**情報工学実験IIレポート**

複雑ネットワーク分析の実践

人間情報システム工学科4年　45番

山口惺司

実験日： 2024年11月13日

締切： 2024年12月4日

提出日： 2024年12月4日

|  |  |
| --- | --- |
| 評価項目 | やった/一部やった/やっていない/何をやったかの概要 |
| **輸出または輸入の有向グラフを作成した** | やった |
| **エッジに重みを設定した** | やった |
| **グラフを描画した** | やった |
| **何らかの特徴量を求めた** | やった |
| **グラフやデータの説明を書いた** | やった |
| 追加的課題：1 | やった |
| 追加的課題： |  |

**背景と実験目的**

* 多数のノードとエッジからなる, ネットワークを再構築できる
* 有向グラフを用いることができる
* ノードやエッジに関する情報を, ネットワーク再構築に利用することができる
* 複数の機能を用いて, 複雑ネットワークの描画を行うことができる
* 各種中心性など, ネットワーク全体やノードの特徴値を求め, その意味を解釈し, 説明することができる

**課題1**

【問題】

**基礎的課題**

世界のほぼすべての国々の, もしくは, いくつかの注目した国々の間での, 輸出または輸入比率もしくは輸出入を元とした何らかの指標を元に, 有向グラフを作成する.

MatPlotLibで, 作成したグラフを描画する

各ノード, もしくはネットワークの任意の特徴を求める関数を最低1つ用いる

得られた特徴値と, ネットワークの構造からみてとれることからわかる事を説明する

**追加的課題1**

輸出入割合を重みとしてエッジを作成する際に, 「輸出割合\*輸出国のGDP」の値を用いるなどして, 貿易金額を想定して, エッジの重みに反映する

ノードを描画する際に, 円の直径をGDPに比例させる

追加の特徴を求める関数を用いて, 指標を表す値を求め, それによって得られる概要を説明する

**【アルゴリズム・解き方】**

　今回は2024年時点でのEU加盟国に注目して輸出率を元に, 有向グラフを作成した.

**解き方：**

1. EU加盟国に注目するために, export.csvに入っているEU加盟国をピックアップし, リストにする
2. ピックアップした国のGDPを調べ, リストにする
3. export.csvを開き, 各国のGDPと輸出割合からweightを求め, エッジを作成する
4. 各国のGDPからノードの大きさを設定し, 作成する
5. エッジ, ノード, ラベルをplotする
6. 媒介中心性を求め, printする

**【実行結果】**

3回実行したときに出力されたネットワークを図1~3に, 出力された媒介中心性の値を図4に示す.

ダイアグラム

自動的に生成された説明

図1 実行結果1

ダイアグラム, 概略図

自動的に生成された説明

図2 実行結果2

ダイアグラム

自動的に生成された説明

図3 実行結果3

新聞の記事のスクリーンショット

中程度の精度で自動的に生成された説明

図4 出力した媒介中心性の値

**【考察】**

　GDPの高いドイツやフランス, スペイン, イタリア, オランダなどは比較的ネットワークの中心にいることが分かる.

図4の媒介中心性の値を見ると, ドイツやオーストリア, フランス, スロバキアの値が高いことが分かる.

このような国はEU内での貿易ネットワーク内での中心的な役割を果たしていることを示している.

また, 値の低いマルタ, ルクセンブルク, アイルランドなどは, ネットワーク内での貿易中継地点としての役割はほとんどないということを示している.

小国である, または地理的に孤立している国であるなどの理由で媒介中心性の値が小さくなっていると考える.

**まとめ**

　基礎的課題, 追加的課題を満たしているため, 本実験は目的が達成されたといえる.

また, 今回はEUに注目してネットワークを作成し, 分析したが, GDPによる違いや, それぞれの国がネットワーク内でどのような役割を果たしているかなど, 視覚的に見ることができてよかった.

**参考文献**

グローバルノート- 国際統計・国別統計専門サイト 統計データ配信: https://www.globalnote.jp/post-1409.html

付録

**【プログラムソース】**

import csv

import networkx as nx

import matplotlib.pyplot as plt

countries = [

    "Austria", "Belgium", "Bulgaria", "Croatia", "Cyprus",

    "Czechia", "Denmark", "Estonia", "Finland", "France",

    "Germany", "Greece", "Hungary", "Ireland", "Italy",

    "Latvia", "Lithuania", "Luxembourg", "Malta", "Netherlands",

    "Poland", "Portugal", "Romania", "Slovakia", "Slovenia",

    "Spain", "Sweden"

]

countries\_gdp = [

    527662, 632399, 101611, 82711, 32239,

    343208, 407092, 41297, 295618, 3052712,

    4527009, 238275, 212464, 551554, 2301603,

    43640, 77844, 85780, 22335, 1154694,

    811736, 287187, 351074, 132832, 69168,

    1620558, 584914

]

# 有向グラフを作成

G = nx.DiGraph()

f = open('export.csv', 'r')

reader = csv.reader(f)

for row in reader:

    if len(row) >= 3:

        a = row[0]

        b = row[1]

        weight = float(row[2]) / 10

        if a in countries and b in countries:

            a\_index = countries.index(a)

            a\_gdp = countries\_gdp[a\_index]

            weight = float(row[2]) \* a\_gdp / 10000000

            G.add\_edge(a, b, weight=weight, type='export')

# ノード位置を決定

pos = nx.spring\_layout(G, k=1.1)

# ノードを描画

node\_sizes = []

for country in G.nodes:

    if country in countries:

        index = countries.index(country)

        node\_sizes.append(countries\_gdp[index] / 10000)

nx.draw\_networkx\_nodes(G, pos, node\_color='yellow', node\_size=node\_sizes)

# export.csvのエッジ

export\_edges = [(u, v) for u, v, d in G.edges(data=True) if d['type'] == 'export']

export\_weights = [G[u][v]['weight'] \* 10 for u, v in export\_edges]

nx.draw\_networkx\_edges(G, pos, edgelist=export\_edges, edge\_color='cyan', connectionstyle = 'arc3, rad = 0.3', width=1)

# ラベルを描画

nx.draw\_networkx\_labels(G, pos, font\_size=8)

# グラフを表示

plt.axis('off')

plt.show()

# 中央性の計算

b=nx.betweenness\_centrality(G)

for k,v in sorted(b.items()):

      print(str(k) + ": " + str(v))