HW0

Problem 1:

function R = projection(A, k)

p = size(A, 2);

P = rand(p, k);

colsums = sum(A, 1);

for i = 1:p

A(:,i) = A(:,i) ./ colsums(i);

end

R = A \* P;

Iris:



Outliers: Xprojected(I1, :) = 0.0166 0.0123



Outliers: Xprojected(I1, :) = 0.0250 0.0321

HallOfFame:



Outliers: Yprojected(I2, :) = 0.2264 0.2524



Outliers: Yprojected(I2, :) = 0.2818 0.2685

Problem 2:

1.

heroes;

[V, E] = eig(hero\_network);

e = diag(E);

plot(e, 'g.');



2.

[largest, index] = max(e);

spectralnorm = norm(hero\_network)

Yes, the largest eigenvalue is equal to the spectral norm = 0.0058

3.

eigenvector = V(:,index) %print eigenvector corresp to largest eigenvalue

plot(eigenvector, 'b.');

The eigenvector corresponding to the largest eigenvalue is nonegative.



%find entry with largest value in eigenvector:

[largest, index] = max(eigenvector);

names(index)

Index = 82, so the 82nd superhero corresponds to the max value in the eigenvector with the largest eigenvalue.

Printing names(index) shows that the 82nd superhero is Captain America.

4.

%degree sequence

Nonzero\_entries = logical(hero\_network);

d = sum(Nonzero\_entries, 2)

plot(d, ‘r.’);



[largest, index] = max(d);

index = 82

The 82nd superhero has the largest value. This corresponds to Captain America. The results match the result from the eigenvalue & eigenvector analysis.

5.

D = diag(d);

L = D - Nonzero\_entries;

eigenvals = eig(L);

for i=1:size(eigenvals, 1)

if abs(eigenvals(i)) < 1e-14,

eigenvals(i) = 0;

end

end

z = ~logical(eigenvals);

numzeros = sum(z)

There are no zero eigenvalues, using the threshold 1e-14.

6.

%Perron-Frobenius

colsums = sum(hero\_network, 2) %sum cols of adjacency matrix

mincols = min(colsums)

maxcols = max(colsums)

The min is 8.8778e-05, max is 0.0136. The max eigenvalue is 0.0058, which is between those two values. The theorem holds.

Problem 3:

1.

%Jacobi function

function [V, L] = jacobi(A)

n = size(A, 1);

V = eye(n);

offDiag = logical(1-eye(n));

while (norm(A(offDiag), 'fro')^2/norm(A, 'fro')^2 > n\*n\*eps)

max = -Inf;

%choose p, q so |A(p, q)| is maximized

for p = 1:(n-1)

for q = (p+1):n

curr = abs(A(p, q));

if curr > max,

max = curr;

maxp = p;

maxq = q;

end

end

end

[c, s] = symschur2(A, maxp, maxq);

J = eye(n);

J(maxp, maxp) = c; J(maxp, maxq) = s; J(maxq, maxp) = -s; J(maxq, maxq) = c;

A = J'\*A\*J;

V = V\*J;

end

L = diag(diag(A));

%---------------------------------------------

%rotation function

function [c, s] = symschur2(A, p, q)

if A(p, q) ~= 0,

r = (A(q, q) - A(p, p)/(2\*A(p, q)));

if r >= 0,

t = 1/(r + sqrt(1+(r^2)));

else

t = -1/(-r + sqrt(1+(r^2)));

end

c = 1/sqrt(1+(t^2));

s = t\*c;

else

c = 1;

s = 0;

end

Running jacobi function on hero\_network:

[V L] = jacobi(hero\_network)

First eigenvalue: 0.0058

The values match the results from before, using eig(hero\_network)= 0.058