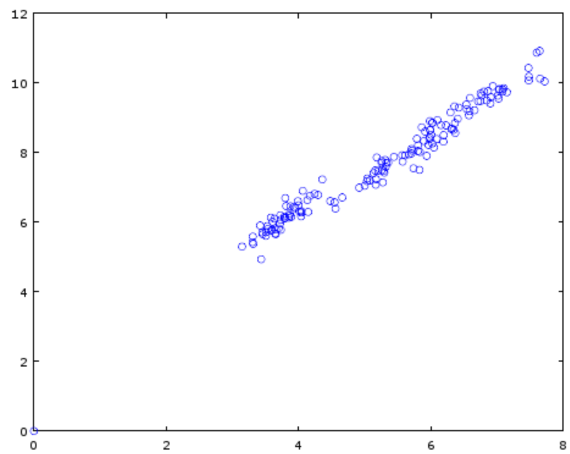
Eric Yang

304263623

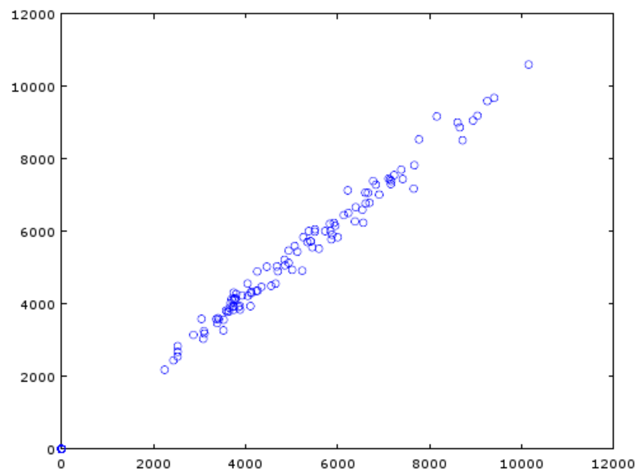
Homework 0

1. function [reduced] = randomProjection (A, k)
[b,c] = size(A);
P = rand(c,k);
denom = repmat(sum(A,1),[b,1]);
C = A ./ denom;
reduced = C * P;

This function yields these plots for Iris and HallOfFame respectively:

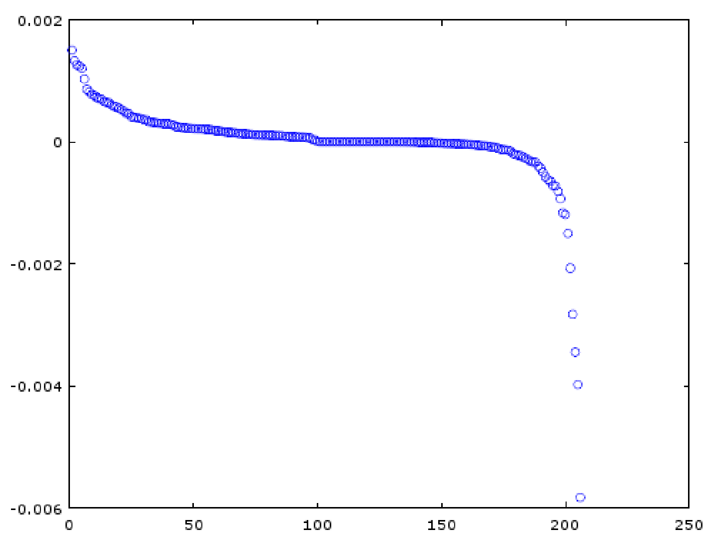


The greatest outlier is (7.64263 , 10.90997) for Iris



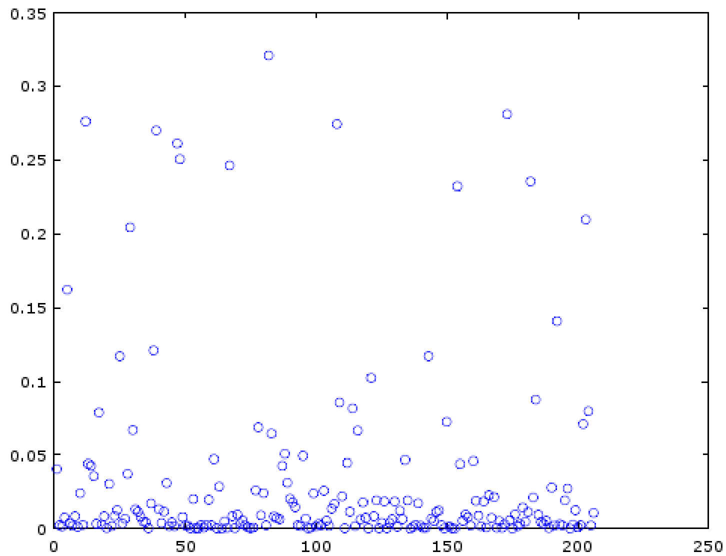
The greatest outlier is (1.0153e+004 , 1.0593e+004) for HallOfFame

2.1. Eigenvalues of H found with eig() plotted against index numbers:



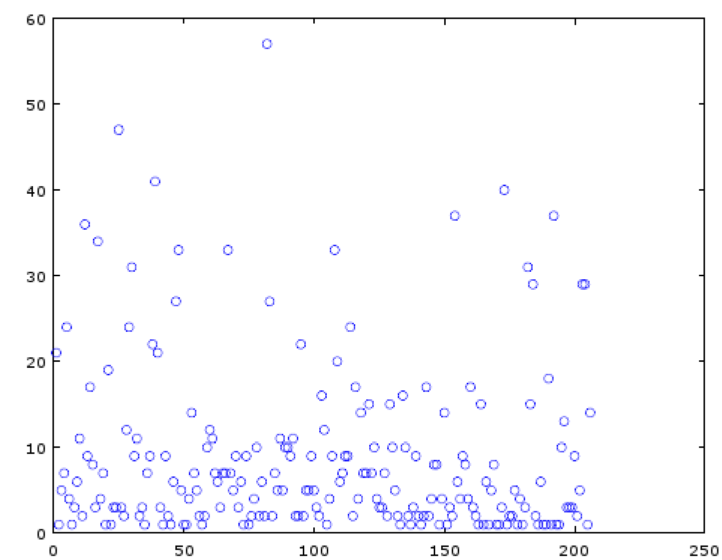
2.2. The largest eigenvalue found using eigs() is 0.0058208 which is the same as the spectral norm found with norm() at 0.0058208.

2.3. Used eigs() to find the eigenvector corresponding to eigenvalue 0.0058208 and plotted against index:



The graph shows that no entries of the eigenvector are negative. The largest entry at 0.32085 corresponds to index number 82 and is the hero Captain America.

2.4. Below is a plot of the degree sequence of the hero network that was found by converting all nonzero entries in the network to 1 and then summing all the rows:



The hero with the most connections at 57 connections has index number 82 which is Captain America.

2.5. After executing L = D – A with D = diag() on the degree sequence from the previous problem and A = logical() on the hero network, there is only 1 zero eigenvalue of L.

2.6. The min and max sums that I found were 8.8778e-005 and 0.013602 respectively.

3. Below is an implementation of the Jacobi method that finds the eigenvalues and eigenvectors of a real symmetric matrix A:

```
function [Q L] = RealSymmetricEig(A)
        n = size(A,1);
        Q = eye(n);
        if ( norm( (A - A'), 'fro') / norm(A,'fro') > n * n * eps )
                disp('Error! Matrix not symmetric!')
                A = (A + A')/2;
        end
        OffDiagonal = logical(1 - eye(n));
        while( norm(A(OffDiagonal),'fro')^2 / norm(A,'fro')^2 > n * n * eps )
                maxvalue = -Inf;
                for i = 1:(n-1)
                        for j = (i+1):n
                                if (abs(A(i,j)) > maxvalue), maxvalue = abs(A(i,j)); maxi = i; maxj = j; end
                        end
                 end
                i = maxi;
                j = maxj;
                a = A(i,i);
                b = A(i,j);
                d = A(j,j);
                T = JacobiQRotation(a,b,d);
                Q( 1:n, [i j] ) = Q( 1:n, [i j] ) * T;
                A( [i j], 1:n ) = T' * A( [i j], 1:n );
                A( 1:n, [i j] ) = A( 1:n, [i j] ) * T; % apply T on right to columns [i j] of A.
        end
        L = diag(diag(A));

Function T = Qrotation(x,y)
        c = 1;
        s = 0;
        if(abs(y) >0)
                if (abs(y) >= abs(x))
                        cotangent = x/y;
```

```
                s = 1/sqrt(1 + cotangent^2);
                c = s * cotangent;
        else
                tangent – y/x;
                c = 1/sqrt(1 + tangent^2);
                s = c * tangent;
        end
end
T = [c –s, s c];
```

This function does return the same eigenvalues as earlier for input H and the same first eigenvector.