

レポート課題：プログラミング応用

XXX-XXXXXXX-X 松岡大輔

July 9, 2025

1 レポートのタイトル

SNAP を利用したソーシャルネットワークのグラフ可視化とオイラー閉路の存在検証

2 取得したデータの説明

SNAP (Stanford Network Analysis Project)¹ で公開されているデータセットを用いた。データセットの詳細は以下に記載されている。

"Social circles: Facebook" <https://snap.stanford.edu/data/ego-Facebook.html>

ソーシャル・ネットワーク・サービスの Facebook におけるサークル、あるいは友だちリストのデータを取得して匿名化したものである。今回、無向グラフの分析を前提として、SNS のデータを採した。Facebook は他の SNS(たとえば X や instagram) と比較して、相互フォローの割合が高いため、ノード間のフォロー・フォロワー関係を無向グラフとして捉えることが可能であるとされる。有向グラフを無向グラフに変換することも可能であるらしいが、今回はその方法は取らなかった。

また、過去には SNS の API から直接データを取得して分析することが比較的容易であったが、たとえば X は昨今の仕様変更で、無料ユーザはデータの取得に制限がある。有料ユーザの料金はそれなりに高額であり、今回のレポートのために課金するわけにはいかなかったので、公開されている匿名化されたデータセットを利用した。やや古いデータではあるが、今回の分析目的には十分役立つと判断した。

3 コードの主要な部分とその説明

コードは Python で記述し、ライブラリとして以下の三つを利用した。

- snap
- networkx
- matplotlib

snap は、SNAP で公開されているライブラリである。networkx と matplotlib でおおむね代替可能なのだが、SNAP で公開されているデータセットのファイルの処理が簡便であったため、データの取り込み部分に採用した。

¹<https://snap.stanford.edu/>

networkx は取り込んだデータのグラフ化に利用した。matplotlib は描画に利用した。実行環境は Google Colab である。

実行時に問題となったのは、snap が 2020 年以來アップデートされていないため、Python 3.7 までにしか対応しておらず、最新のバージョンではインストールできないという不具合だった。今回、Google Colab で利用する Python のバージョンを一時的に 3.7 に下げることに対応した。以下はデータの処理に関するコードの一部である。

Listing 1: データ処理

```
1 import snap
2 import networkx as nx
3 import matplotlib.pyplot as plt
4
5 g = snap.LoadEdgeList(snap.TNGraph, "0.edges", 0, 1) # "0.edges" is the
      name of data file
6 snap.SaveEdgeList(g, "graph.txt", "Save as tab-separated list of edges"
      ) # save the graph as graph.txt
7
8 G_nx_loaded = nx.read_edgelist("graph.txt", create_using=nx.DiGraph())
      # load the graph
9
10 # グラフの可視化
11 pos = nx.spring_layout(G_nx_loaded, seed=42) # You can choose a
      different layout algorithm
12 nx.draw(G_nx_loaded, pos, with_labels=False, node_size=10, width=0.5)
13 plt.title("Graph Visualization (from graph.txt)")
14 plt.show()
```

4 可視化と分析の結果

これに対して、このグラフにオイラー閉路が存在するかどうかの検証を行った。チェックするための関数は networkx に存在するので、それを利用する。

Listing 2: オイラー閉路の存在チェック

```
1 has_eulerian_circuit = nx.is_eulerian(G_nx_loaded)
2
3 print(has_eulerian_circuit)
```

結果は False である。つまり、オイラー閉路は含まれない。

さらに、このグラフの部分グラフも含めて、オイラー閉路が存在するか検証しようとしたが、計算量が膨大になるため現実的ではない。したがって、部分グラフの一部をランダムにサンプリングして、オイラー閉路の存在確率を仮に計算した。コードは以下の通りである。なお、コードの生成には AI を活用し、適宜手を加えた。

Listing 3: オイラー閉路の存在チェック

```
1 import random
2
3 # サンプリング・パラメータの定義
```

Graph Visualization (from graph.txt)



Figure 1: 可視化の結果

```
4 num_subgraphs_to_sample = 100
5 subgraph_size = 50
6
7 eulerian_subgraphs_count = 0
8 total_subgraphs_checked = 0
9
10 # サブグラフのサンプリングとオイラー閉路の存在チェック
11 all_nodes = list(G_nx_loaded.nodes())
12 num_nodes_in_graph = len(all_nodes)
13
14 print(f"元のグラフのノード数: {num_nodes_in_graph}")
15 print(f"サンプリングする部分グラフの数: {num_subgraphs_to_sample}")
16 print(f"各部分グラフのサイズ ノード数(): {subgraph_size}")
17
18 if subgraph_size > num_nodes_in_graph:
19     print("エラー: 部分グラフのサイズが元のグラフのノード数より大きいです。
20         subgraph_size を調整してください。")
21 else:
22     for i in range(num_subgraphs_to_sample):
23         # Randomly select a subset of nodes
24         sampled_nodes = random.sample(all_nodes, subgraph_size)
25
26         # Create an induced subgraph
27         subgraph = G_nx_loaded.subgraph(sampled_nodes)
28
29         # Check if the subgraph has an Eulerian circuit
30         if nx.is_eulerian(subgraph):
31             eulerian_subgraphs_count += 1
```

```

31     total_subgraphs_checked += 1
32     print(f"チェック済み部分グラフ数: {total_subgraphs_checked}/{
33           num_subgraphs_to_sample}", end='\r')
34
35     print(f"\チェック完了。n")
36     print(f"オイラー閉路を含む部分グラフの数: {eulerian_subgraphs_count}")
37     print(f"チェックした部分グラフの総数: {total_subgraphs_checked}")
38
39     if total_subgraphs_checked > 0:
40         percentage_eulerian = (eulerian_subgraphs_count /
41                                total_subgraphs_checked) * 100
42     print(f"オイラー閉路を含む部分グラフの割合: {percentage_eulerian:.2f}%")

```

出力は以下の通り。

```

元のグラフのノード数: 333
サンプリングする部分グラフの数: 100
各部分グラフのサイズ (ノード数): 50
チェック済み部分グラフ数: 100/100
チェック完了。
オイラー閉路を含む部分グラフの数: 0
チェックした部分グラフの総数: 100
オイラー閉路を含む部分グラフの割合: 0.00%

```

つまり、部分グラフまで含めても、オイラー閉路の存在確率はゼロであると推定される。同様のデータが他に9ファイルあるので、それらについても検証するべきだが、レポートの提出期限が迫っているので、追って検証することにする。

Facebookのようなソーシャル・ネットワーク・サービスにおいてオイラー閉路が存在するとしたら、それはどういう状態かということ、それはオイラー閉路の定義より「そのグラフのノードが連結、かつ、全ノードの次数が偶数」である。このような状態は理論的には存在しうるものの、現実にはおそらく存在しないだろうということが推定された。