**گزارش تمرین تحلیل شبکه - CA-HepTh Collaboration Network فایل hw01-2**

**1 معرفی مجموعه‌داده**

CA-HepTh یک شبکه همکاری مبنی بر مقالات حوزه High Energy Physics در arXiv است که در دوره زمانی 1992 تا 2003 انتشار یافته است. هر گره معنی یک پژوهشگر را دارد و هر یال نشان‌دهنده همکاری در یک مقاله است.

* تعداد گره‌ها: 9877
* تعداد یال‌ها: 25998

**2 خواندن داده و ساخت گراف**

import networkx as nx

import gzip

G = nx.Graph() # ساخت گراف بدون جهت

# خواندن فایل فشرده ca-HepTh.txt.gz

with gzip.open('ca-HepTh.txt.gz', 'rt') as f:

for line in f:

if line.startswith('#'):

continue # رد کردن خطوط کامنت

u, v, \*\_ = line.strip().split()

G.add\_edge(int(u), int(v)) # اضافه کردن یال

print(f"تعداد گره‌ها: {G.number\_of\_nodes()}")

print(f"تعداد یال‌ها: {G.number\_of\_edges()}")

**خروجی:**

تعداد گره‌ها: 9877

تعداد یال‌ها: 25998

**3 محاسبه شاخص‌های گراف**

# بررسی همبند بودن و انتخاب بزرگترین کامپوننت

if not nx.is\_connected(G):

G = G.subgraph(max(nx.connected\_components(G), key=len)).copy()

# شاخصها

degrees = dict(G.degree())

max\_degree = max(degrees.values())

avg\_degree = sum(degrees.values()) / G.number\_of\_nodes()

avg\_path\_length = nx.average\_shortest\_path\_length(G)

diameter = nx.diameter(G)

avg\_clustering = nx.average\_clustering(G)

print("الف) مشخصات شبکه:")

print(f"- ماکزیمم درجه: {max\_degree}")

print(f"- متوسط درجه: {avg\_degree:.2f}")

print(f"- متوسط فاصله: {avg\_path\_length:.2f}")

print(f"- قطر شبکه: {diameter}")

print(f"- متوسط ضریب خوشه‌بندی: {avg\_clustering:.4f}")

**خروجی:**

- ماکزیمم درجه: 65

- متوسط درجه: 5.75

- متوسط فاصله: 5.95

- قطر شبکه: 18

- متوسط ضریب خوشه‌بندی: 0.4816

**4 رسم نمودار توزیع درجه**

import matplotlib.pyplot as plt

import numpy as np

degree\_values = list(degrees.values())

plt.figure(figsize=(12, 5))

# هیستوگرام عادی

plt.subplot(1, 2, 1)

plt.hist(degree\_values, bins=30, color='skyblue', edgecolor='black')

plt.title("Degree Distribution (Normal)")

plt.xlabel("Degree")

plt.ylabel("Count")

# log-log

plt.subplot(1, 2, 2)

degree\_count = np.bincount(degree\_values)

degrees\_list = np.arange(len(degree\_count))

plt.loglog(degrees\_list[degree\_count > 0], degree\_count[degree\_count > 0], marker='o', linestyle='None')

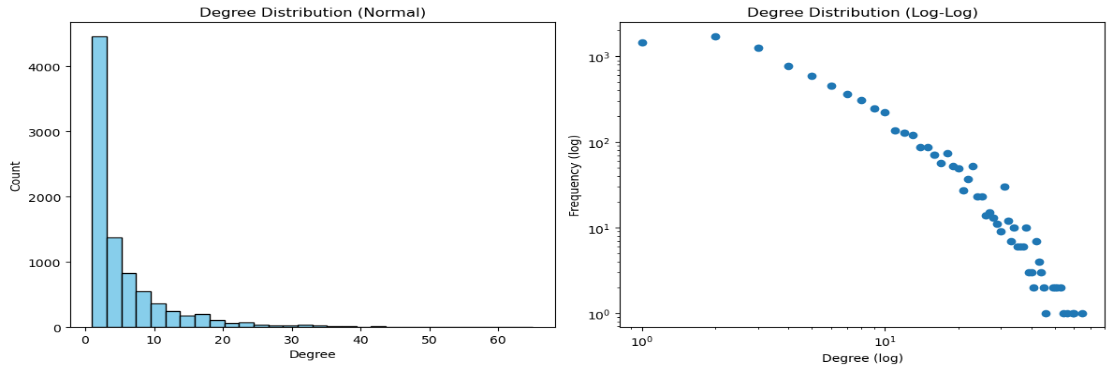
plt.title("Degree Distribution (Log-Log)")

plt.xlabel("Degree (log)")

plt.ylabel("Frequency (log)")

plt.tight\_layout()

plt.show()

**توضیح:** توزیع درجه در نمودار log-log معمولاً شکل Power Law دارد.

**5 شناسایی گره‌های با بالاترین درجه**

top\_nodes = sorted(degrees.items(), key=lambda x: x[1], reverse=True)[:5]

print("ج) 5 گره با بیشترین درجه:")

for node, deg in top\_nodes:

clust = nx.clustering(G, node)

print(f"گره {node} -> درجه: {deg}, ضریب خوشه‌بندی: {clust:.4f}")

**خروجی:**

گره 1441 -> درجه: 65, ضریب خوشه‌بندی: 0.0635

گره 19615 -> درجه: 60, ضریب خوشه‌بندی: 0.0661

گره 63113 -> درجه: 59, ضریب خوشه‌بندی: 0.0795

گره 30744 -> درجه: 56, ضریب خوشه‌بندی: 0.0506

گره 16164 -> درجه: 54, ضریب خوشه‌بندی: 0.0985

**جمع‌بندی**

این تمرین نمایش می‌دهد که بیشتر شبکه‌های همکاری علمی دارای خصوصیات شبکه‌های scale-free هستند (گره‌های با درجه بسیار بالا). همچنین ضریب خوشه‌بندی بالا نشان می‌دهد که همکاران گره احتمالاً خودشان هم با هم همکاری دارند.