

（2023-2024学年 第2学期）

网络科学基础实验报告

**题 目 网络模型及统计特性分析**

**所在学院 自动化学院、人工智能学院**

**专 业 人工智能**

**年级班级 B210416**

**学 号 B21080526**

**姓 名 单家俊**

**授课教师 宋玉蓉**

2024 年 6月 10 日

## 一、实验目的

本实验旨在研究和对比ER随机图模型、WS小世界模型和BA无标度模型的特性。通过生成和分析这些网络模型，了解它们的平均路径长度、度分布和聚类系数等统计特征，掌握它们在不同应用场景中的适用性。

## 二、实验内容

1、建立ER、WS和BA网路模型生成算法

2、对比分析这三种模型的平均路径长度，度分布和聚类系数等统计特征

3、对BA模型进行适当修改，使其更加符合现实网络统计特性，如：生成幂律值在2-3之间，聚类系数在0.1以上的无标度网络。

4、生成一个随机几何图，分析其网络统计特性

## 三、实验结果

1、ER网络生成算法

**import** networkx **as** nx  
**import** matplotlib.pyplot **as** plt  
*# 生成一个含有20个节点、以概率p = 0.2连接的ER随机图：*G = nx.erdos\_renyi\_graph(20, 0.2)  
*# shell布局*pos = nx.shell\_layout(G)  
*# 画出网络模型*nx.draw(G, pos, with\_labels=**False**, node\_size=30)  
*# 获取各节点与相应度值的字典*degree = dict(nx.degree(G))  
*# 平均度为所有节点度之和除以总节点数*average\_degree = sum(degree.values()) / len(G)  
print(**"平均度为："**, average\_degree)  
*# 最短路径长度***try**:  
 average\_shortest\_path\_length = nx.average\_shortest\_path\_length(G)  
 print(**"最短路径长度为："**, average\_shortest\_path\_length)  
**except** nx.NetworkXError **as** e:  
 print(**"无法计算最短路径长度:"**, e)  
*# 平均集类系数*average\_clustering = nx.average\_clustering(G)  
print(**"平均聚类系数为："**, average\_clustering)  
*# 获取度分布, 返回所有位于区间[0, dmax]的度值的频数列表*degree\_distribution = nx.degree\_histogram(G)  
x = range(len(degree\_distribution)) *# 生成X轴序列，从0到最大度*y = [z / float(sum(degree\_distribution)) **for** z **in** degree\_distribution] *# 将频次转化为频率  
# 绘制度分布图*plt.figure(figsize=(5.8, 5.2), dpi=150) *# 调整显示参数*plt.rcParams[**'font.sans-serif'**] = [**'SimHei'**] *# 用来正常显示中文标签*plt.xlabel(**"Degree"**, size=14) *# Degree*plt.ylabel(**"Frequency"**, size=14) *# Frequency*plt.xticks(fontproperties=**'Times New Roman'**, size=13) *# 字体样式*plt.yticks(fontproperties=**'Times New Roman'**, size=13) *# 字体样式*plt.plot(x, y) *# 折线图*plt.show() *# 显示图像*

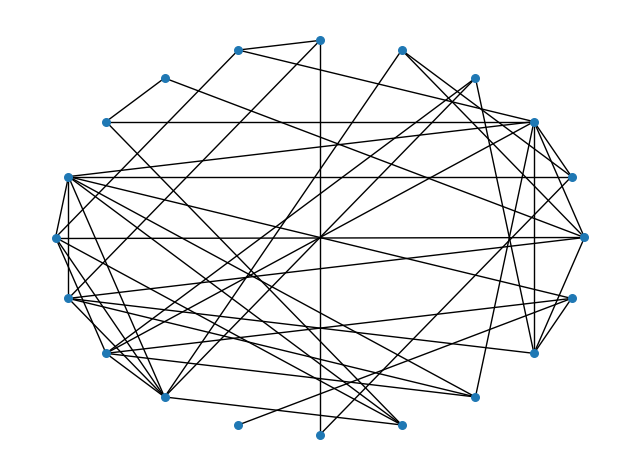
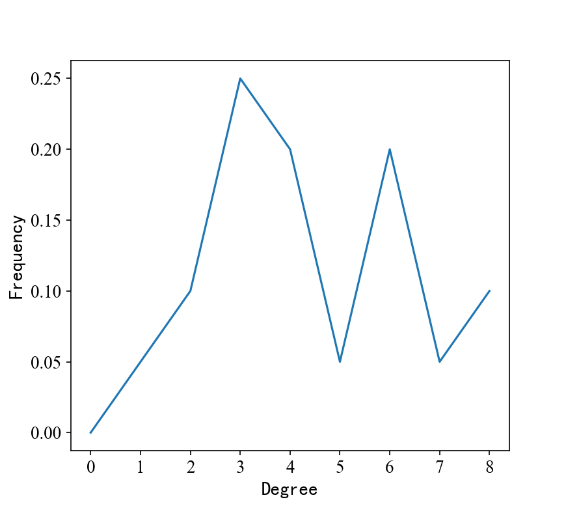
WS网络生成算法：

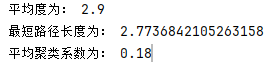
**import** networkx **as** nx  
**import** matplotlib.pyplot **as** plt  
*# 生成一个含有20个节点、每个节点有4个邻居、以概率p=0.3随机化重连边的WS小世界网络*G = nx.random\_graphs.watts\_strogatz\_graph(20, 4, 0.3)  
*# circular布局*pos = nx.circular\_layout(G)  
nx.draw(G, pos, with\_labels = **False**, node\_size = 30)  
*# 获取各节点与相应度值的字典*degree = dict(nx.degree(G))  
*# 平均度为所有节点度之和除以总节点数*print(**"平均度为："**, sum(degree.values())/len(G))  
*# 最短路径长度*print(**"最短路径长度为："**,nx.average\_shortest\_path\_length(G))  
*# 平均集类系数*print(**"平均聚类系数为："**,nx.average\_clustering(G))  
*# 获取度分布,返回所有位于区间[0, dmax]的度值的频数列表*degreeDis = nx.degree\_histogram(G)  
x = range(len(degreeDis)) *# 生成X轴序列，从1到最大度*y = [z / float(sum(degreeDis)) **for** z **in** degreeDis] *# 将频次转化为频率*plt.figure(figsize=(5.8, 5.2), dpi=150) *# 调整显示参数*plt.rcParams[**'font.sans-serif'**] = [**'SimHei'**] *# 用来正常显示中文标签*plt.xlabel(**"Degree"**, size=14) *# Degree*plt.ylabel(**"Frequency"**, size=14) *# Frequency*plt.xticks(fontproperties=**'Times New Roman'**, size=13) *# 字体样式*plt.yticks(fontproperties=**'Times New Roman'**, size=13) *# 字体样式*plt.plot(x,y) *# 折线图*plt.show() *# 显示图像*

BA网络生成算法：

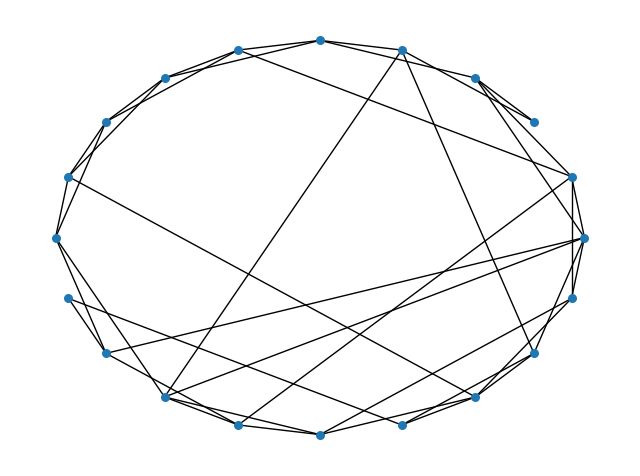
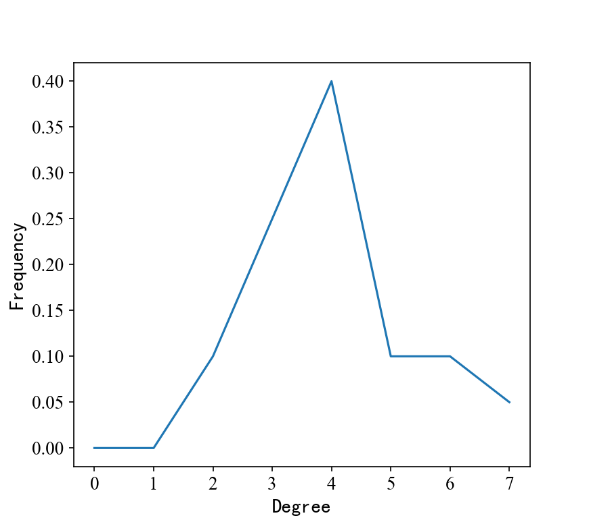
**import** networkx **as** nx  
**import** matplotlib.pyplot **as** plt  
*# 生成一个含有20个节点、每次加入1条边的BA无标度网络。*G = nx.random\_graphs.barabasi\_albert\_graph(20, 1)  
*# spring 布局*pos = nx.spring\_layout(G)  
nx.draw(G, pos, with\_labels = **False**, node\_size = 30)  
*# 获取各节点与相应度值的字典*degree = dict(nx.degree(G))  
*# 平均度为所有节点度之和除以总节点数*print(**"平均度为："**, sum(degree.values())/len(G))  
*# 最短路径长度*print(**"最短路径长度为："**,nx.average\_shortest\_path\_length(G))  
*# 平均集类系数*print(**"平均聚类系数为："**,nx.average\_clustering(G))  
*# 获取度分布,返回所有位于区间[0, dmax]的度值的频数列表*degreeDis = nx.degree\_histogram(G)  
x = range(len(degreeDis)) *# 生成X轴序列，从1到最大度*y = [z / float(sum(degreeDis)) **for** z **in** degreeDis] *# 将频次转化为频率*plt.figure(figsize=(5.8, 5.2), dpi=150) *# 调整显示参数*plt.rcParams[**'font.sans-serif'**] = [**'SimHei'**] *# 用来正常显示中文标签*plt.xlabel(**"Degree"**, size=14) *# Degree*plt.ylabel(**"Frequency"**, size=14) *# Frequency*plt.xticks(fontproperties=**'Times New Roman'**, size=13) *# 字体样式*plt.yticks(fontproperties=**'Times New Roman'**, size=13) *# 字体样式*plt.plot(x,y) *# 折线图*plt.show() *# 显示图像*

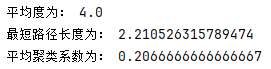
2、ER网络：

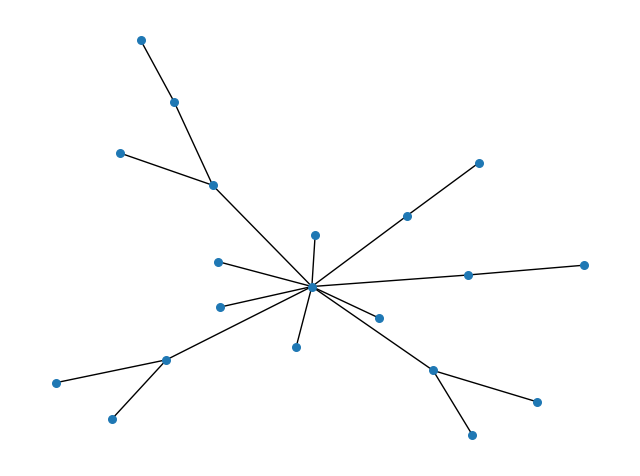
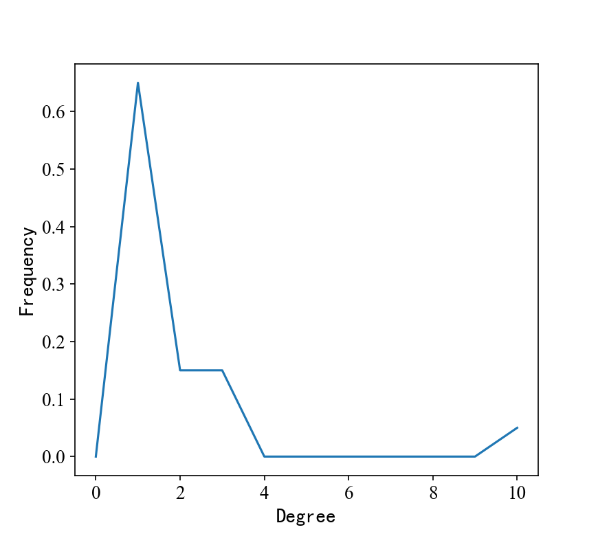


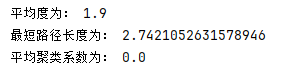
WS网络：



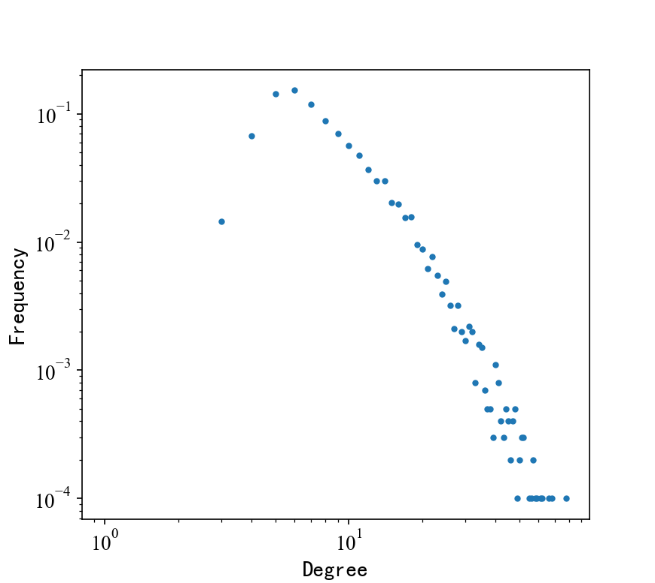
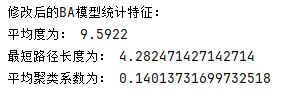
BA网络：

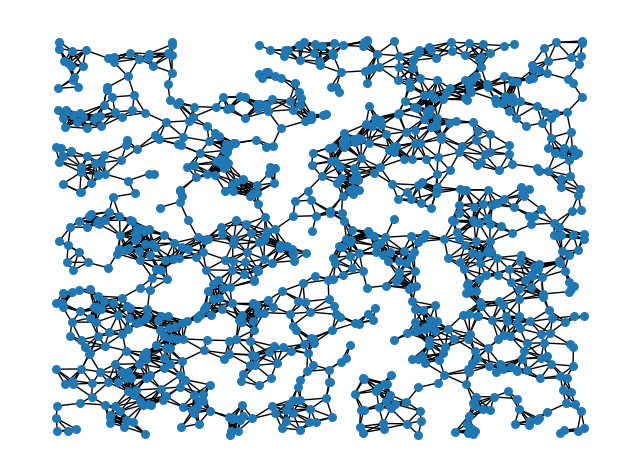
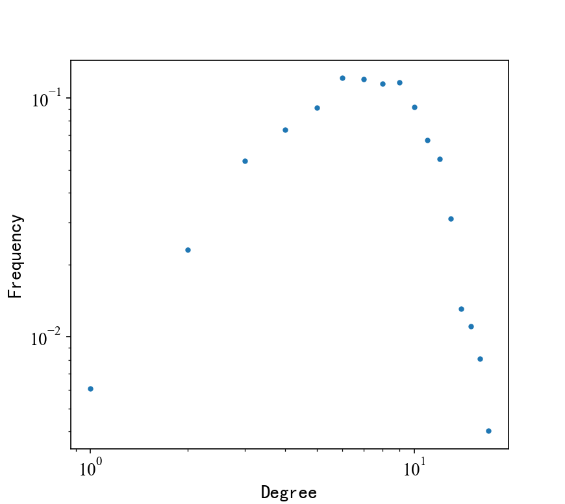


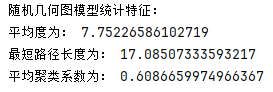
3、修改BA模型：

通过引入三角形闭合使BA模型更符合现实网络的统计特性。

4、随机几何图



## 四、实验心得

通过本次实验，我深入理解了ER、WS和BA三种经典网络模型的特性和区别。ER模型的随机性使其度分布接近泊松分布，网络较为均匀；WS模型通过重连边，兼具较短的平均路径长度和较高的聚类系数，体现了小世界特性；BA模型则通过优先连接机制，生成了度分布符合幂律分布的无标度网络。

在生成和分析这些模型的过程中，我掌握了网络生成方法和特性计算方法。ER模型虽然简单，但适用于分析随机网络。WS模型在模拟社交网络等实际场景中非常有用，因为它能同时满足局部集群和全局小世界特性。BA模型适用于描述互联网等大规模无标度网络。

为了使BA模型更符合现实网络的统计特性，我引入了三角形闭合机制，显著提高了其聚类系数，使其更加真实。这一改进不仅提升了模型的准确性，也增强了我对复杂网络结构的理解。

总之，通过这次实验，我不仅掌握了网络模型的基础知识，还提高了分析和改进网络模型的能力。这些知识对理解现实中的复杂网络具有重要意义。