



دانشکده مهندسی صنایع و سیستم ها
گروه فناوری اطلاعات
گرایش سامانه های شبکه ای

تحلیل شبکه های اطلاعات
تمرین ۵ - یک شبکه تصادفی

دانشجو:

محمد رضا صبی پور

استاد:

دکتر بابک تیمورپور

نیمسال دوم تحصیلی ۱۴۰۳-۱۴۰۴

اردیبهشت ۱۴۰۴

۱ شرح تمرین

توجه: در همه تمرینها در صورتی که از هوش مصنوعی استفاده می کنید نام هوش مصنوعی و درج کامل پرسش و پاسخها الزامی است و نوع سوالها نشان دهنده ادراک صحیح شما می باشد. البته به پاسخهای هوش مصنوعی نباید اکتفا کنید و باید اثبات ریاضی، کد یا توضیح مناسب را خودتان هم بیان کنید.

یک شبکه تصادفی با ۱۰۰۰ گره و احتمال برقراری رابطه $p=0.005$ یال ایجاد کنید. تعداد یالهای شبکه چند تا است؟ توزیع درجه را در مقیاس لگ-لگ ترسیم کنید. متوسط فواصل بین دو گره، قطر شبکه و ضریب خوشه‌پذیری را محاسبه کنید. مقدار تئوری این مقادیر را نیز (غیر از قطر) حساب کنید. آیا به هم نزدیکند.

شبکه را به فرمت لیست یال edge list (فقط به این فرمت پذیرفته است) در یک فایل txt ذخیره کنید و سپس در gephi بخوانید و مصور کنید.

فهرست مطالب

ب	۱ شرح تمرین.....
۲	۲ تشکیل گراف.....
۲	۳ تحلیل شبکه.....
۴	۴ تحلیل شبکه (دستی).....
۵	۵ مصورسازی در Gephi.....
۶	۶ استفاده از ChatGPT.....
۳	جدول ۱-۳ ویژگی‌های شبکه.....
۳	شکل ۱ نمودار توزیع درجه.....
۴	شکل ۲ توزیع درجه در مقیاس Log-Log.....
۶	شکل ۳ شبکه مصور شده به فرمت Force Atlas.....

۲ تشکیل گراف

ابتدا با استفاده از کتابخانه igraph در پایتون، یک گراف با ۱۰۰۰ رأس و احتمال تشکیل یال ۰.۰۰۵ می‌سازیم. این کار به کمک متد Erdos_Renyi انجام پذیر است.

```
import igraph as ig
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

g = ig.Graph.Erdos_Renyi(n=1000, p=0.005)
graph = {
    'number_of_edges': g.ecount(),
    'diameter': g.diameter(),
    'avg_path': g.average_path_length(),
    'clustering': g.transitivity_undirected()
}
print(graph)
degree_distribution = []
for v in g.vs():
    degree_distribution.append(v.degree())

sns.displot(degree_distribution, kde=True, color='orange')
plt.show()
```

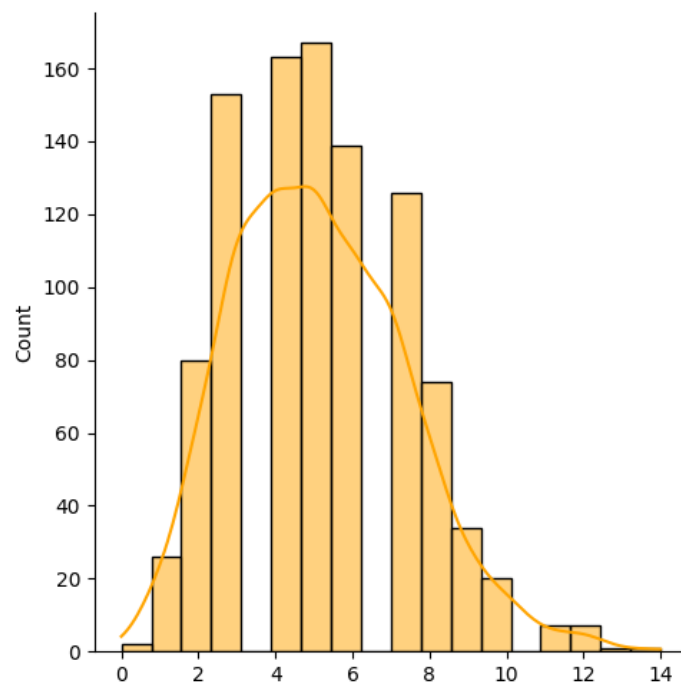
۳ تحلیل شبکه

پس از تشکیل شبکه، مقادیر خواسته شده از جمله میانگین طول مسیر، قطر شبکه، توزیع درجه، ضریب خوشه‌پذیری و تعداد یال‌ها را با استفاده از کتابخانه igraph محاسبه می‌کنیم. مقادیر به‌دست آمده در جدول ۳-۱ قابل مشاهده هستند.

جدول ۳-۱ ویژگی‌های شبکه

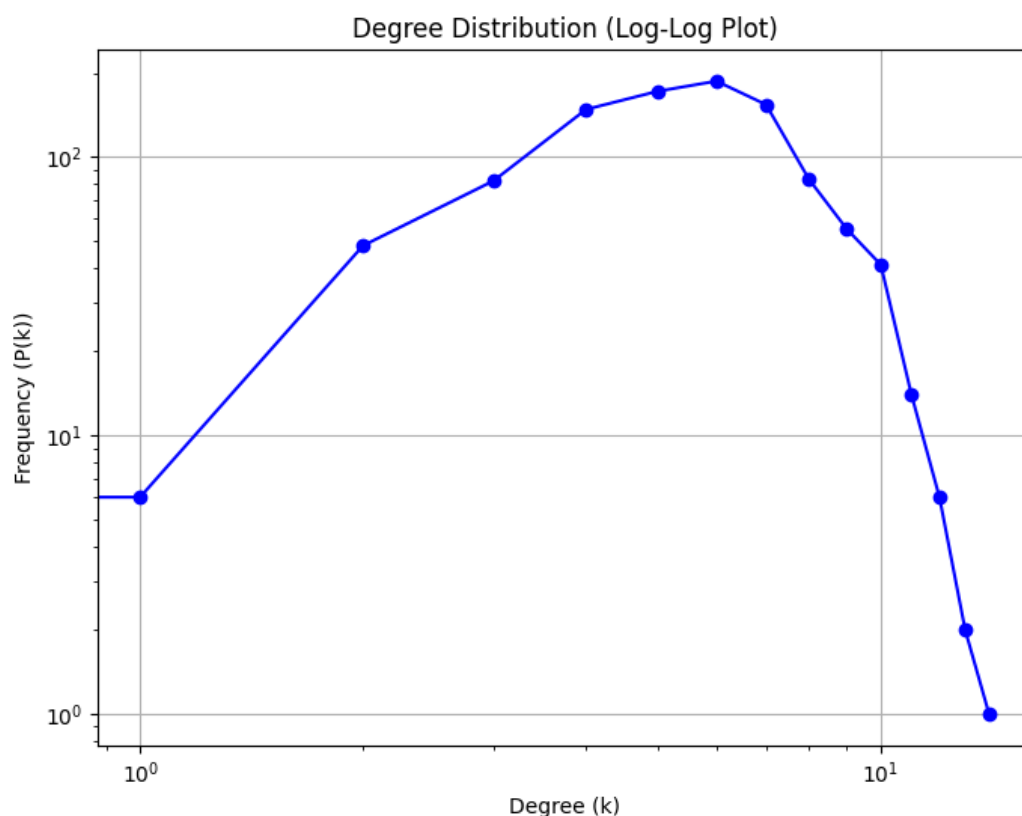
تعداد یال	۲۵۶۳
قطر شبکه	۹
میانگین طول مسیر	۴.۴۲
ضریب خوشه‌پذیری	۰.۰۰۳

در شکل ۱ نمودار توزیع درجات رسم شده است. بنظر می‌رسد که بیشتر گره‌ها دارای درجه‌های مشابه هستند و تنها تعدادی از گره‌ها با درجه بالا وجود دارند که در توزیع درجه نمودار به صورت قله‌های بالا و یا نقاط بیرون‌زن دیده می‌شوند.



شکل ۱ نمودار توزیع درجه

همچنین نمودار توزیع درجه در مقیاس Log-Log نیز در شکل ۲ رسم شده است.



شکل ۲ توزیع درجه در مقیاس Log-Log

۴ تحلیل شبکه (دستی)

برای محاسبه مقادیر خواسته شده به صورت دستی، از قطعه کد زیر استفاده شده است. در این حالت مقدار ضریب خوشه‌پذیری برابر با p خواهد بود و متوسط فواصل بین دو گره برابر ۴.۲۹ است. با توجه به مقادیر به دست آمده، مشاهده می‌شود که مقادیر محاسبه شده به صورت دستی بسیار نزدیک به مقادیر محاسبه شده توسط igraph هستند.

```

n = 1000
p = 0.005
expected_k = p * (n-1)
avg_distance_theoretical = math.log(n) / math.log(expected_k)
print(avg_distance_theoretical)

```

۵ مصورسازی در Gephi

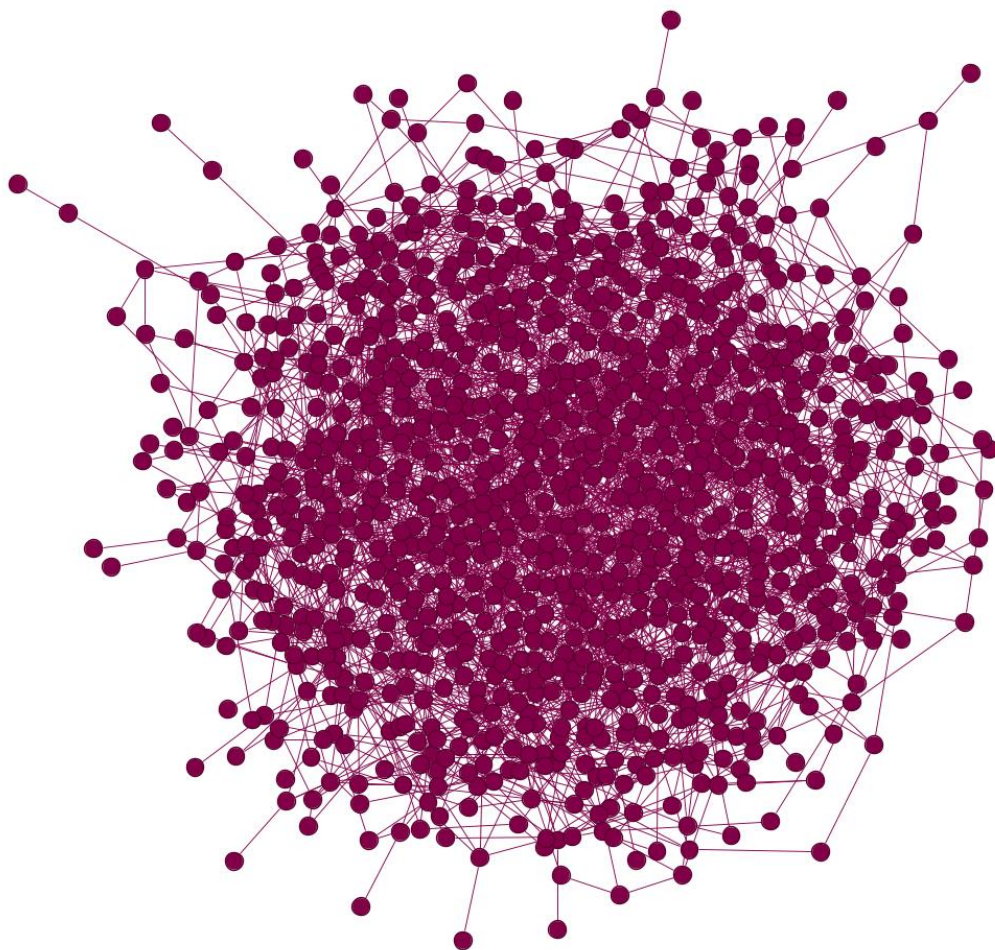
به منظور انتقال شبکه از igraph به Gephi، ابتدا یک لیست کمانه‌ای با فرمت txt می‌سازیم. لیست کمانه‌ای شامل دو ستون است که ستون اول شماره رأس مبدا و ستون دوم شامل شماره رأس مقصد است. مراحل استخراج شبکه تصادفی تولید شده به فرمت لیست کمانه‌ای در قطعه کد زیر قابل مشاهده است.

```

with open("graph_edges.txt", "w") as f:
    for e in g.es():
        f.write(f"{e.source}{e.target}\n")

```

سپس فایل ذخیره شده را در نرم افزار Gephi باز کرده و نوع گراف را undirected انتخاب می‌کنیم. در نهایت با اضافه کردن رنگ و انتخاب چیدمان مناسب (Force Atlas) گراف را ترسیم می‌کنیم. خروجی در شکل ۳ قابل مشاهده است.



شکل ۳ شبکه مصور شده به فرمت Force Atlas

۶ استفاده از ChatGPT

Draw the log log plot of degree distribution using matplotlib

Response:

Here's the Python code that you can run in your local environment:

Code to Generate Log-Log Plot of Degree Distribution:


```

import igraph as ig
import matplotlib.pyplot as plt
import numpy as np

# Create a random graph (Erdős-Rényi model)
g = ig.Graph.Erdos_Renyi(n=1000, p=0.005)

# Get the degree distribution
degree_distribution = g.vs.degree()

# Plot the degree distribution using log-log scale
plt.figure(figsize=(8, 6))
plt.loglog(np.unique(degree_distribution),
           np.bincount(degree_distribution)[np.unique(degree_distribution)-1],
           marker='o', linestyle='-', color='blue')

# Customize the plot
plt.xlabel('Degree (k)')
plt.ylabel('Frequency (P(k))')
plt.title('Degree Distribution (Log-Log Plot)')
plt.grid(True)
plt.show()

```

Steps to Run the Code:

Install Required Libraries:

You need to install igraph and matplotlib if you haven't done so already. You can do this using pip:

pip install igraph matplotlib

Copy and paste the code above into a Python script or Jupyter notebook and execute it.

Interpretation:

The plot will show the degree distribution of the network on a log-log scale, which is useful for identifying power-law behaviors in the network.

If you have any further questions or encounter any issues while running the code, feel free to reach out!