

多智能体系统实验报告

专业：	人工智能
学号：	58119125
年级：	19 级
姓名：	蒋卓洋

签名：蒋卓洋

时间：2022/6/13

一、实验选题与问题描述

- 1.1 实现全局耦合网络，展示网络图，并计算平均路径长度和聚类系数。
- 1.2 实现最近邻耦合网络，展示网络图，并计算平均路径长度和聚类系数。
- 1.3 实现星形网络，展示网络图，并计算平均路径长度和聚类系数。
- 1.4 实现随机网络，展示网络图，并计算平均路径长度和聚类系数。
- 2.1 实现并展示 WS 小世界模型构建，并计算平均路径长度和聚类系数。
- 2.2 实现并展示 NW 小世界模型构建，并计算平均路径长度和聚类系数。

二、实现步骤与流程

1. 统一的基础功能实现：

1.1. 选定存储网络图的数据结构

- (1) 选定邻接矩阵存储的无向图作为网络图的基础数据结构。
- (2) 所有网络运算操作都基于邻接矩阵实现。

1.2. 编写统一的网络图可视化方法

- (1) 选择让结点以相等的间隔角度以原点为中心，排列在相同半径的圆周上，故按照结点个数等分圆周得到间隔角度：

$$\theta_0 = \frac{2\pi}{num}$$

- (2) 用间隔角度与直角坐标的对应关系，分配结点在图上的坐标，对应关系如下：

$$\begin{cases} y_k = R * \sin(k * \theta_0) \\ x_k = R * \cos(k * \theta_0) \end{cases}$$

- (3) 根据对应关系，在画布上绘制所有网络结点对应的坐标点。
- (4) 根据邻接矩阵的值，连接结点之间的边。

1.3. 编写统一的平均路径长度方法

- (1) 平均路径定义：任意两节点间最短路径长度 d 的平均值
- (2) 平均路径长度公式：

$$L = \frac{1}{\frac{1}{2}N(N-1)} \sum_{i \geq j} d_{ij}$$

- d_{ij} 表示节点 i 和 j 之间的最短路径长度;

- N 表示节点个数

(3) 求取邻接矩阵对应的平均路径长度:

① G 为用邻接矩阵表示的带权图, 将 v_0 到其余顶点的路径长度初始化为权值;

$$S \leftarrow \{v_0\}, \text{dist}[i] = G.\text{arcs}[v_0][v_i];$$

② 选择 v_k , 使得 $\text{dist}[v_k] = \min(\text{dist}[i] \mid v_i \in V - S)$, v_k 为目前求得的下一条从 v_0 出发的最短路径的终点。

③ 修改从 v_0 出发到集合 $V - S$ 上任一顶点 v_i 的最短路径的长度。如果 $\text{dist}[k] + G.\text{arcs}[k][i] < \text{dist}[i]$, 则将 $\text{dist}[i] = \text{dist}[k] + G.\text{arcs}[k][i]$

④ 重复(2)、(3) $n-1$ 次, 即可按最短路径长度的递增顺序, 逐个求出 v_0 到图中其它每个顶点的最短路径。

1.4. 编写统一的聚类系数计算方法

(1) 聚类系数定义: 表示一个图中节点聚集程度的系数, 分为局部聚类系数和全局聚类系数。在现实中的网络中, 尤其是在特定的网络中, 由于相对高密度连接点的关系, 节点总是趋向于建立一组严密的组织关系。在现实世界的网络, 这种可能性往往比两个节点之间随机设立了一个连接的平均概率更大。

(2) 局部聚类系数公式:

$$C_i = \frac{E_i}{\frac{1}{2}k_i(k_i - 1)}$$

- E_i 表示节点 i 的邻居之间存在的实际边数;

- k_i 表示节点 i 的邻居个数

(3) 根据局部聚类系数公式, 分别求出每一结点的聚类系数, 进而求取邻接矩阵对应的平均(局部)聚类系数

2. 基础网络实现

2.1. 全局耦合网络

- (1) 全局耦合网络的定义：任意两节点间均有边相连
- (2) 邻接矩阵实现：所有矩阵元素值均置 1，同时去除自环。

2.2. 最近邻耦合网络

- (1) 最近邻耦合网络的定义：任意节点均与其左右 $K/2$ 个邻居节点相连 (K 为偶数)

- (2) 邻接矩阵实现：利用右上三角矩阵进行连续的邻居连边

- ① 初始化邻接矩阵： `originNetwork = sparse(N, N);`
- ② 利用 matlab 的 `diag()` 函数，基于矩阵加法，从一个邻居的连续连边到 K 个邻居的进行循环连续连边：

```
for i = 1 : K
    % 例如：i = 1 时，每个结点与其后的一个邻居连续连边
    % 情况一：邻接矩阵中该节点后一个邻居的下标在该节点下标
    %           之后，如 (2, 3) (4, 5)
    originNetwork = originNetwork + diag(ones(1, N-i), i);
    % 情况二：邻接矩阵中该节点后一个邻居的下标在该节点下标
    %           之前，如 (5, 0)
    originNetwork = originNetwork + diag(ones(1, i), N-i);
```

2.3. 星形网络

- (1) 星形网络的定义：存在一个中心节点，其余节点均只与该中心节点连接
- (2) 邻接矩阵实现：在随机选定的中心结点多对应的邻接矩阵行（列）中置 1，同时去除自环。

2.4. 随机网络：

- (1) 星形网络的定义：任意两点间以一定概率 p 建立交互连接
- (2) 邻接矩阵实现：遍历邻接矩阵的一半三角区域，以概率 p 生成 0 或 1 的随机数，同时对称三角区域进行对称赋值。

3. 小世界模型实现

3.1. WS 小世界模型

- (1) 生成最近邻耦合网络作为规则网络：调用 2.2 中的方法。
- (2) 断边重连：
 - ① 邻接点遍历（向后遍历后半即可，避免重复）
 - ② 随机断边：以 p 为概率随机对表示边的矩阵元素置 0
 - ③ 随机重连：随机选点，在防止自环的情况下，对选定点对的矩阵元素置 1。

3.2. NW 小世界模型

- (1) 生成最近邻耦合网络作为规则网络：调用 2.2 中的方法
- (2) 随机加边：
 - ① 生成随机加边矩阵
 - ② 原本无边，则用随机加边矩阵加边
 - ③ 去除自环

三、实验结果与分析

1. 基础网络可视化与计算结果

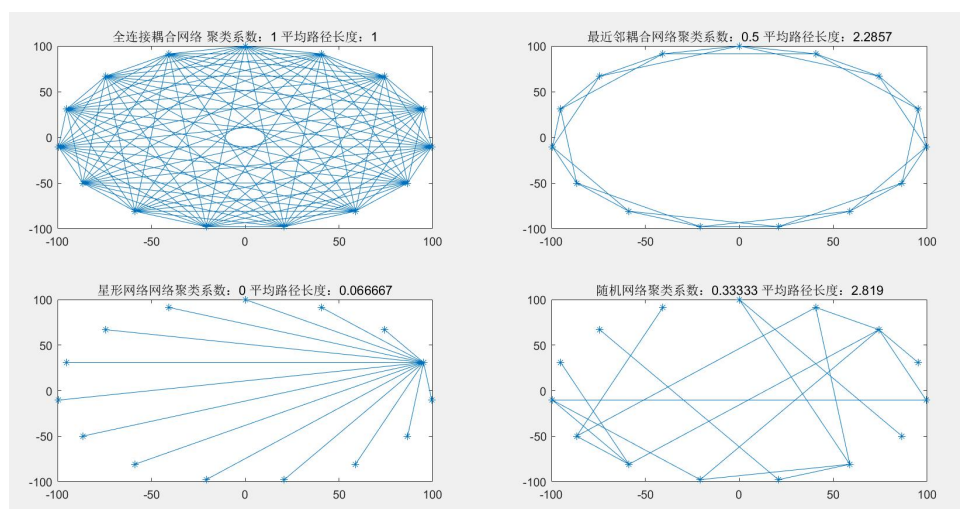


图 1. 基础网络可视化与计算结果

2. 小世界模型可视化与计算结果

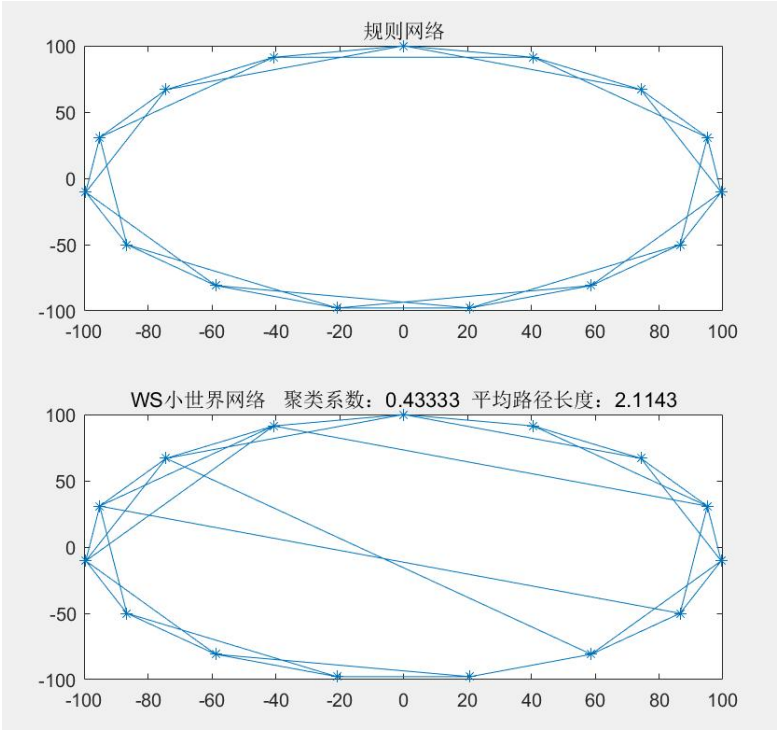


图 2. WS 小世界可视化与计算结果

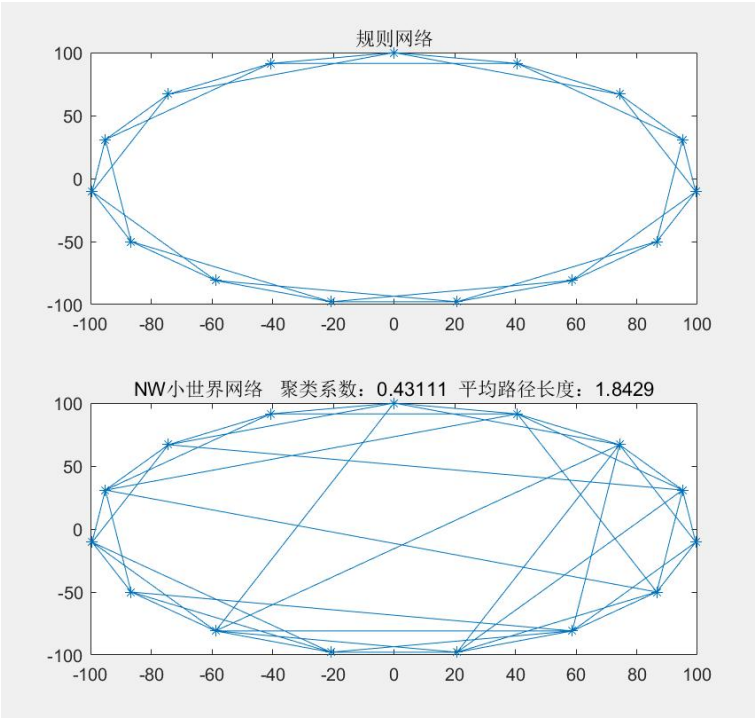


图 3. NW 小世界可视化与计算结果

四、代码附录

1.平均路径长度计算函数定义	1
2.聚类系数计算函数定义	2
3.邻接矩阵的网络可视化函数定义	3
4.基础网络定义	3
(1) 全连接耦合网络	3
(2) 最近邻耦合网络	4
(3) 星形网络	5
(4) 随机网络	5
5.小世界模型定义	6
(1) WS 小世界网络	6
(2) NW 小世界模型	8
6.可视化网络生成	8

1.平均路径长度计算函数定义

```
function avpath = pathLength(A)
```

```
%初始化

N=length(A);

B=sparse([],[],[],N,N,N^2);% 创建一个由 N 个非零值构成的稀疏矩阵，但实际上为此矩阵分配可以存储 N^2
个非零值的空间。

%预定义

L=0;

i=1;

B=B+A^i;

M(i)=nnz(B)-nnz(diag(B)); %忽略自环，返回矩阵非零位置个数，即边总数

%求取所有最短路径

while M(i)>L

    L=M(i);

    i=i+1;

    B=B+A^i;

    M(i)=nnz(B)-nnz(diag(B)); %忽略自环

end

%最长的最短路径数量
```

```
ls=i-1;
```

```
%代入公式，求最短路径和并除以可能的路径总数
```

```
avpath= sum((1:ls).*[M(1) M(2:ls)-M(1:ls-1)]) / (N*(N-1));
```

2.聚类系数计算函数定义

```
function C_net_avcc=clusteringCoefficient(A)
```

```
N=length(A);
```

```
D_k=sum(A,2);
```

```
%预定义
```

```
cc=zeros(1,N);
```

```
for i=1:N
```

```
    neighbours=setdiff(unique([find(A(i,:)) find(A(:,i))']),i);
```

```
    num_neighbours=length(neighbours);
```

```
    if num_neighbours > 1
```

```
        B=A(neighbours,neighbours);
```

```
        num_connect=nnz(B)-nnz(diag(B));
```

```
        cc(i)=num_connect/(num_neighbours*(num_neighbours-1));
```

```
    else
```

```
        cc(i)=0;
```

```
    end
```

```
end
```

```
C_one=length(find(D_k==1));
```

```
C_net_avcc=(C_one+sum(cc))/N;
```


3.邻接矩阵的网络可视化函数定义

```
function matrix2graph(matrix)

    nodeNum = size(matrix,1);

    %节点坐标生成

    angle=0:2*pi/nodeNum:2*pi-2*pi/nodeNum;

    x=100*sin(angle);

    y=100*cos(angle);

    XYCoords = [x,y]';

    %邻接矩阵绘制拓扑图

    gplot(matrix,XYCoords,'-*')

end
```

4.基础网络定义

(1) 全连接耦合网络

```
function globalNetwork = GCN(N)

    % 生成邻接矩阵

    originNetwork = zeros(N);

    for i = 1:N

        for j = 1:N

            originNetwork(i,j) = 1;

        end

    end

    globalNetwork = originNetwork;

    % 计算

    CCgcn = num2str(clusteringCoefficient(globalNetwork));

    PLgcn = num2str(pathLength(globalNetwork));
```

```

% 作图

Tgcn = '全连接耦合网络';

CC = '聚类系数: ';

PL = '平均路径长度: ';

matrix2graph(globalNetwork)

title([Tgcn,CC,CCgcn,PL,PLgcn])

end

```

(2) 最近邻耦合网络

```

function nearistNeighborNetwork = NNCN(N,K)

% 生成邻接矩阵

originNetwork = sparse(N, N);

for i = 1 : K

    originNetwork = originNetwork + diag(ones(1,N-i),i);

    originNetwork = originNetwork + diag(ones(1, i), N-i);

end

originNetwork = originNetwork + originNetwork';

nearistNeighborNetwork = originNetwork;

% 计算

CCnncn = num2str(clusteringCoefficient(nearistNeighborNetwork));

PLnncn = num2str(pathLength(nearistNeighborNetwork));

% 作图

Tnncn = '最近邻耦合网络';

CC = '聚类系数: ';

PL = '平均路径长度: ';

matrix2graph(nearistNeighborNetwork)

title([Tnncn,CC,CCnncn,PL,PLnncn])

end

```

(3) 星形网络

```
function starNetwork = SCN(N,nodeIndex)

% 生成邻接矩阵

originNetwork = zeros(N);

for i = 1:N

    originNetwork(nodeIndex,i) = 1;

end

starNetwork = originNetwork;

% 计算

CCscn = num2str(clusteringCoefficient(starNetwork));

PLscn = num2str(pathLength(starNetwork));

% 作图

Tscn = '星形网络网络';

CC = '聚类系数: ';

PL = '平均路径长度: ';

matrix2graph(starNetwork)

title([Tscn,CC,CCscn,PL,PLscn])

end
```

(4) 随机网络

```
function randomNetwork = ERN(N,p)

% 生成邻接矩阵

originNetwork = zeros(N);

coords = zeros(N,2);

for i = 1:N

    for j = 1:N

        if(rand(1,1)<p)
```

```

        originNetwork(i,j) = 1;

        originNetwork(j,i) = 1;

    end

end

end

randomNetwork = originNetwork;

% 计算

CCern = num2str(clusteringCoefficient(randomNetwork));

PLern = num2str(pathLength(randomNetwork));

% 作图

Tern = '随机网络';

CC = '聚类系数: ';

PL = '平均路径长度: ';

matrix2graph(randomNetwork)

title([Tern,CC,CCern,PL,PLern])

end

```

5.小世界模型定义

(1) WS 小世界网络

```

function WS(N,K,p)

%N 个节点，每个节点与相邻节点共有 2K 条连边，断边重连概率为 p

A = NNCN(N,K);

% 生成规则网络图像

subplot(2,1,1)

matrix2graph(A)

title('规则网络');

% 断边重连

```

```

for i=1:N

    for j=1:K %邻接点遍历（后一半）

        x=i;y=i+K; %邻接点在矩阵中的位置

        if y>N

            y=mod(y,N);

        end

        % 随机断边

        p_r=rand();

        if p_r<=p

            A(x,y)=0;A(y,x)=0;

            % 随机重连

            k=0;

            while k==0

                y=unidrnd(N);%随机选点

                if A(x,y)==0 && x~=y %防止自环

                    A(x,y)=1; A(y,x)=1;%重连

                    k=1;

                end

            end

        end

    end

end

end

% 生成 WS 小世界网络图像

subplot(2,1,2)

matrix2graph(A)

title(['WS 小世界网络 ' ; ' 聚类系数： ',num2str(clusteringCoefficient(A)),' 平均路径长度： ',num2str(pathLength(A))]);

end

```

(2) NW 小世界模型

```
function NW(N,K,p)

%N 个节点，每个节点与相邻节点共有 2K 条连边，随机加边的概率为 p

    A = NNCN(N,K);

    % 生成规则网络图像

    subplot(2,1,1)

    matrix2graph(A)

    title('规则网络');

% 随机加边

    kk=(rand(N,N)<p);%生成随机加边矩阵

    A = logical(A + kk);%原本无边， 则用随机加边矩阵加边

    A = A -diag(diag(A));%去除自环

    % 生成 NW 小世界网络图像

    subplot(2,1,2)

    matrix2graph(A)

    title(['NW 小世界网络 ',' 聚类系数： ',num2str(clusteringCoefficient(A)), ' 平均路径长度： ',num2str(pathLength(A))]);

end
```

6.可视化网络生成

全连接耦合网络

```
subplot(2,2,1)

GCN15 = GCN(15);

% 最近邻耦合网络

subplot(2,2,2)
```

```
NNCN15 = NNCN(15,2);
```

```
% 星形网络
```

```
subplot(2,2,3)
```

```
SCN15 = SCN(15,4);
```

```
% 随机网络
```

```
subplot(2,2,4)
```

```
ERN15 = ERN(15,0.1);
```

```
% WS 小世界网络网络
```

```
figure
```

```
WS(15,2,0.2)
```

```
% NW 小世界网络网络
```

```
figure
```

```
NW(15,2,0.1)
```

