

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)
```



