

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。

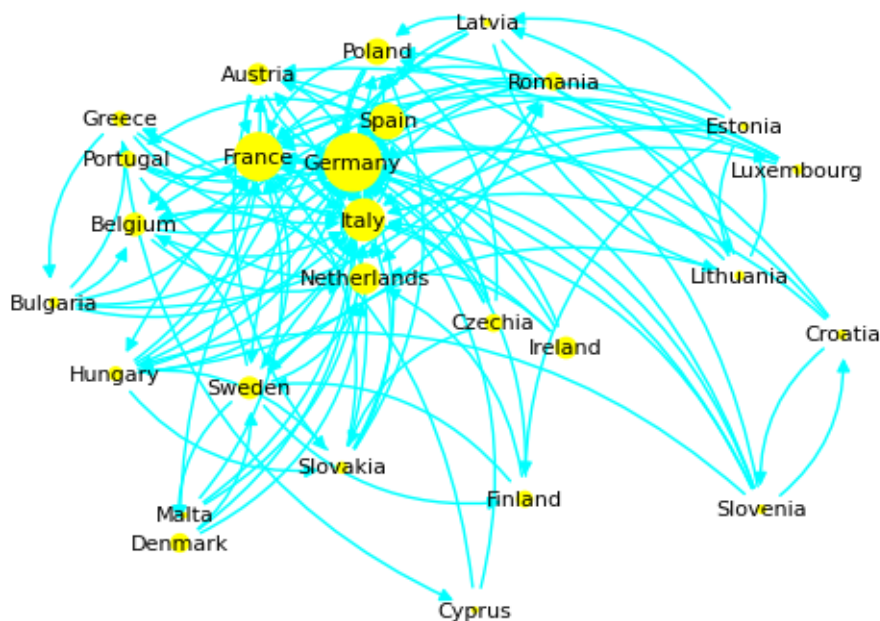


図 1 実行結果 1

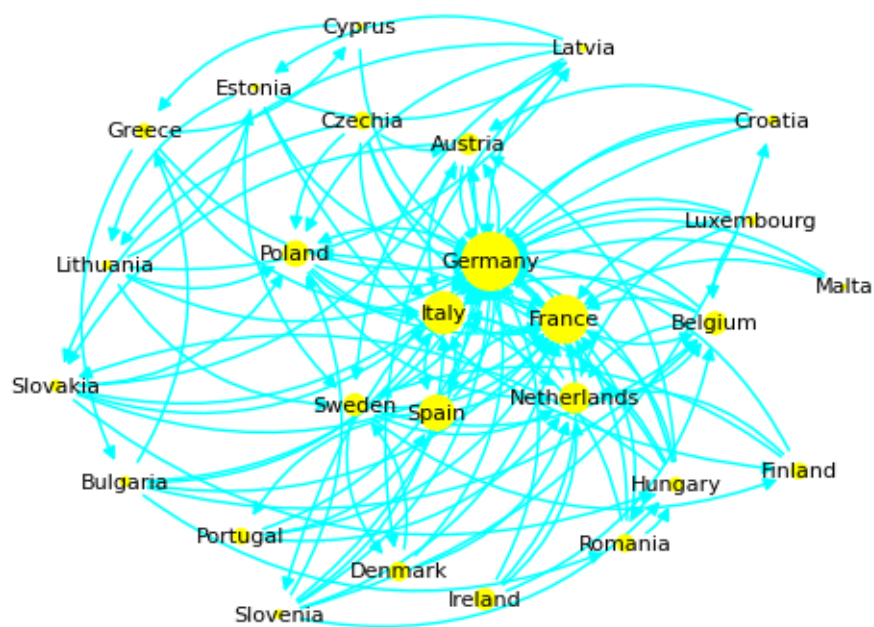


図 2 実行結果 2

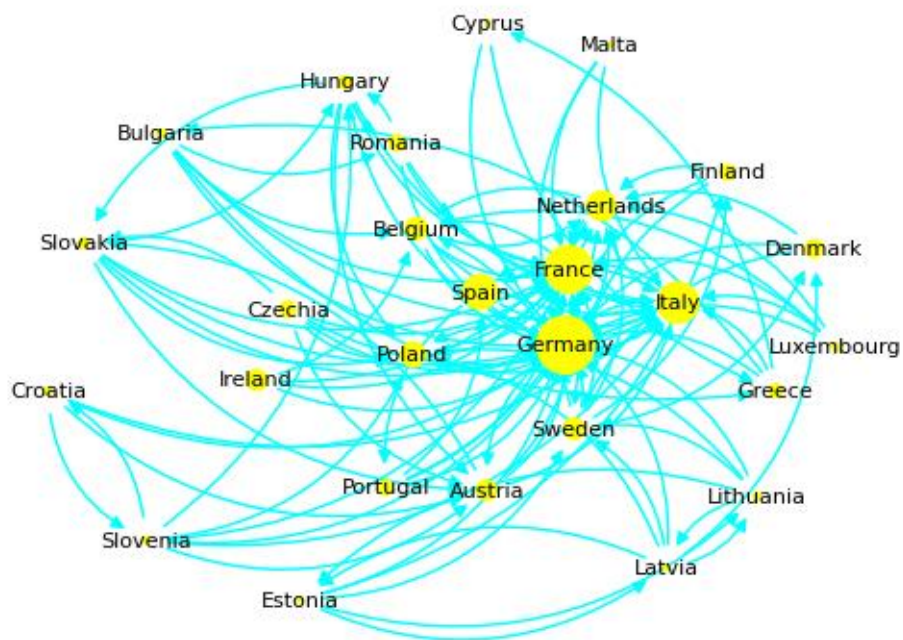


図 3 実行結果 3

Austria: 0.08607692307692308
Belgium: 0.0023076923076923075
Bulgaria: 0.007948717948717949
Croatia: 0.0
Cyprus: 0.0
Czechia: 0.0
Denmark: 0.0003846153846153846
Estonia: 0.0015384615384615385
Finland: 0.0005128205128205128
France: 0.05566666666666666
Germany: 0.1743846153846154
Greece: 0.007692307692307693
Hungary: 0.038461538461538464
Ireland: 0.0
Italy: 0.035358974358974365
Latvia: 0.0020512820512820513
Lithuania: 0.0005128205128205128
Luxembourg: 0.0
Malta: 0.0
Netherlands: 0.010846153846153846
Poland: 0.009
Portugal: 0.0007692307692307692
Romania: 0.004615384615384615
Slovakia: 0.06
Slovenia: 0.004538461538461539
Spain: 0.03884615384615384
Sweden: 0.020025641025641024

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たることが分かる。

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、視覚的に見る事ができてよかった。

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))
```