

经济管理学院

课程报告

(复杂网络与社会计算)

题目: week2 课程作业

课程教师: 赵吉昌

学院/专业: 信息管理与信息系统

学生姓名: 范春

学 号: 21377061

2024 年 3 月 7 日



北京航空航天大学
BEIHANG UNIVERSITY

1、下载如下不同的网络数据，利用 `networkx` 进行初步的结构分析，如比较平均度，密度、聚集系数、度分布（包括幂指数）等的差异。结合实际的数据背景，分析这些差异的可能含义。

针对上述问题，我编写代码进行了相关计算，结果如下：

	as-skitter	HR-edges	HU-edges	RO-edgers
顶点数	1696415	54573	47538	41773
边数	11095298	498202	222887	125826
密度	7.710899e-06	0.00033457	0.00019726	0.00014422
平均度	13.080877	18.258186	9.377214	6.024274
聚集系数	0.258147	0.136477	0.116187	0.091212
幂律分布系数 γ	2.342975	4.717437	4.926857	4.270884
幂律分布系数 p	0.009492	0.114942	0.091595	0.089588

其度分布及幂律分布图分别如下所示：

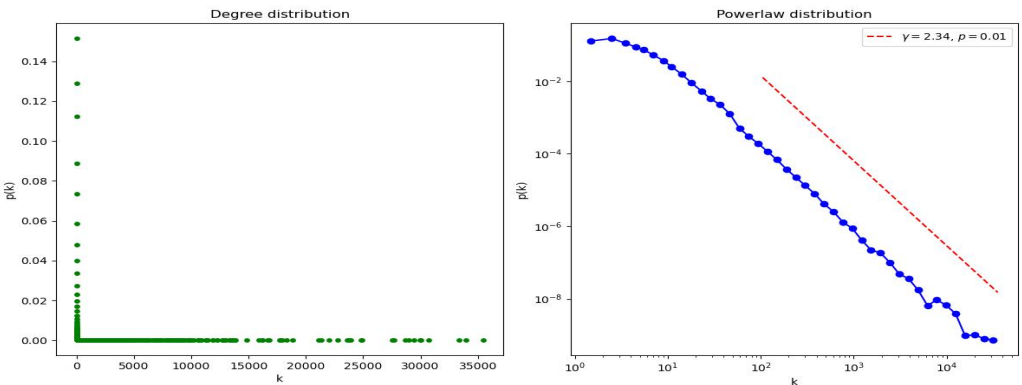


图 1 as-skitter

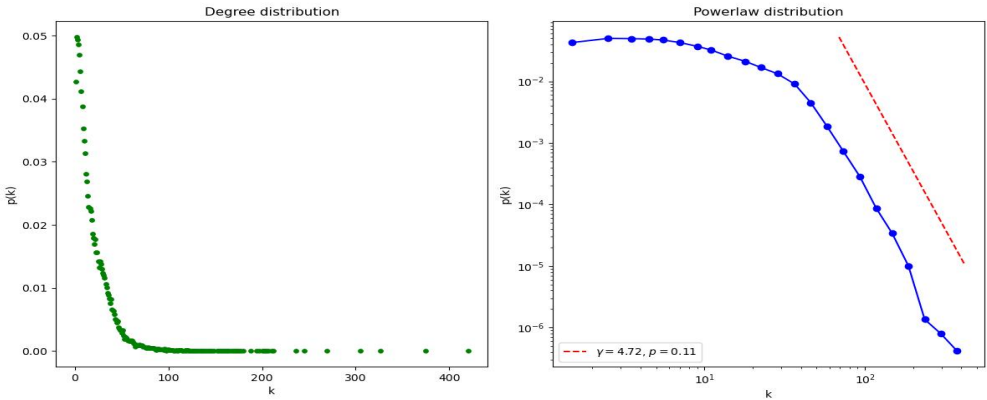


图 2 HR-edges

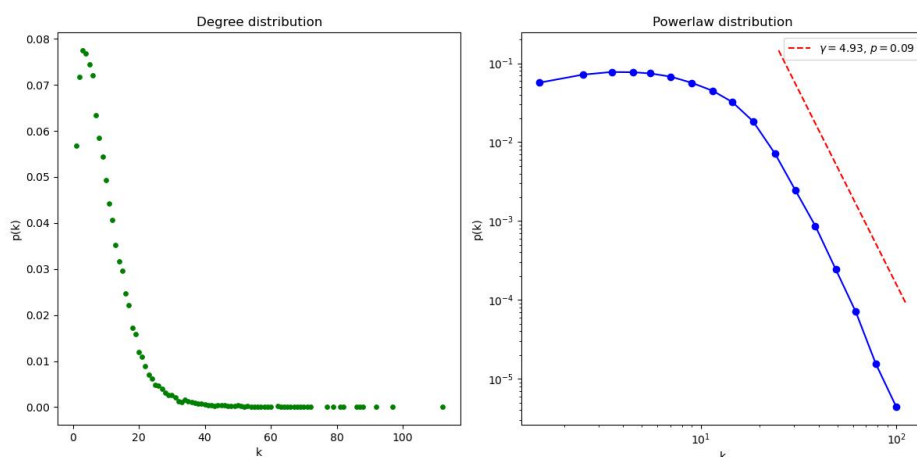


图 3 HU-edges

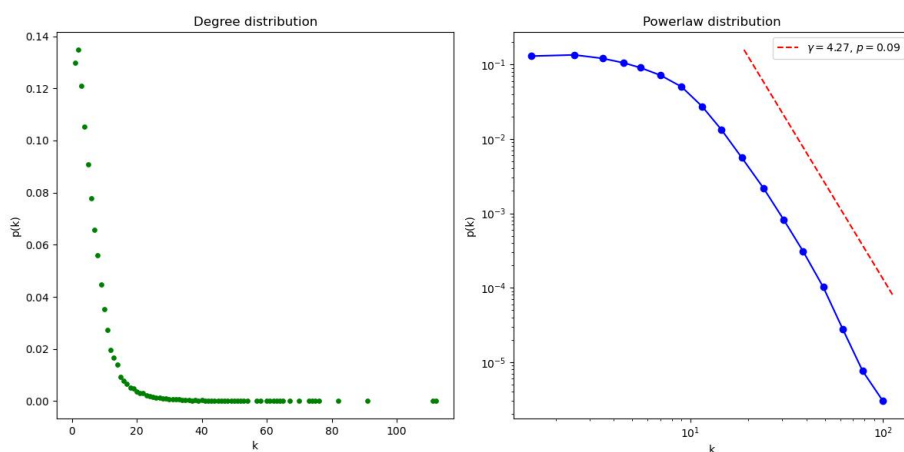


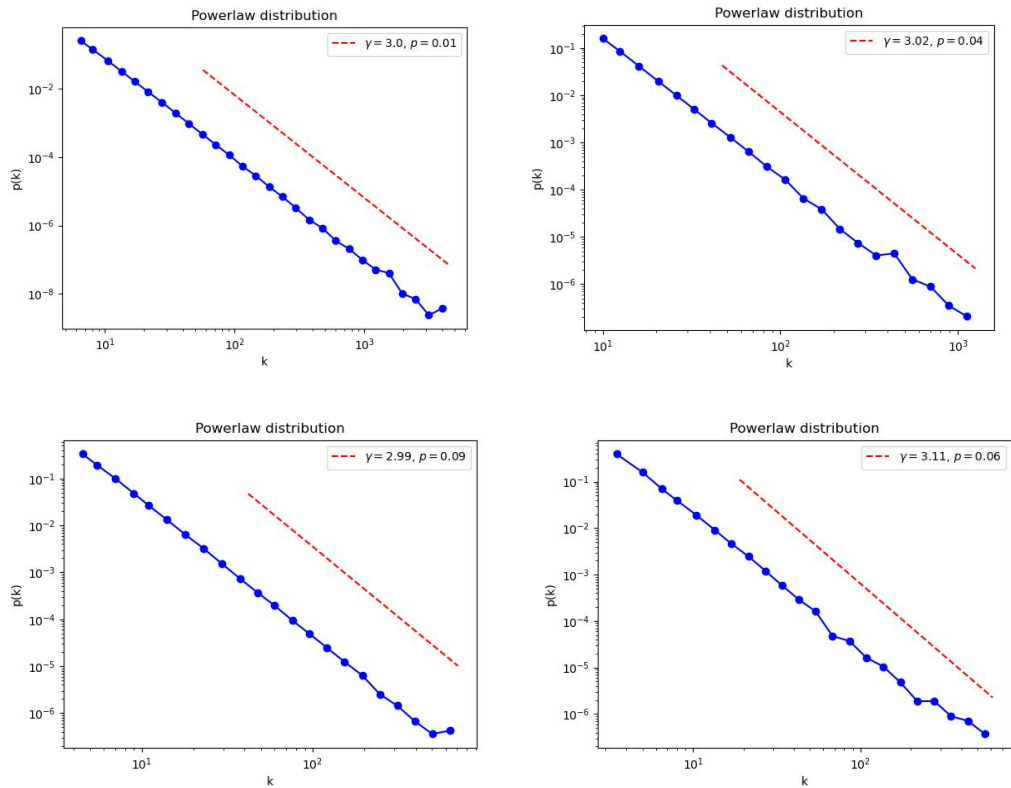
图 4 RO-edges

由上表及上图可知，as-skitter 对应的网络规模最大，但其密度最小，说明该网络图的稠密情况最低，而它的聚集系数是最大的，表明该网络节点之间的联系最为紧密。在这四个网络中，聚集系数都随着节点数和边数的减小而减小，幂律分布系数 γ ，即幂律指数都比较接近 3，但与 3 的差距还是有点大。

2、参照 1 中的文献，用 BA 模型生成对应规模的网络，并比较其与上述实际网络的异同。

	as-skitter	HR-edges	HU-edges	RO-edges
顶点数	1696415	54573	47538	41773
边数	10178454	491076	190136	125310
密度	7.0737e-06	0.00032979	0.00016828	0.00014363
平均度	11.999958	17.9970315	7.99932685	5.9995691
聚集系数	0.000113	0.00270579	0.00174287	0.00173006

幂律分布系数 γ	3.003983	3.015271	2.98981	3.105795
幂律分布系数 p	0.013665	0.042772	0.085156	0.05772



经过对比发现，用 BA 模型生成的对应规模的网络的聚集系数远小于实际网络，另外模拟生成的网络的幂律指数比实际网络更接近 3，满足无标度性。

3、尝试计算上述网络的平均最短路径。如果 `networkx` 太慢，有无别的方式可以进行相对快速的可靠估计？

计算出了部分数据的平均最短路径，结果如下：

HR-edges	HU-edges	RO-edgers
4.506565884550811	5.3409423368686895	6.348893165004872

由于 `networkx` 太慢，所以并未计算文本文件数据的平均最短路径。由上述计算结果可以初步判定，网络规模越大，其平均最短路径反而越小，具备小世界效应。针对计算速度较慢的问题，我们可以采用 Monte Carlo 方法。该方法的具体步骤如下：

- (1) 选择一个起始节点，然后从该节点开始进行随机游走；
- (2) 在每一步中，随机选择一个邻居节点进行移动；
- (3) 重复上述步骤多次，记录每次游走所经过的节点和路径长度；

(4) 对于每对节点，计算它们之间的平均距离，并最终计算整个网络的平均最短路径长度。

具体代码如下：

```
def monte_carlo_shortest_path_length(G, num_samples=1000):
    total_length = 0
    count = 0
    for _ in range(num_samples):
        node1, node2 = random.sample(list(G.nodes), 2)
        try:
            length = nx.shortest_path_length(G, node1, node2)
            total_length += length
            count += 1
        except nx.NetworkXNoPath:
            continue
    avg_length = total_length / count if count > 0 else 0
    return avg_length
```

估计结果如下，与实际计算结果非常接近，估计效果良好。

```
PS C:\Users\范春> & C:/Anaconda3/python.exe c:/Users/范春/Desktop/1.py
Estimated average shortest path length: 4.518
HR_edges is ok!
Estimated average shortest path length: 6.343
RO_edges is ok!
Estimated average shortest path length: 5.307
HU_edges is ok!
```