

به نام خدا



دانشکده مهندسی صنایع و سیستم ها

گروه فناوری اطلاعات

نام و نام خانوادگی: محمدرضا صبی پور

شماره دانشجویی: ۴۰۳۶۶۲۴۱۰۰۳

تمرین ۱ - مرکزیت و اجتماع یابی (انفرادی)

درس:

تحلیل شبکه های اطلاعات

استاد:

دکتر بابک تیمورپور

۱۴۰۳/۱۲/۸

فهرست

۳.....	شرح تمرین.....
۳.....	مقدمه.....
۳.....	جمع آوری داده.....
۳.....	تشکیل گراف.....
۴.....	محاسبه مرکزیت و خروجی گراف.....
۴.....	مصورسازی در gephi.....
۶.....	اجتماع‌یابی و مقدار ماژولاریتی.....

شرح تمرین

هدف از این تمرین جمع آوری داده کلوپ کاراته زاخاری از اینترنت، ترسیم گراف آن و محاسبه مرکزیت ها و ماژولاریتی است. در این تمرین ابتدا داده کلوپ کاراته زاخاری را دانلود کرده و به صورت لیست برداری ذخیره می کنیم. در ادامه این داده در کتابخانه `igraph` خوانده و گراف آن را تشکیل می دهیم. سپس این گراف را به همراه مرکزیت محاسبه شده خروجی گرفته و در نرم افزار `gephi` مصورسازی می کنیم. در انتها با استفاده از یک الگوریتم اجتماع یابی، این اجتماع ها را پیدا کرده و مقدار ماژولاریتی را محاسبه می کنیم.

مقدمه

این دیتاست شامل دو بخش می باشد: مربی و مدیریت. به دلیل وجود اختلاف نظر بین این دو نفر، باشگاه به دو بخش تقسیم شده است. اعضای جدید که عضو این باشگاه می شوند باید یکی از این دو نفر را انتخاب کنند.

جمع آوری داده

داده ورودی شامل ۳۵ راس و ۷۶ یال می باشد که از وب سایت زیر دریافت شده است و در یک فایل با فرمت CSV ذخیره شده است.

<https://networks.skewed.de/net/karate>

داده ها به صورت لیست برداری ذخیره شده اند به این صورت که هر یال یک مشاهده است که شامل دو ویژگی: راس شروع و راس انتهایی می باشد.

تشکیل گراف

در این مرحله با استفاده از کتابخانه `igraph` در زبان پایتون، دیتای دانلود شده را خوانده و سپس گراف آن را تشکیل می دهیم. قطعه کد زیر مراحل فوق را انجام می دهد.

```

import igraph as ig
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import csv
import math

edges = []
# url : https://networks.skewed.de/net/karate
with open('edges.csv', 'r') as f:
    reader = csv.DictReader(f)
    for line in reader:
        edges.append((int(line['# source']), int(line['# target'])))

g = ig.Graph(35)
g.add_edges(edges)

```

محاسبه مرکزیت و خروجی گراف

بعد از انجام مراحل بالا لازم است که مقدار مرکزیت دلخواه را محاسبه کنیم. برای این تمرین مرکزیت بینابینی در نظر گرفته شده است. سپس این مرکزیت به همراه گراف در یک فایل graphml ذخیره می‌شود.

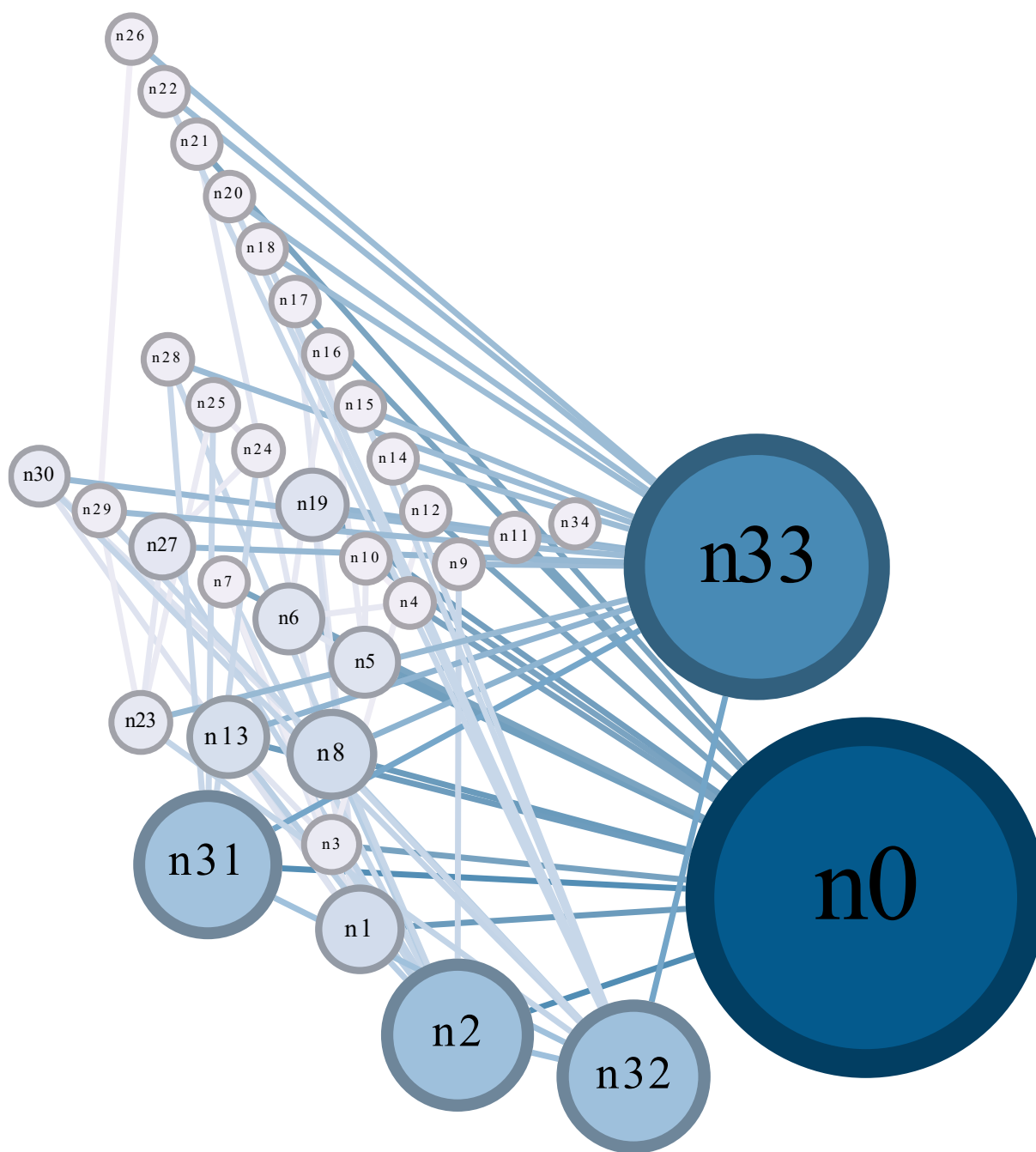
```

betweenness = [0 if math.isnan(v) else v for v in g.betweenness()]
g.vs['betweenness'] = betweenness
g.write_graphml('graph.graphml')

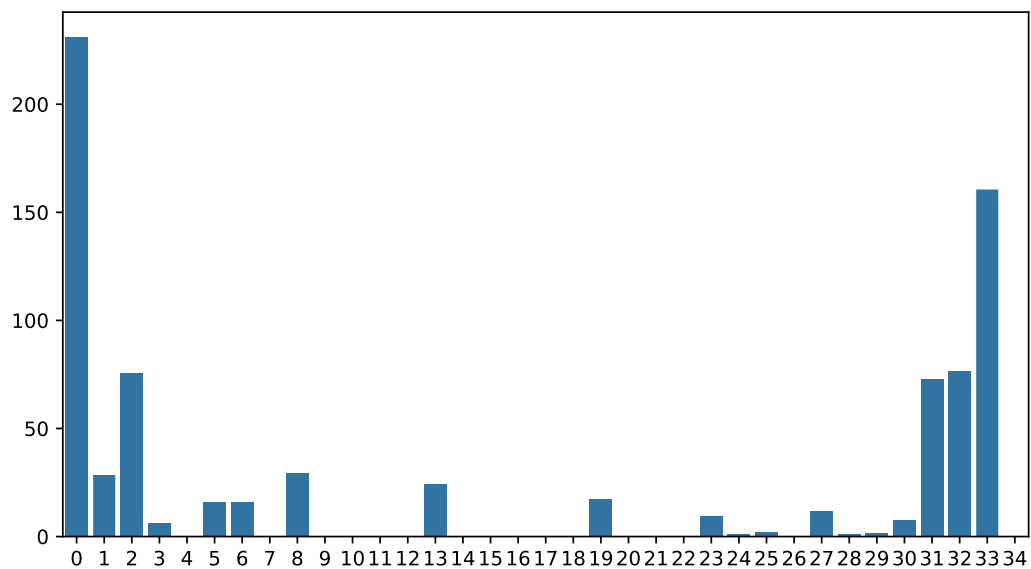
```

مصورسازی در gephi

در این مرحله فایل خروجی را در نرم‌افزار gephi باز می‌کنیم. سپس با استفاده از مقدار مرکزیت اندازه گره‌ها را تنظیم می‌کنیم. برای این تمرین به منظور مصورسازی بهتر، اندازه گره‌ها با توجه به مرکزیت آنها بین ۵ تا ۳۵ در نظر گرفته شده است. سپس از یک پالت رنگ برای رنگ آمیزی گراف بر اساس مرکزیت‌ها استفاده می‌کنیم. نتیجه نهایی در شکل صفحه بعد قابل مشاهده است.



همانطور که مشاهده می‌شود گره‌های با آیدی صفر و ۳۳ بیشترین مقدار مرکزیت بینابینی را دارند. همچنین در این گراف مقدار مرکزیت بینابینی برای اکثر گره‌ها یکسان است. شکل صفحه بعد نمودار توزیع مرکزیت بینابینی را نمایش می‌دهد.



اجتماع یابی و مقدار ماژولاریتی

در نهایت با استفاده از الگوریتم fastgreedy موجود در کتابخانه igraph عمل اجتماع یابی را انجام می دهیم. سپس مقدار ماژولاریتی را محاسبه می کنیم. مقدار ماژولاریتی بدست آمده برای این گراف ۰/۳۸۰۶۷۰۶۱۱۴۳۹۸۴۲۱۵ می باشد. قطعه کد زیر شامل برنامه کامل تمرین مذکور می باشد که مراحل اجتماع یابی و محاسبه مقدار ماژولاریتی نیز در انتهای آن آمده اند.

```

import igraph as ig
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import csv
import math

edges = []
# url : https://networks.skewed.de/net/karate
with open('edges.csv', 'r') as f:
    reader = csv.DictReader(f)
    for line in reader:
        edges.append((int(line['# source']), int(line[' target'])))

g = ig.Graph(35)
g.add_edges(edges)
betweenness = [0 if math.isnan(v) else v for v in g.betweenness()]
g.vs['betweenness'] = betweenness
#g.write_graphml('graph.graphml')
index = [i for i in range(0,35)]
sns.barplot(x=index, y=betweenness)
plt.show()
communities = g.community_fastgreedy()
print(communities.as_clustering().modularity)

```