

SI 与 SIR 模型仿真 (Barabási-Albert 随机网络)

根据作业要求，在 Barabási-Albert 随机网络上分别运行 SI 和 SIR 模型，并绘制感染比例随时间的变化曲线。

参数：N=1000, m=20, beta=0.005, gamma=0.01, 初始感染者=1

```
In [2]: # 安装必要的库（如果尚未安装）
import sys
```

```
In [3]: import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
import ndlib.models.epidemics as ep
import ndlib.models.ModelConfig as mc

# 1. 设置参数
N = 1000
m = 20
beta = 0.005
gamma = 0.01
steps = 200 # 增加步数以确保观察到完整过程

# 2. 生成 Barabási-Albert 随机网络
# seed=42 保证每次生成的图一致，方便复现
G = nx.barabasi_albert_graph(N, m, seed=42)
print(f"网络生成完毕：节点数={G.number_of_nodes()}，边数={G.number_of_edges()}")
```

ChatGPT

网络生成完毕：节点数=1000, 边数=19600

In [4]: # 3. SI 模型仿真

```
model_si = ep.SIModel(G)
cfg_si = mc.Configuration()
cfg_si.add_model_parameter('beta', beta)
cfg_si.add_model_initial_configuration("Infected", [0]) # 初始感染节点 0
model_si.set_initial_status(cfg_si)

# 运行仿真
iterations_si = model_si.iteration_bunch(steps)
trends_si = model_si.build_trends(iterations_si)

# 提取数据
# trends_si[0]['trends']['node_count'][状态] 是一个列表
# 注意: ndlib 的某些版本返回的键可能是整数 0, 1 而不是字符串 '0', '1'
node_counts = trends_si[0]['trends']['node_count']

# 尝试获取感染数据 (状态 1)
if 1 in node_counts:
    si_infected_data = node_counts[1]
elif '1' in node_counts:
    si_infected_data = node_counts['1']
else:
    # 如果都找不到, 打印键值以便调试, 并设为全0
    print(f"Error: Key 1 not found. Available keys: {list(node_counts.keys())}")
    si_infected_data = [0] * len(iterations_si)

si_t = range(len(si_infected_data))

# 归一化
si_infected_ratio = [x / N for x in si_infected_data]

print("SI 模型仿真完成")
```

ChatGPT

SI 模型仿真完成

In [5]: # 4. SIR 模型仿真

```
model_sir = ep.SIRModel(G)
cfg_sir = mc.Configuration()
cfg_sir.add_model_parameter('beta', beta)
cfg_sir.add_model_parameter('gamma', gamma)
cfg_sir.add_model_initial_configuration("Infected", [0]) # 初始感染节点 0
model_sir.set_initial_status(cfg_sir)

# 运行仿真
iterations_sir = model_sir.iteration_bunch(steps)
trends_sir = model_sir.build_trends(iterations_sir)

# 提取数据
# '0': Susceptible, '1': Infected, '2': Removed
node_counts = trends_sir[0]['trends']['node_count']

def get_data(nc, key_int, key_str):
    if key_int in nc:
        return nc[key_int]
    elif key_str in nc:
        return nc[key_str]
    else:
        return [0] * len(iterations_sir)

sir_susceptible_data = get_data(node_counts, 0, '0')
sir_infected_data = get_data(node_counts, 1, '1')
sir_removed_data = get_data(node_counts, 2, '2')

# 归一化
sir_susceptible_ratio = [x / N for x in sir_susceptible_data]
sir_infected_ratio = [x / N for x in sir_infected_data]
sir_removed_ratio = [x / N for x in sir_removed_data]

print("SIR 模型仿真完成")
```

ChatGPT

SIR 模型仿真完成

```
In [6]: # 5. 绘图
plt.figure(figsize=(12, 5))

# 子图 1: SI 模型
plt.subplot(1, 2, 1)
plt.plot(si_t, si_infected_ratio, label='Infected (SI)', color='red')
plt.title('SI Model on BA Graph')
plt.xlabel('Time Step')
plt.ylabel('Fraction of Population')
plt.ylim(-0.05, 1.05)
plt.grid(True)
plt.legend()

# 子图 2: SIR 模型
plt.subplot(1, 2, 2)
# 注意: SIR 的时间步可能和 SI 不完全一样 (虽然都是 steps, 但 build_trends 处理后长度一致)
sir_t = range(len(sir_infected_ratio))
plt.plot(sir_t, sir_susceptible_ratio, label='Susceptible', color='blue', linestyle='--')
plt.plot(sir_t, sir_infected_ratio, label='Infected', color='red')
plt.plot(sir_t, sir_removed_ratio, label='Removed', color='green', linestyle='-.')
plt.title('SIR Model on BA Graph')
plt.xlabel('Time Step')
plt.ylabel('Fraction of Population')
plt.ylim(-0.05, 1.05)
plt.grid(True)
plt.legend()

plt.tight_layout()
plt.show()
```

