研究プロジェクト結果

16B13360

Hana Hoshino

TOOL

- **Name:** Networkx
- Written by pure Python

  → Easy to install
- Demo on the right

```
import networkx as nx
import matplotlib.pyplot as plt


# Import file
g = nx.read_edgelist(
        'facebook_combined.txt',
        create_using=nx.Graph(),
        nodetype=int
        )


nx.draw_networkx(g)
plt.show(g)
```

TOOL

N=39,

20 to 170 times slower

Algorithm				NetworkX
Single-source shortest p				0.152 s
PageRank				3.949 s
K-core	0.014 s	0.014 s	0.022 s	0.714 s
Minimum spanning tree	0.040 s	0.031 s	0.044 s	2.045 s
Betweenness	244.3 s (~4.1 mins)	601.2 s (~10 mins)	946.8 s (edge) + 353.9 s (vertex) (~ 21.6 mins)	32676.4 s (edge) 22650.4 s (vertex) (~15.4 hours)

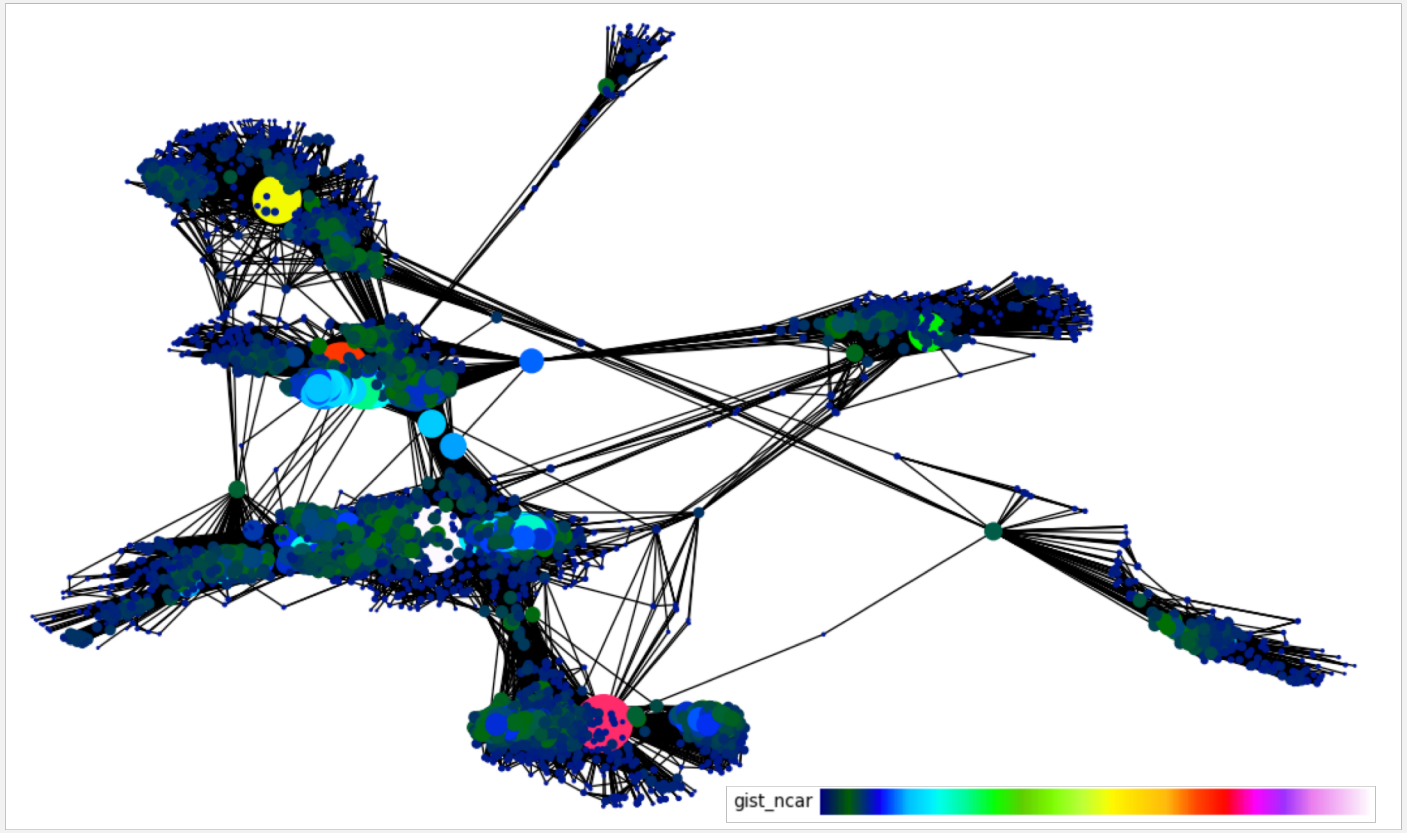
https://graph-tool.skewed.de/performance

DATA

- **Name：** Social circles: Facebook
- **File Format:** text file
- **About:** A dataset of 'friends list' from Facebook 2012
- **Nodes:** 4039
- **Edges:** 88234

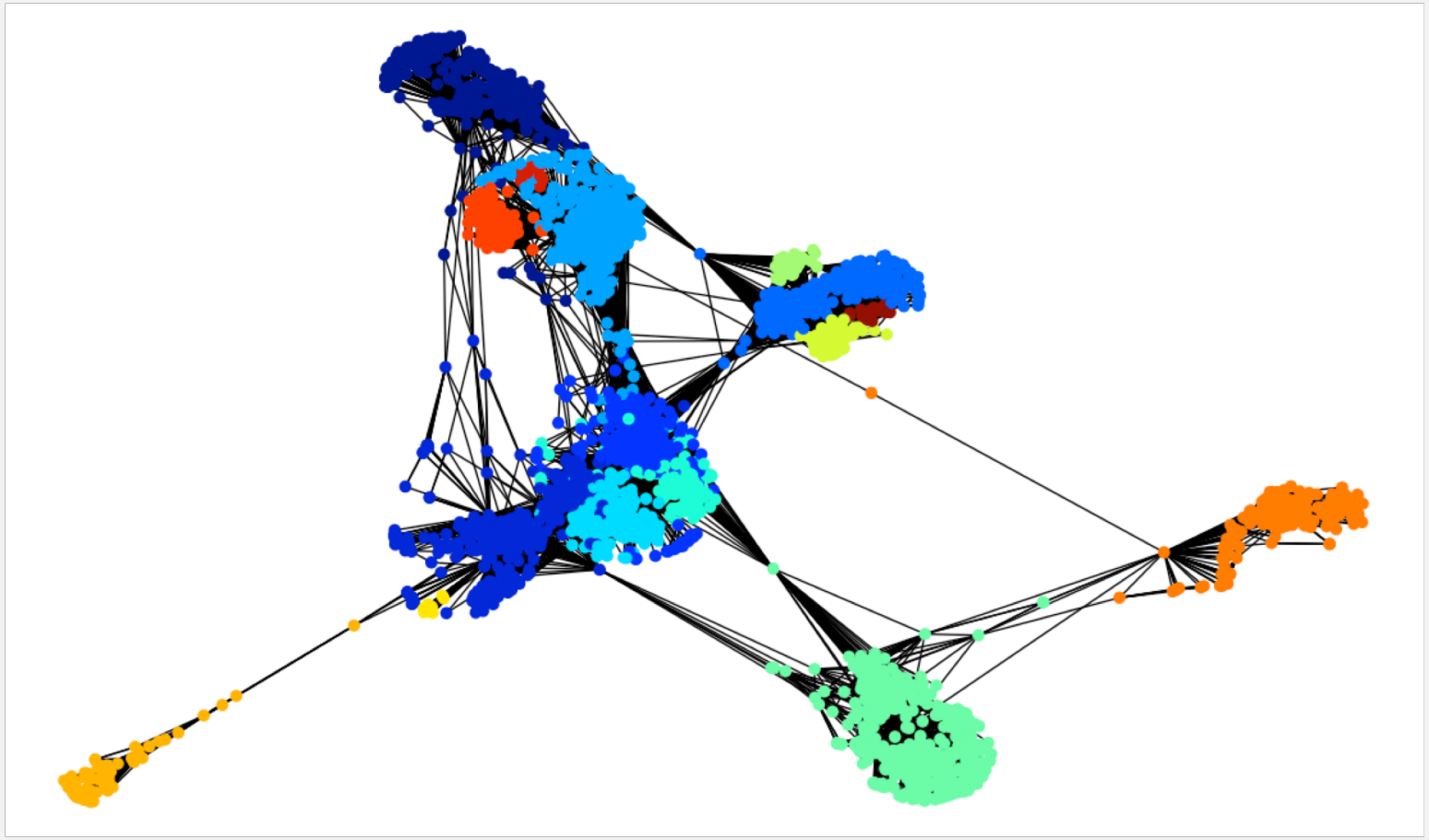
Person A's ID	Friend α's ID
Person A's ID	Friend β's ID
Person B's ID	Friend γ's ID

OUTPUT



Average Degree: 43.6910
Layout: Spring Layout

OUTPUT



With the help
of module '**community**'

RESULT

- **Average Degree:** 43.6910
- **Average Shortest Path**: 3.6923
- **Community:** 16
- **Max Degree:** 1045
- **Min Degree:** 1
- **Max Degree Centrality:** 0.2587914809311154
- **Min Degree Centrality:** 0.0002476473501733531 3

FUTURE WORKS

- Add more node attributes (ex: age, education, country, etc.) and analyze each community
- Look at each community in more details

REFERENCE

- https://blog.dominodatalab.com/social-network-analysis-with-networkx/
- https://matplotlib.org/mpl_examples/color/colormaps_reference_05.png
- https://snap.stanford.edu/data/ego-Facebook.html

THANK YOU FOR LISTENING