1-3

November 5, 2021

Problem 1-3
Assignment 2

group members: 1) Shuhan Xiao (Uni-id: kg410 Matrikelnr.: 3160697), 2) Klaus Kades (Uni-id: fw448 Matrikelnr.: 3408463), 3) Lucas-Raphael Müller (Uni-id: al413 Matrikelnr.: 3205638), 4) Melanie Schellenberg (Uni-id: qh400 Matrikelnr.: 3146390)

[ ]:
```
# Install a pip package in the current Jupyter kernel
! pip install numpy pandas python-igraph matplotlib pycairo cairocffi networkx
```

```
Requirement already satisfied: numpy in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (1.21.3)
Requirement already satisfied: pandas in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (1.3.4)
Requirement already satisfied: python-igraph in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (0.9.7)
Requirement already satisfied: matplotlib in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (3.4.3)
Requirement already satisfied: pycairo in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (1.20.1)
Requirement already satisfied: cairocffi in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (1.3.0)
Requirement already satisfied: networkx in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (2.6.3)
Requirement already satisfied: python-dateutil>=2.7.3 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
pandas) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
pandas) (2021.3)
Requirement already satisfied: texttable>=1.6.2 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
python-igraph) (1.6.4)
Requirement already satisfied: pillow>=6.2.0 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
matplotlib) (8.3.2)
Requirement already satisfied: cycler>=0.10 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
```

```
matplotlib) (0.10.0)
Requirement already satisfied: pyparsing>=2.2.1 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
matplotlib) (3.0.3)
Requirement already satisfied: kiwisolver>=1.0.1 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
matplotlib) (1.3.2)
Requirement already satisfied: cffi>=1.1.0 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
cairocffi) (1.14.6)
Requirement already satisfied: pycparser in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
cffi>=1.1.0->cairocffi) (2.20)
Requirement already satisfied: six in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
cycler>=0.10->matplotlib) (1.16.0)
```

[ ]:
```
!ls /workplace/CNA/Complex-Network-Analysis-Exercises/assignment-2/
 ↪ucidata-zachary/*
!head /workplace/CNA/Complex-Network-Analysis-Exercises/assignment-2/
 ↪ucidata-zachary/out.ucidata-zachary
```

```
/workplace/CNA/Complex-Network-Analysis-Exercises/assignment-2/ucidata-
zachary/meta.ucidata-zachary
/workplace/CNA/Complex-Network-Analysis-Exercises/assignment-2/ucidata-
zachary/out.ucidata-zachary
/workplace/CNA/Complex-Network-Analysis-Exercises/assignment-2/ucidata-
zachary/README.ucidata-zachary
% sym unweighted
% 78 34 34
1 2
1 3
2 3
1 4
2 4
3 4
1 5
1 6
```

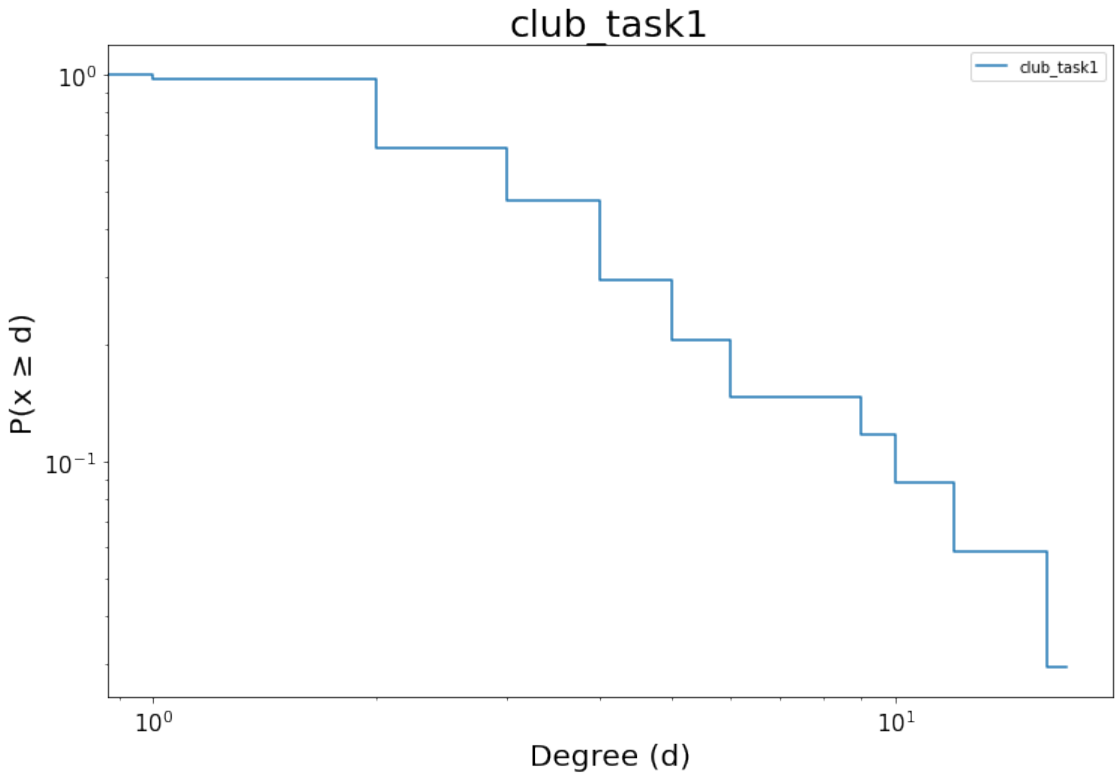
[ ]:
```
import matplotlib.pyplot as plt
import numpy as np

#calculation of cumulative values
def calculate(degree_freq, threshold):
    cumulative_value=0
    for degree, i in enumerate(range(len(degree_freq))):
        if degree>=threshold:
```

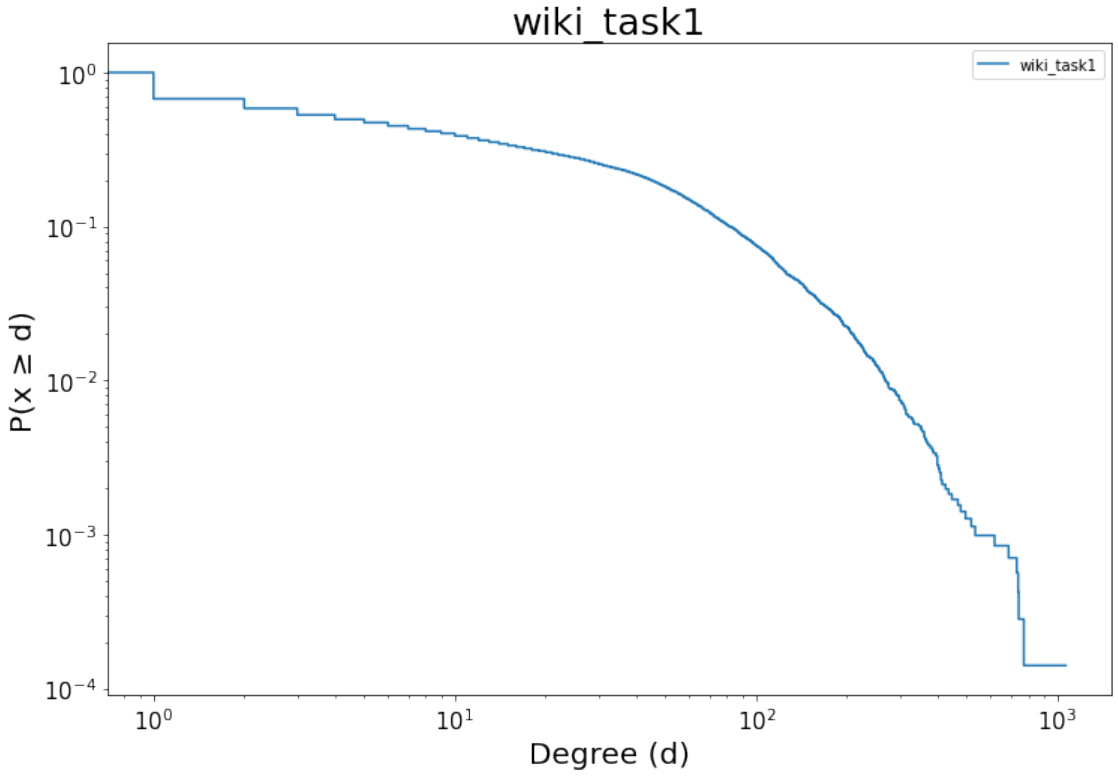
```
            cumulative_value+=degree_freq[i]
        else:
            continue
    return cumulative_value


def cumulative_degree_p(data, save_name, task2):

    #generation of graph
    g = nx.Graph()
    for x,y in data.itertuples(index=False):
        g.add_edge(x, y)
    graphs=[g]
    name={}
    name[0]=save_name[:-4]
    fig = plt.figure(figsize=(12, 8))
    ax = fig.add_subplot(1, 1, 1)

    if task2==True:
        g_random =nx.gnm_random_graph(n=len(g.nodes), m=len(g.edges), seed=2,␣
↪directed=False)
        graphs=[g,g_random]
        name[1]='random'


    for id, graph in enumerate(graphs):
        #get degree distribution
        degree_freq = nx.degree_histogram(graph)

        #calculation of cumulative degree distribution
        cumulative_degree_prob=[]
        for threshold in range(len(degree_freq)):
            cumulative_value = calculate(degree_freq, threshold)
            if threshold==0:
                normalization_value= cumulative_value
            cumulative_degree_prob.append(cumulative_value)

        #normalization of cumulative degree distribution
        cumulative_degree_prob_norm=[value/normalization_value for value in␣
↪cumulative_degree_prob]
        degrees=len(degree_freq)

        #plotting
        ax.step(range(degrees) , cumulative_degree_prob_norm, label=name[id])
        ax.set_xlabel('Degree (d)', fontsize=20)
        ax.set_ylabel('P(x \u2265 d)', fontsize=20)
        ax.tick_params(axis='both', which='major', labelsize=15)
```

```
        ax.set_yscale('log')
        ax.set_xscale('log')
        ax.legend()
        plt.title(save_name[:-4], fontsize=25)
    plt.savefig(save_name)
```

[ ]:
```
"""
Task1
"""
import pandas as pd
import networkx as nx

# #Zachary data (N<250)
colnames=[ 'X', 'Y']
club = pd.read_csv('/workplace/CNA/Complex-Network-Analysis-Exercises/
 ↪assignment-2/ucidata-zachary/out.ucidata-zachary', sep=' ', skiprows=2,␣
 ↪names=colnames)


cumulative_degree_p(club, 'club_task1.png', task2=False)


#WikiVote data (N>2500)
colnames=[ 'X', 'Y']
wiki = pd.read_csv('/workplace/CNA/Complex-Network-Analysis-Exercises/
 ↪assignment-2/Wiki-Vote.txt', sep='\t', skiprows=4, names=colnames)

print(wiki)

cumulative_degree_p(wiki, 'wiki_task1.png', task2=False)
```

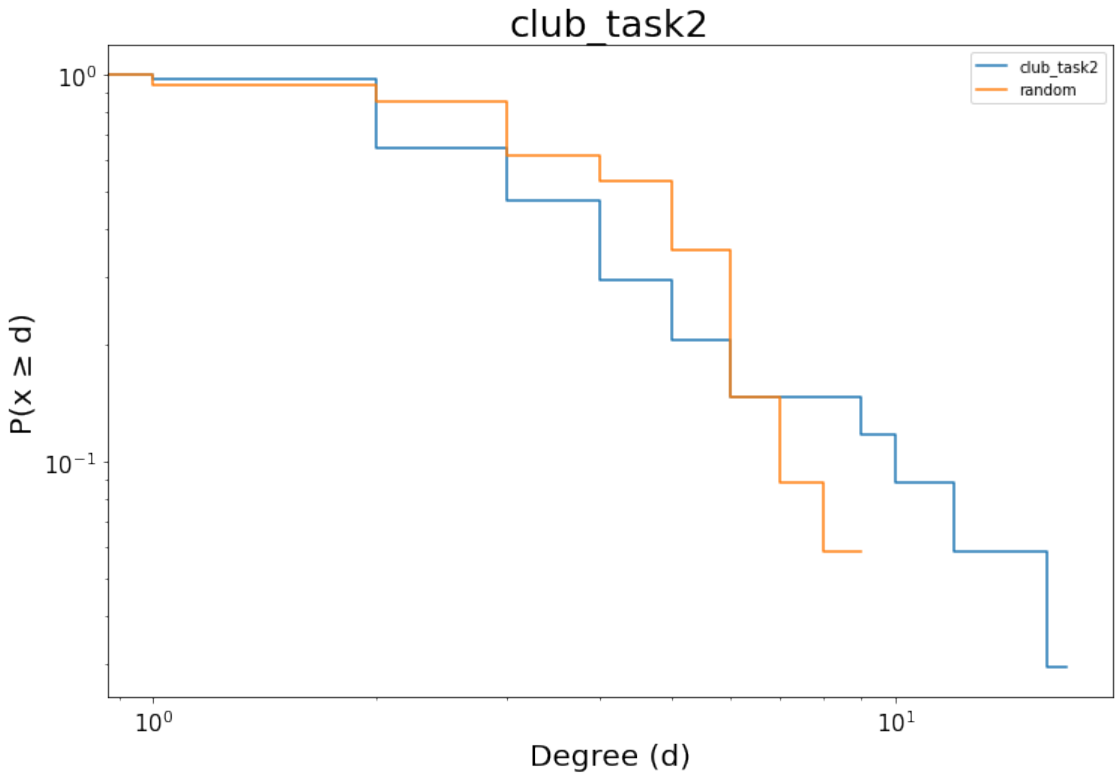
```
           X     Y
0         30  1412
1         30  3352
2         30  5254
3         30  5543
4         30  7478
…       …   …
103684  8272  4940
103685  8273  4940
103686  8150  8275
103687  8150  8276
103688  8274  8275

[103689 rows x 2 columns]
```

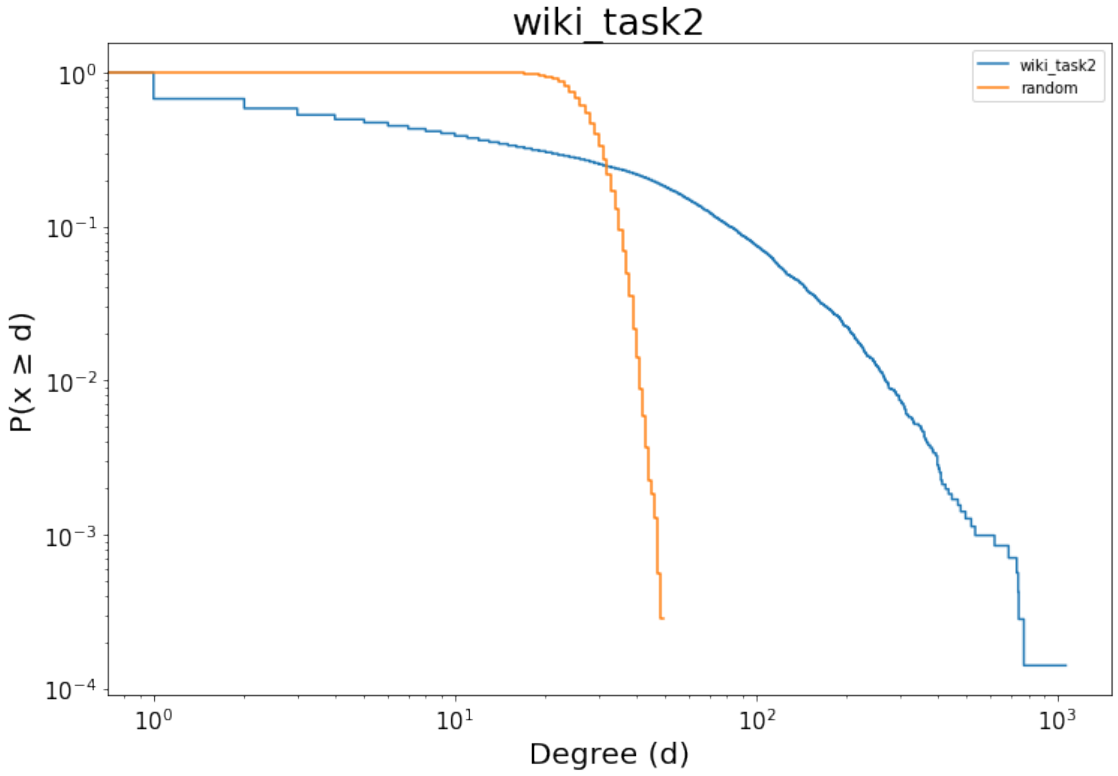

club_task1
club_task1
10^0
10^{-1}
P(x ≥ d)
10^0
10^1
Degree (d)


wiki_task1
wiki_task1
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
P(x ≥ d)
10^0
10^1
10^2
10^3
Degree (d)

[ ]:
```
import pandas as pd
import networkx as nx

# #Zachary data (N<250)
colnames=[ 'X', 'Y']
club = pd.read_csv('/workplace/CNA/Complex-Network-Analysis-Exercises/
 ↪assignment-2/ucidata-zachary/out.ucidata-zachary', sep=' ', skiprows=2,␣
 ↪names=colnames)


cumulative_degree_p(club, 'club_task2.png', task2=True)


#WikiVote data (N>2500)
colnames=[ 'X', 'Y']
wiki = pd.read_csv('/workplace/CNA/Complex-Network-Analysis-Exercises/
 ↪assignment-2/Wiki-Vote.txt', sep='\t', skiprows=4, names=colnames)

print(wiki)

cumulative_degree_p(wiki, 'wiki_task2.png', task2=True)
```

```
            X     Y
0          30  1412
1          30  3352
2          30  5254
3          30  5543
4          30  7478
…         …    …
103684   8272  4940
103685   8273  4940
103686   8150  8275
103687   8150  8276
103688   8274  8275

[103689 rows x 2 columns]
```


club_task2
club_task2
random
10^0
10^{-1}
P(x ≥ d)
10^0
10^1
Degree (d)


wiki_task2
wiki_task2
random
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
P(x ≥ d)
10^0
10^1
10^2
10^3
Degree (d)

[ ]: