Homework 7

1.

(1)

```
load univ_cn.mat

v = rank_cn;        % research rank of universities
webpage = univ_cn;  % webpage of universities in mainland china
W = W_cn;           % Link weight matrix

D = sum(W,2);
n = length(D);
idnz = find(D>0);
T = zeros(n,n);
T(idnz,idnz) = diag(1./D(idnz)) * W(idnz,idnz);

alpha = 0.85
```

```
alpha = 0.8500
```

```
T1 = alpha * T + (1-alpha)*ones(n,1)*ones(1,n)/n;
[evec,eval] = eigs(T1',1);
[score_page]=evec/sum(evec);  % pagerank score

% PageRank
[~,idp]=sort(score_page,'descend');
webpage{idp(1:5)}
```

```
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'nju.edu.cn'
ans =
'uestc.edu.cn'
```

```
score_out = D; % out-degree score
score_in = sum(W,1)'; % in-degree score
score_research = max(v)-v; % research score
% Authority ranking
[~,id] = sort(score_in,'descend');
webpage{id(1:5)}
```

```
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'uestc.edu.cn'
ans =
```

```
'sjtu.edu.cn'
ans =
'nju.edu.cn'
```

```
% Hub ranking
[~,id] = sort(score_out,'descend');
webpage{id(1:5)}
```

```
ans =
'pku.edu.cn'
ans =
'ustc.edu.cn'
ans =
'zsu.edu.cn'
ans =
'njau.edu.cn'
ans =
'sjtu.edu.cn'
```

(b).

```
[U,S,V] = svds(W);
u1 = U(:,1)/sum(U(:,1));
v1 = V(:,1)/sum(V(:,1));
[~,idu]=sort(u1,'descend');
[~,idv]=sort(v1,'descend');
webpage{idu(1:5)}
```

```
ans =
'pku.edu.cn'
ans =
'ustc.edu.cn'
ans =
'zsu.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'zju.edu.cn'
```

```
webpage{idv(1:5)}
```

```
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'uestc.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'nju.edu.cn'
```

(c).

```
[~,id] = sort(score_research,'descend');
 webpage{id(1:5)}
```

```
ans =
```

```
'pku.edu.cn'
ans =
'tsinghua.edu.cn'
ans =
'fudan.edu.cn'
ans =
'nju.edu.cn'
ans =
'zju.edu.cn'
```

(d).

```
alpha = [0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.85 0.9]; % alpha=0.85 is Google's PageRank choice

for i=1:length(alpha),
 T1 = alpha(i) * T + (1-alpha(i))*ones(n,1)*ones(1,n)/n;
    [evec,eval] = eigs(T1',1);
 [score_page(:,i)]=evec/sum(evec);  % pagerank score
    % PageRank
    [~,idp]=sort(score_page(:,i),'descend');
    fprintf('alpha: %.2f feet.\n', alpha(i));
    webpage{idp(1:5)}
end
```

```
alpha: 0.10 feet.
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'uestc.edu.cn'
ans =
'nju.edu.cn'
ans =
'sjtu.edu.cn'
alpha: 0.20 feet.
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'uestc.edu.cn'
ans =
'nju.edu.cn'
ans =
'sjtu.edu.cn'
alpha: 0.30 feet.
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'nju.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'uestc.edu.cn'
alpha: 0.40 feet.
ans =
'tsinghua.edu.cn'
ans =
```

```
'pku.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'nju.edu.cn'
ans =
'uestc.edu.cn'
alpha: 0.50 feet.
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'nju.edu.cn'
ans =
'uestc.edu.cn'
alpha: 0.60 feet.
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'nju.edu.cn'
ans =
'uestc.edu.cn'
alpha: 0.70 feet.
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'nju.edu.cn'
ans =
'uestc.edu.cn'
alpha: 0.85 feet.
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'nju.edu.cn'
ans =
'uestc.edu.cn'
alpha: 0.90 feet.
ans =
'tsinghua.edu.cn'
ans =
'pku.edu.cn'
ans =
'sjtu.edu.cn'
ans =
'nju.edu.cn'
ans =
'uestc.edu.cn'
```

4.

```
X=importdata('HongLouMeng374.txt').data;
A=X*X.';
%load('karate.mat');
D=diag(sum(A,1));
L=D-A;
```

(a).

```
[val,vec] = eig(L);
val=diag(val);
[val,id] = sort(val,'ascend');
vec=vec(:,id);

lambda2=val(2);
v2=vec(:,2);
f=v2;
```

(b).

```
[f_sort,idf]=sort(f,'ascend');
i=100;
Si=(1:i);
```

(c).

```
d=diag(D);
n=length(d);
tem=[];
for i=1:d
    Si_mol=sum(d(1:i));
    com_Si_mol=sum(d(i:n));
    par_Si_mol=0;
    for x=1:i
        for y=i:d
            par_Si_mol=A(x,y)+par_Si_mol;
        end
    end
    tem=[tem, par_Si_mol/(min(com_Si_mol,Si_mol))];
end

alpha_f=min(tem);
```

```
disp(alpha_f);
```

```
    0.0142
```

```
disp(lambda2);
```

```
   -0.1598
```

(d).

alpha_f>lambda2

```
indS_pos=find(f>=0);
indS_neg=find(f<0);
tem=[];
for i=1:length(indS_pos)
    Si_mol=sum(d(indS_pos(i)));
    com_Si_mol=sum(d)-Si_mol;
    par_Si_mol=0;
    for j=1:length(indS_neg)
        par_Si_mol=A(indS_pos(i),indS_pos(j))+par_Si_mol;
    end
    tem=[tem, par_Si_mol/(min(com_Si_mol,Si_mol))];
end

h=min(tem);
disp(h);
```

```
     0
```

```
disp(alpha_f);
```

```
    0.0142
```

```
disp(lambda2);
```

```
   -0.1598
```

5.

```
k=5;
n=300;
N=k*n;
B=rand(k,k);
B1=B;
sum_row=sum(B1,2);
for i=1:k
    B(i,:)=B1(i,:)/sum_row(i);
end
theta=ones(N,1);

warning=0;
for i=1:k
    for j=1:k
        for r=1:n
            for s=1:n
                if(theta((i-1)*n+r)*theta((j-1)*n+s)*B(i,j)>1)
                    warning=warning+1;
                end
            end
```

```
        end
    end
end


[A,labels]=DCSBM(n,B,theta);
m=size(A,1);
for i=1:m
    for j=1:m
        if(isnan(A(i,j)))
            A(i,j)=0;
        end
    end
end

d=zeros(1,m);
for i=1:m
   d(i)=sum(A(i,:));
end
D=diag(d);

[evec,eval]=eig(D-A,D);
v1=D^(-1/2)*evec(:,m-k+1:m);
[cidx, ctrs] = kmeans(v1,k);

L=D^(-1/2)*(D-A)*D^(-1/2);
[evec1,eval1]=eig(L);
v2=evec(:,1:k);
for i=1:m
    v2(i,:)=v2(i,:)/norm(v2(i,:),2);
end
[cidx1, ctrs1] = kmeans(v2,k);
%Computing Normalized Mutual Information
I1=NormalizedMI(labels.',cidx);
I2=NormalizedMI(labels.',cidx1);
fprintf('I1: %.4f feet.\n', I1);
```

```
I1: 0.1513 feet.
```

```
fprintf('I2: %.4f feet.\n', I2);
```

```
I2: 0.2904 feet.
```

```
function [A,labels]=DCSBM(n,B,theta)
    Theta=diag(theta);
    k=5;
    N=n*k;
    z=zeros(k,N);
    for i=1:k
        for j=1:k
            z(i,1+(j-1)*n:n+(j-1)*n)=1;
        end
    end
    z=z.';
```

```
    A=Theta*z*B*z.'*Theta;
    labels=[ones(1,n),2.*ones(1,n),3.*ones(1,n),4.*ones(1,n),5.*ones(1,n)];
end


function [NMI] = NormalizedMI(trueLabel, partitionMatrix)
    % normalized mutual information
    % Author: Weike Pan, weikep@cse.ust.hk
    % Ref: Dhilon, KDD 2004 Kernel k-means, Spectral Clustering and Normalized Cuts
    % Section 6.3
    % High NMI value indicates that the clustering and true labels match well

    % usage: NormalizedMI([1 1 1 2 2 2]', [1 2 1 3 3 3]')

    %%
    truey = trueLabel;
    [m1, c] = size(truey); % c: class #

    PM = partitionMatrix;
    [m2, k] = size(PM); % k: cluster #

    %%
    % check whether m1 == m2
    if m1 ~= m2
        error('m1 not equal m2');
    else
        m = m1;
    end

    %% change the truelable or the partition matrix: m \times c
    if c == 1
        c = length( unique(truey) );
        tmp = zeros(m,c);
        for i = 1 : c
            tmp((truey == i), i) = 1;
        end
        truey = tmp;
    end

    if k == 1
        k = length( unique(PM) );
        tmp = zeros(m,k);
        for i = 1 : k
            tmp((PM == i), i) = 1;
        end
        PM = tmp;
    end

    %%

    % ****************************
    % calculate the confusion matrix
    for l = 1 : 1 : k
        for h = 1 : 1 : c
```

```
            n(l,h) = sum( (truey(:,h) == 1) & (PM(:,l) == 1) );
        end
    end



    % ****************************
    NMI = 0;
    for l = 1 : 1 : k

        for h = 1 : 1 : c
            NMI = NMI + (n(l,h)/m) * log(  ( n(l,h)*m + eps) / ( sum(n(:,h))*sum(n(l,:)) + eps)

        end

    end

    Hpi = - sum( (sum(PM)/m) .* log( sum(PM)/m + eps ) );
    Hvarsigma = - sum( (sum(truey)/m) .* log( sum(truey)/m + eps ) );

    % NMI = 2*NMI/(Hpi + Hvarsigma);

    % JMLR03, A. Strehl and J. Ghosh. Cluster ensembles -- a knowledge reuse framework for comb
    NMI = NMI/sqrt(Hpi*Hvarsigma);
end
```