

# “网络科学基础”-第一次上机报告

班级： 物联网 2303

姓名： 邱佳亮

学号： 3230611072

上机日期： 2024.11.1,第9周周五下午七八节课

2024 秋-网络科学基础（物联网 23）-第一次上机报告提交

## 一、上机题目

用 Matlab 编程实现相关网络参数的计算，输出相关信息。

## 二、上机目的

1. 熟悉、掌握程序语句的功能。

## 三、功能描述、上机程序（含必要的注释）、上机调试运行结果

### 1.功能描述

计算网络的度分布和平均度、联合度分布、各节点的最临近平均值  $k$ ，该网络是否是同配网络、该网络是否是正相关网络、计算该网络的网络直径  $D$  和平均路径长度  $L$ 。

### 2.上机程序

```
1. n=6; % 定义图的顶点数为 6
2. a=zeros(n); % 初始化一个 n*n 的邻接矩阵 a，所有元素为 0
3. a(1,[2,3,5])=1; % 将顶点 1 与顶点 2、3、5 相连
4. a(2,[3,4])=1; % 将顶点 2 与顶点 3、4 相连
5. a(3,6)=1; % 将顶点 3 与顶点 6 相连
6. a(4,6)=1; % 将顶点 4 与顶点 6 相连
7. a=a+a' % 将邻接矩阵 a 与其转置相加，得到无向图的邻接矩阵
8. d=sum(a) % 计算每个顶点的度（与该顶点相连的边数）
9. degree_range=minmax(d) % 获取度的范围（最小值和最大值）
10. ud=[degree_range(1):degree_range(2)]; % 创建一个包含所有可能度的向量
11. pinshu=hist(d,ud) % 计算每个度的频数
12. df=[ud;pinshu/n] % 创建一个频率分布表，其中第二列是每个度的频率（频数除以顶点数）
13. ave_degree=mean(d) % 计算平均度
14. M=sum(d)/2 % 计算总边数
15. [i,j]=find(triu(a)) % 查找邻接矩阵上三角的元素的行标和列标
16. ki=d(i);kj=d(j); % 获取起点和终点的度
17. kij=[ki',kj'];kij=sort(kij,2) % 将起点和终点的度组合在一起，并按列排序
18. bpin=[]; % 初始化联合度分布的向量
```

```

19. for i=1:length(ud)
20.     for j=i:degree_range(2)
21.         b(i,j)=0;kk=[];
22.         for k=1:size(kij,1)
23.             b(i,j)=b(i,j)+length(findstr(kij(k,:),[i,j]));
% 计算度为 i 和 j 的顶点对的数量
24.             if length(findstr(kij(k,:),[i,j]))
25.                 kk=[kk,k]; % 如果找到度为 i 和 j 的顶点对, 则记录
其索引
26.             end
27.         end
28.         kij(kk,:)=[]; % 从 kij 中移除已经计算过的顶点对
29.         bpin=[bpin,[i;j;b(i,j)/M]]; % 将度为 i 和 j 的顶点对的频
率添加到 bpin 中
30.     end
31. end
32. bpin % 输出联合度分布
33. knni=d*a./d % 计算最临近平均度值
34. for i=1:length(ud)
35.     ind=(d==ud(i));
36.     knn(i)=sum(knni(ind)/pinshu(i)); % 计算每个度的 knn 值
37. end
38. knn=[ud;knn] % 创建 knn 值的表
39.
40. function r=mycorrelations(a)
41.     d=sum(a); % 计算每个顶点的度
42.     M=sum(d)/2; % 计算总边数
43.     [i,j]=find(triu(a)); % 查找邻接矩阵上三角的元素的行标和列标
44.     ki=d(i);kj=d(j); % 获取起点和终点的度
45.     r=(ki*kj'/M-(sum(ki+kj)/2/M)^2)/... % 计算并返回相关系数
46.         (sum(ki.^2+kj.^2)/2/M-(sum(ki+kj)/2/M)^2);
47. end
48.
49. function[D,L,dist] =myAPL(a)
50.     A=graph(a) % 根据邻接矩阵 a 创建图对象
51.     dist=distances(A); % 计算图中所有顶点之间的距离
52.     D=max(max(dist)); % 计算直径 (最长的最短路径)
53.     Ldist=tril(dist); % 获取距离矩阵的下三角部分
54.     he=sum(nonzeros(Ldist)); % 计算非零元素的和
55.     n=length(a); % 获取顶点数
56.     L=he/nchoosek(n,2); % 计算平均路径长度
57. end

```

### 3.上机调试及运行结果

```
n=6;
a=zeros(n);
a(1,[2,3,5])=1;
a(2,[3,4])=1;
a(3,6)=1;
a(4,6)=1;
a=a+a'
```

```
a = 6x6
    0     1     1     0     1     0
    1     0     1     1     0     0
    1     1     0     0     0     1
    0     1     0     0     0     1
    1     0     0     0     0     0
    0     0     1     1     0     0
```

```
d=sum(a) %度
```

```
d = 1x6
     3     3     3     2     1     2
```

图 1 度分布

```
df=[ud;pinshu/n] %频率分布表
```

```
df = 2x3
    1.0000    2.0000    3.0000
    0.1667    0.3333    0.5000
```

图 2 频率分布

```
ave_degree=mean(d) %平均度
```

```
ave_degree = 2.3333
```

图 3 平均度

```
bpin=[]; %联合度分布
for i=1:length(ud)
    for j=i:degree_range(2)
        b(i,j)=0;kk=[];
        for k=1:size(kij,1)
            b(i,j)=b(i,j)+length(findstr(kij(k,:),[i,j]));
            if length(findstr(kij(k,:),[i,j]))
                kk=[kk,k];
            end
        end
        kij(kk,:)=[];
        bpin=[bpin,[i;j;b(i,j)/M]];
    end
end
bpin %联合度分布
```

```
bpin = 3x6
    1.0000    1.0000    1.0000    2.0000    2.0000    3.0000
    1.0000    2.0000    3.0000    2.0000    3.0000    3.0000
         0         0    0.1429    0.1429    0.2857    0.4286
```

图 4 联合度分布

```
knni=d*a./d %最临近平均值
```

```
knni = 1x6
    2.3333    2.6667    2.6667    2.5000    3.0000    2.5000
```

```
for i=1:length(ud)
    ind=(d==ud(i));
    knn(i)=sum(knni(ind)/pinshu(i));
end
knn=[ud;knn]
```

```
knn = 4x3
    1.0000    2.0000    3.0000
    3.0000    2.0000    3.0000
    2.5000    2.5556    3.0000
    2.5556    2.5000    2.5556
```

图 5 最临近平均值

```
r=mycorrelations(a)

r = -0.1053
```

图 6 计算相关性

```
[D,L,dist]=myAPL(a)

A =
graph - 属性:
    Edges: [7×2 table]
    Nodes: [6×0 table]
D = 3
L = 1.6667
dist = 6×6
    0    1    1    2    1    2
    1    0    1    1    2    2
    1    1    0    2    2    1
    2    1    2    0    3    1
    1    2    2    3    0    3
    2    2    1    1    3    0
```

图 7 网络直径和平均路径长度

#### 四、上机总结及感想

在本次实验中，我使用 Matlab 编写了一个程序来计算给定网络的网络直径和平均路径长度。这包括了构建邻接矩阵、计算度分布、最临近平均值，以及利用自定义函数来计算相关系数和网络的平均路径长度。首先，我根据给定的网络结构，构建了相应的邻接矩阵。接着，计算了每个顶点的度，并据此生成了度分布表。编写了自定义函数 mycorrelations 来计算网络的相关系数，以及 myAPL 函数来计算网络的平均路径长度和直径。在实验过程中，我遇到了一些挑战，比如如何正确使用 Matlab 的图论工具，以及如何优化代码以提高计算效率。但通过查阅文档和在线资源，我逐步解决了这些问题，并成功完成了程序的编写。这个过程让我认识到了在遇到问题时，耐心和持续学习的重要性。这次实验不仅提高了我的编程技能，还增强了我的问题解决能力，让我对网络科学有了更深入的理解。我期待将这些新学到的技能应用到未来的学习和研究中。