

# Complex Network Analysis

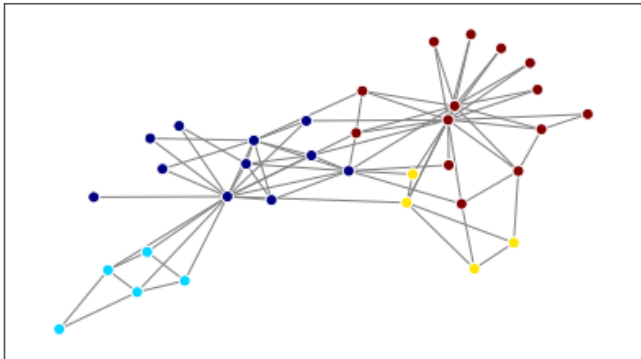
## Seminar, Project

Popović Milutin

17. March 2021

```
[1]: import networkx as nx
```

```
[2]: nx.plot_nice_graph(karate_graph)
```



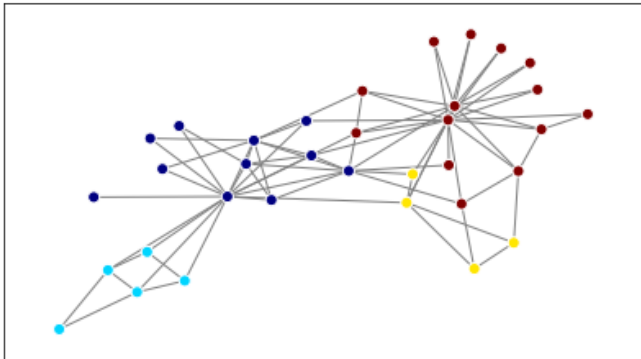
```
$ pip install networkx
```

```
$ pip install networkx
```

→ Go to PyPi (Python Package Index) and download & install networkx

```
[1]: import networkx as nx
```

```
[2]: nx.plot_nice_graph(karate_graph)
```



## Find, install and publish Python packages with the Python Package Index

Search projects 

Or [browse projects](#)

363,125 projects	3,303,299 releases	5,739,682 files	578,672 users
------------------	--------------------	-----------------	---------------

Nodes → Repositories

Edges → Requirements(*Imoprts*)

Directed Graph  
`scipy` → `numpy`



## Directed Graph

scipy  $\rightarrow$  numpy

scipy  $\nrightarrow$  numpy

DiGraph with 361,742 nodes and 719,797 edges

DiGraph with 361,742 nodes and 719,797 edges

→ ~362,000 out of ~363,000 (Not what I expected)!

DiGraph with 361,742 nodes and 719,797 edges

→ ~362,000 out of ~363,000 (Not what I expected)!

→ Significant data has  $k_i > 0 \rightarrow \sim 164,000$  nodes and  $\sim 720,000$  edges

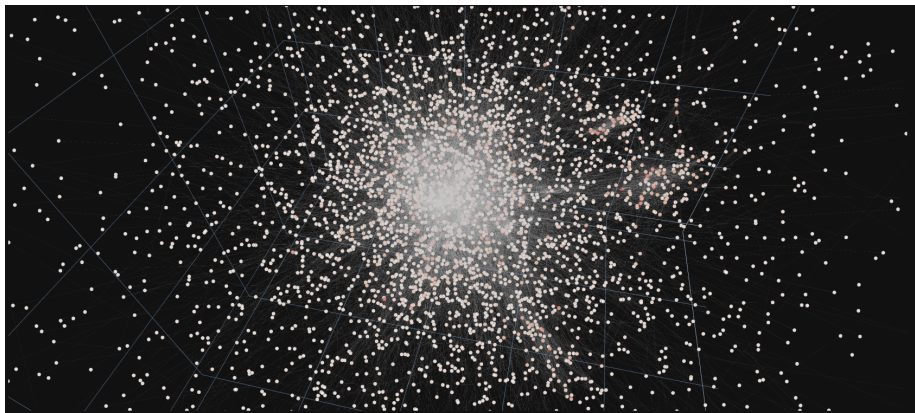


Figure: Plot of the Python Dependencies Network  $k_{in} \geq 10$ ,  $N = 4826$ ,  $L = 20847$   
(red nodes indicate high degree)

## Basic Properties

$$\langle k_{\text{in}} \rangle = 1.99$$

$$\langle k_{\text{out}} \rangle = 1.99$$

$$\langle k \rangle = 8.75$$

$$\langle k^2 \rangle = -$$

## Specific Properties

$$S \simeq 1$$

*Connected*

$$\langle d_{\text{min}} \rangle = 7.63$$

Small-World

$$d_{\text{max}} = 18.09$$

$$r = -0.125$$

Assortative

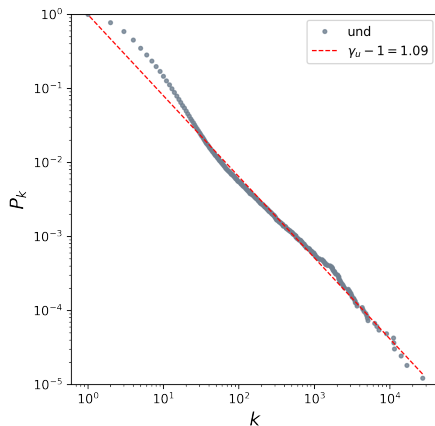
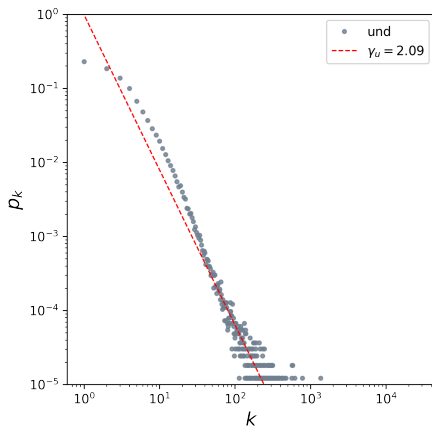


Figure: Undirected (left) distribution  $p_k$  (right) cumulative degree distribution with fit for  $k > 0$

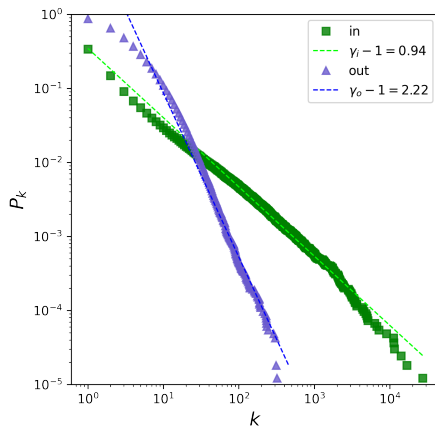
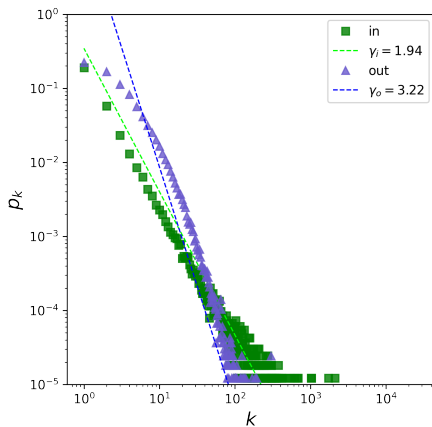


Figure: Directed (left) distribution  $p_k$  (right) cumulative degree distribution with fit for  $k > 0$



Louvain-Algorithm

Optimize **Modularity**  $M$   
 $O(L)$

Info-Map-Algorithm

Optimize **Map-Equation**  $\mathcal{L}$   
 $O(N \log(N))$

Louvain-Algorithm

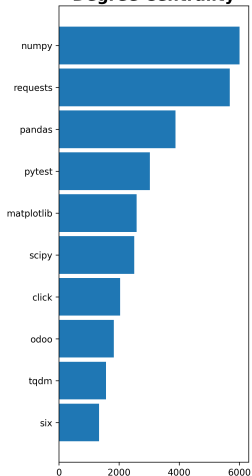
Optimize **Modularity**  $M$   
 $O(N)$

Info-Map-Algorithm

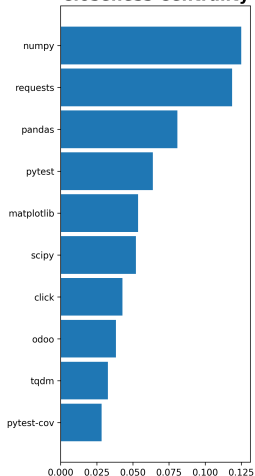
Optimize **Map-Equation**  $\mathcal{L}$   
 $O(N \log(N))$

→ Detect & Analyze Community Structure

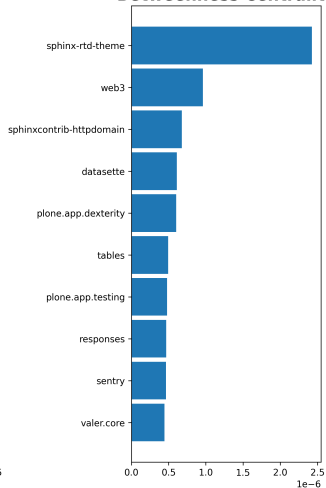
## Degree Centrality



## Closeness Centrality



## Betweenness Centrality



- [1] A.L. Barabási. *Network Science*. Cambridge University Press, 2016. ISBN: 9781107076266. URL:  
<https://books.google.at/books?id=iLtGDQAAQBAJ>.
- [2] Kevin Gullikson. *Python Dependency Analysis*. URL:  
<https://kgullikson88.github.io/blog/pypi-analysis.html>  
(visited on 03/15/2022).
- [3] Python Software Foundation. *Python Package Index*. URL:  
<https://pypi.org/> (visited on 03/15/2022).

*To be continued...*

*Thank You!*