################

a = {

"original\_max\_centralities":"",

"original\_indian\_centralities":"",

"link":"",

"new\_indian\_cetralities":"",

"new\_max\_centralities":""

}

india = 6

df = pandas.read\_csv("sin\_dataset.csv")

df = df.set\_index("Countries")

create\_adjlist()

fh=open("myfile.adjlist", 'rb')

G=nx.read\_adjlist(fh)

fh.close()

powers = calculate\_power()

a["original\_max\_centralities"] = powers

#print("\nCurrent Powers:\n"+str(powers[0])+"\n"+str(powers[1])+"\n"+str(powers[2])+"\n")

a["original\_indian\_centralities"] = (nx.betweenness\_centrality(G)['India'], nx.closeness\_centrality(G)['India'], nx.eigenvector\_centrality(G)['India'])

print("\nOriginal\_max\_centralities:\n"+str(a["original\_max\_centralities"]))

print("\nOriginal\_indian\_centralities:\n"+str(a["original\_indian\_centralities"]))

draw\_graph\_G("0-initial")

df.to\_csv("stochiomatrices/0-initial.csv")

max\_cen = (a["original\_indian\_centralities"][0]+a["original\_indian\_centralities"][1]+a["original\_indian\_centralities"][2])/3

country = -1

my\_graph = nx.Graph()

relation = 1

while(not is\_india\_power\_center()):

country = -1

print("\nNew Relation: "+str(relation))

link\_prediction()

print\_link()

df.iloc[india,country] = 1

df.iloc[country,india] = 1

df.to\_csv("stochiomatrices/"+str(relation)+"-"+a["link"]+".csv")

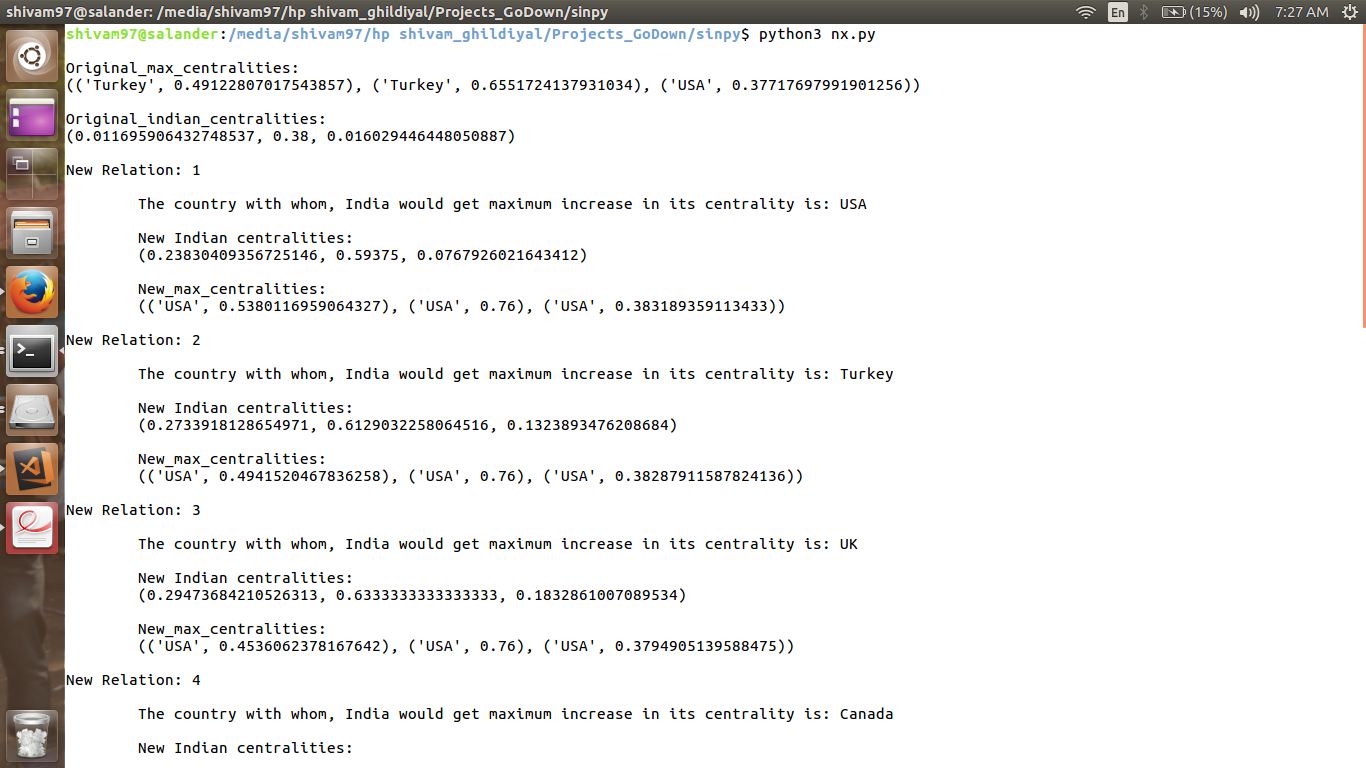
relation +=1

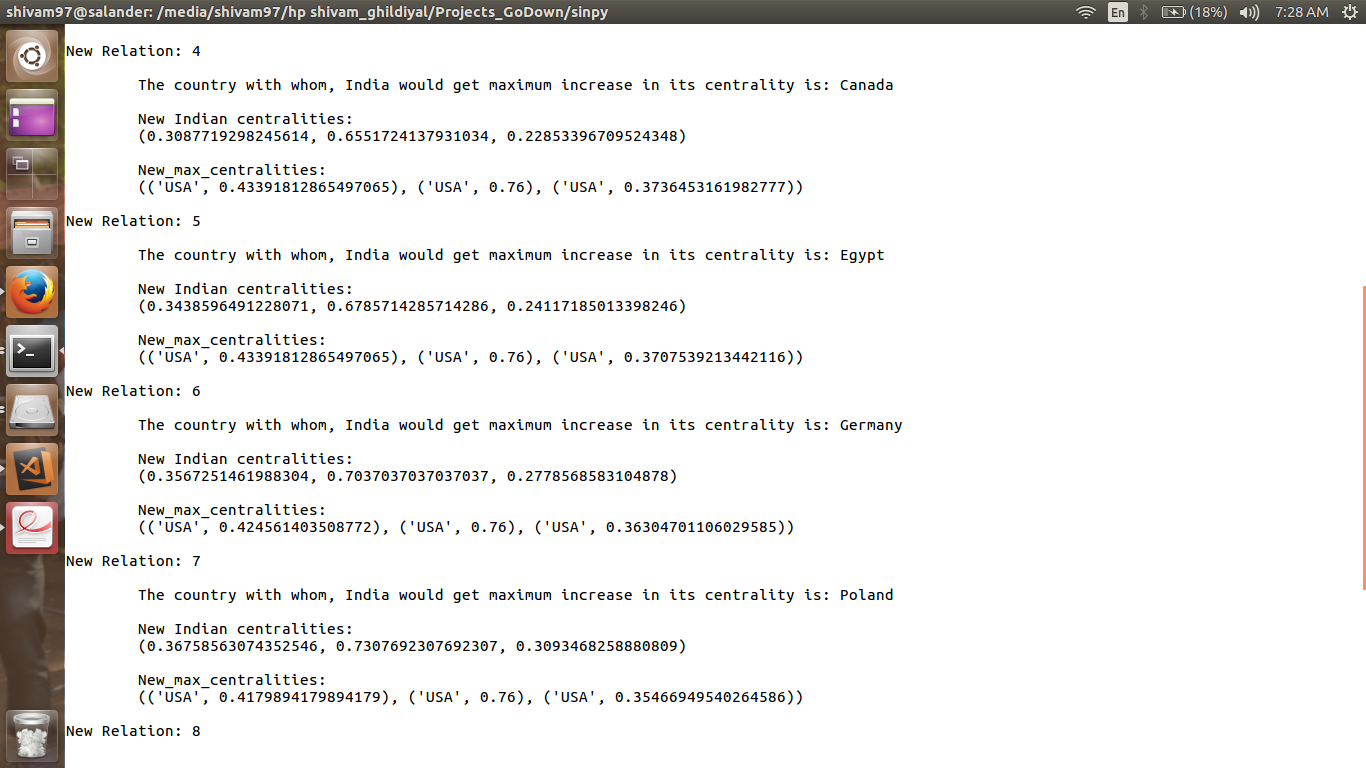
print("\n\*\*\*INDIA HAS BECOME THE POWER CENTER\*\*\*")

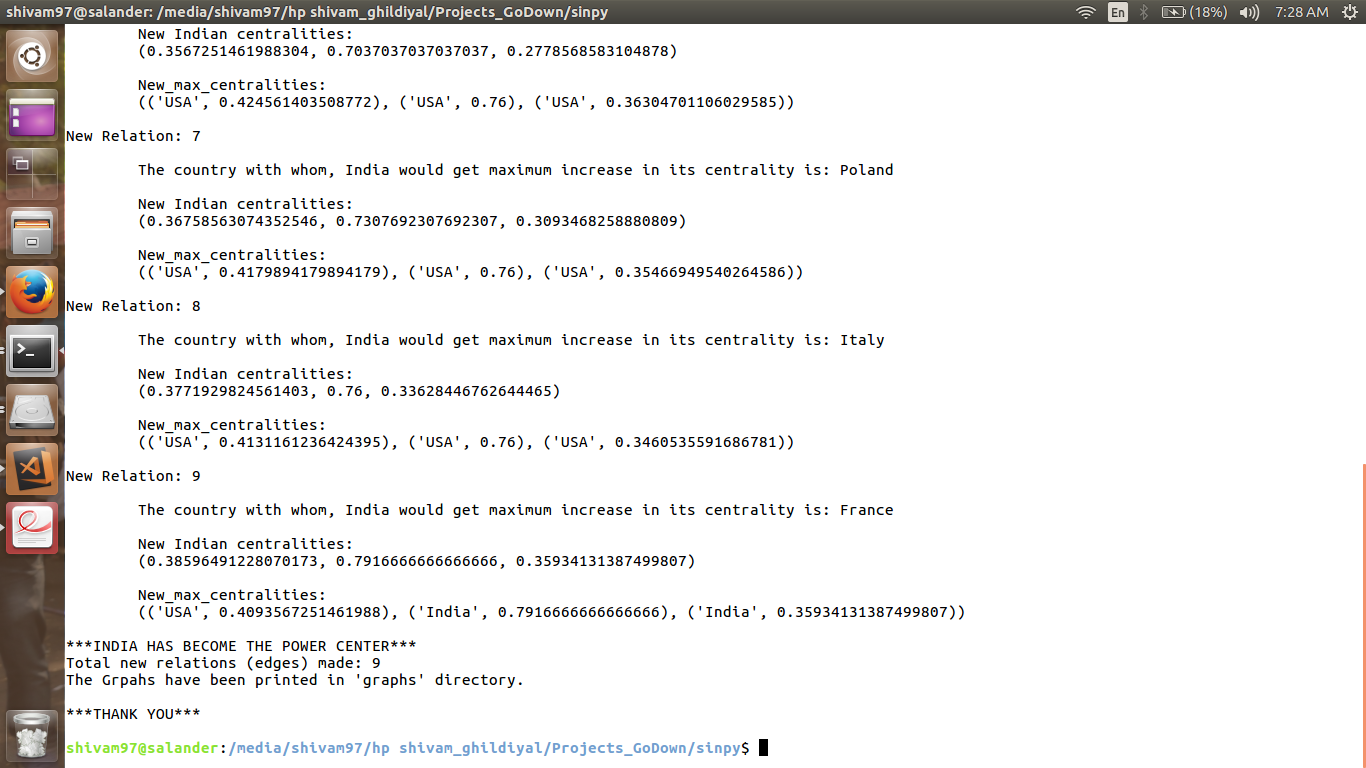
print("Total new relations (edges) made: "+str(relation-1))

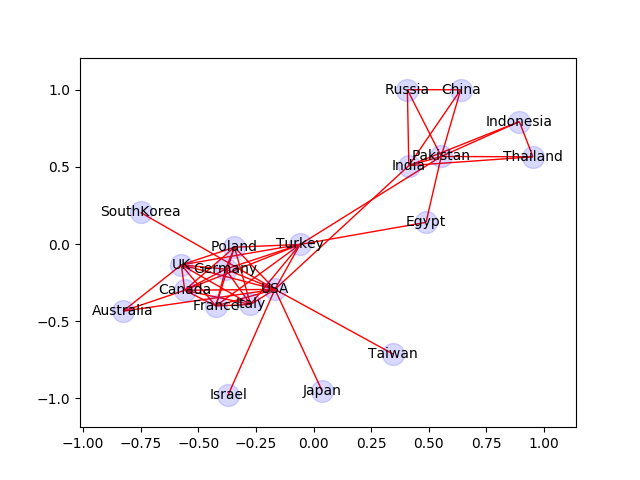
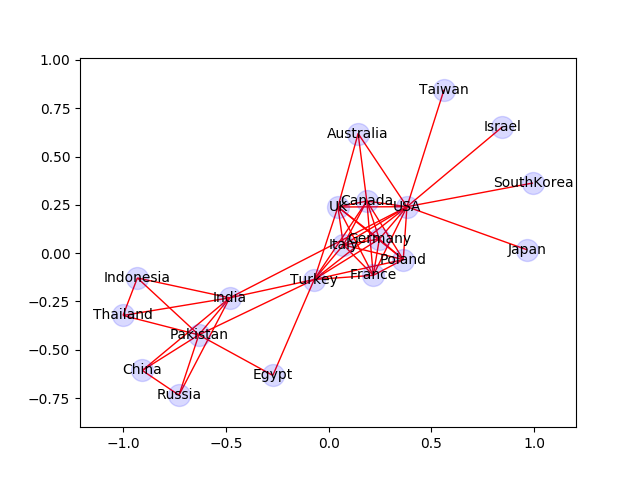
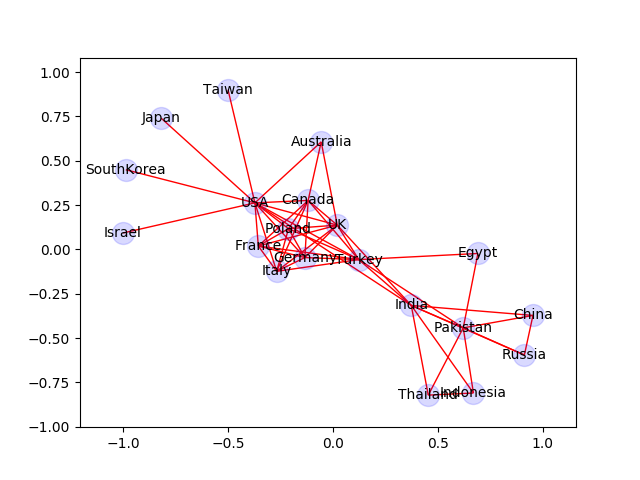
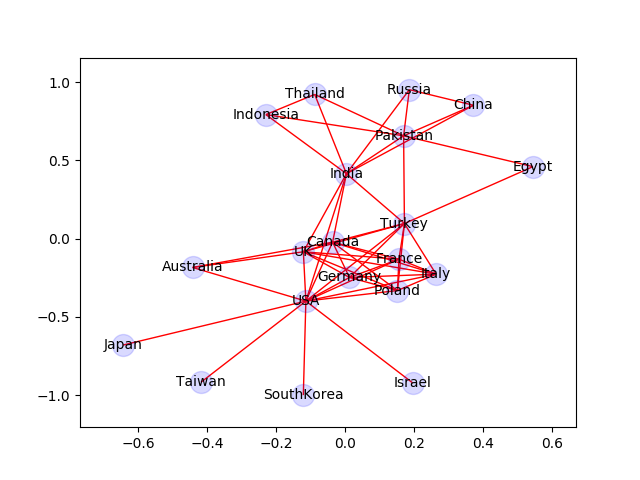
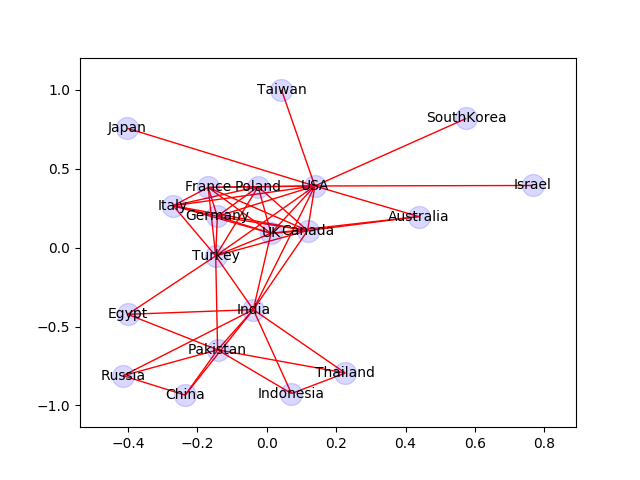
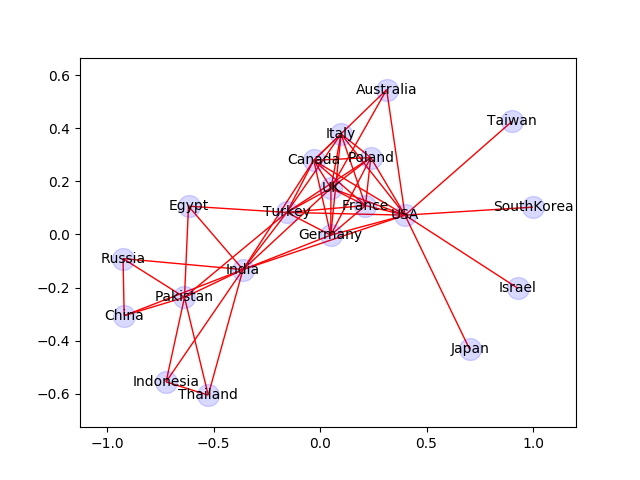
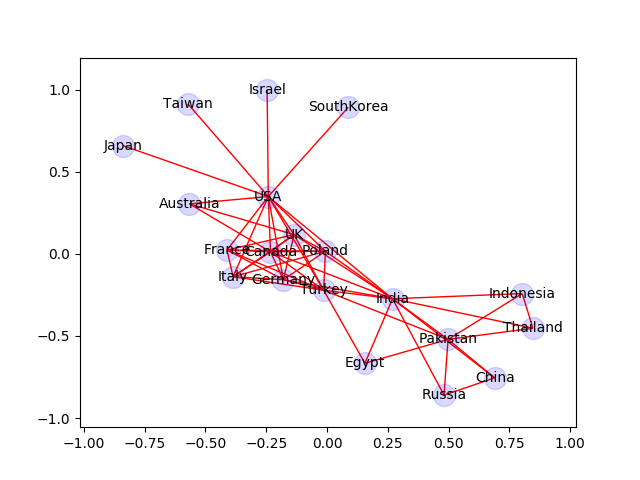
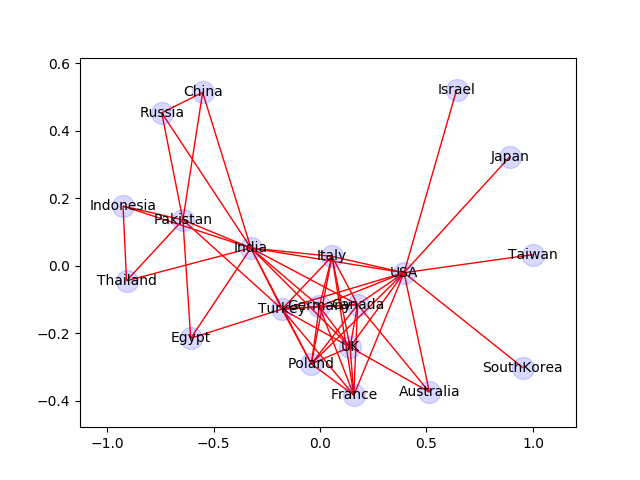
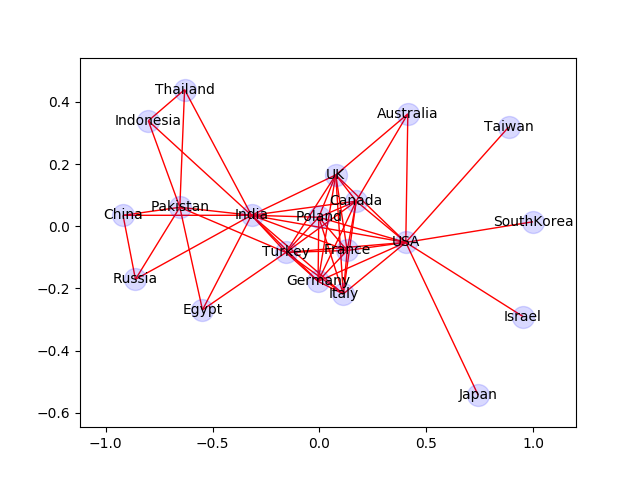
print("The Grpahs have been printed in 'graphs' directory.")

print("\n\*\*\*THANK YOU\*\*\*\n")

1. Output on terminal





1. Graph after 1st relation with USA
2. Graph after 2nd relation with Turkey
3. Graph after 3rd relation with UK
4. Graph after 4th relation with Canada
5. Graph after 5th relation with Egypt
6. Graph after 6th relation with Germany
7. Graph after 7th relation with Poland
8. Graph after 8th relation with Italy
9. Graph after 9th relation with France

**6. Results and Discussion**

The field of international relations is becoming more and more relevant in society today. Effective communication between countries is a key for making beneficial relationships and ensuring a safer world as a result. In short, international relations are all about power and weakness, war and peace, conflicts and cooperation and put simply, understanding of these behaviour patterns is the only way to change it. And there is so much to be changed!

**7. References**

1. Betzler, B. (2002). U.S. Patent Application No. 10/039,061.
2. Seo, K. S. (2007). U.S. Patent Application No. 11/716,242.