MATH 285 HW3

Xiaoyan Chong

November 20, 2015

1.

(a) Draw the graph

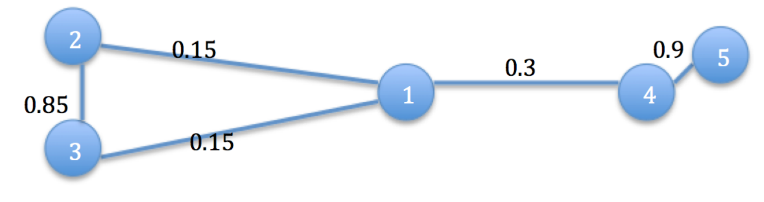

Figure 1: Graph

(b) Find the degrees of all vertices

$$
\begin{aligned}
d_1 &= 0 + .15 + .15 + .3 + 0 = .6 \\
d_2 &= .15 + 0 + .85 + 0 + 0 = 1 \\
d_3 &= .15 + .85 + 0 + 0 + 0 = 1 \\
d_4 &= .3 + 0 + 0 + 0 + .9 = 1.2 \\
d_5 &= 0 + 0 + .0 + .9 + 0 = .9
\end{aligned}
$$

As a result, the degree matrix can be written as:

$$
\text{D}= \begin{pmatrix} 0.6 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1.2 & 0 \\ 0 & 0 & 0 & 0 & 0.9 \end{pmatrix}
$$

(c) (i)$V = \{v_1, v_2, v_3\} \cup \{v_2, v_5\}$

$$
\begin{aligned}
Ncut &= \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)} \\
&= \frac{0.3}{0.6+1+1} + \frac{0.3}{1.2+0.9} \\
&= 0.2582
\end{aligned}
$$

$$
\begin{aligned}
Ratiocut &= cut(A,B)(\frac{1}{|A|} + \frac{1}{|B|}) \\
&= 0.3(\frac{1}{3} + \frac{1}{2}) \\
&= 0.25
\end{aligned}
$$

(ii)$V = \{v_1, v_4, v_5\} \cup \{v_2, v_3\}$

$$
\begin{aligned}
Ncut &= \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)} \\
&= \frac{0.3}{0.6+1.2+0.9} + \frac{0.3}{1+1} \\
&= 0.2611
\end{aligned}
$$

$$
\begin{aligned}
Ratiocut &= cut(A,B)(\frac{1}{|A|} + \frac{1}{|B|}) \\
&= 0.3(\frac{1}{3} + \frac{1}{2}) \\
&= 0.25
\end{aligned}
$$

As a result, $V = \{v_1, v_2, v_3\} \cup \{v_2, v_5\}$ has a smaller Ncut. The two have the same ratio cut.

2. Proof:

$$
\begin{aligned}
x^T L x &= \frac{1}{2}\sum_{i\neq j} w_{ij}(x_i - x_j)^2 \\
&= \frac{1}{2}\Big[\sum_{i,j\in A} 0 + \sum_{i,j\in B} 0 + \sum_{i\in A, j\in B} w_{ij}(\frac{1}{vol(A)} + \frac{1}{vol(B)})^2 + \sum_{i\in B, j\in A} w_{ij}(\frac{1}{vol(A)} + \frac{1}{vol(B)})^2\Big] \\
&= \sum_{i\in A, j\in B} w_{ij}(\frac{1}{vol(A)} + \frac{1}{vol(B)})^2 \\
&= cut(A,B)(\frac{1}{vol(A)} + \frac{1}{vol(B)})^2
\end{aligned}
$$

On the other hand,

$$
\begin{aligned}
x^T D x &= \sum_{i=1}^{n} d_i x_i^2 \\
&= (\sum_{i\in A} d_i)(\frac{1}{vol(A)})^2 + (\sum_{i\in B} d_i)(-\frac{1}{vol(B)})^2 \\
&= vol(A)(\frac{1}{vol(A)})^2 + vol(B)(-\frac{1}{vol(B)})^2 \\
&= \frac{1}{vol(A)} + \frac{1}{vol(B)}
\end{aligned}
$$

As a result,

$$
\begin{aligned}
\frac{x^T L x}{x^T D x} &= \frac{cut(A,B)(\frac{1}{vol(A)} + \frac{1}{vol(B)})^2}{\frac{1}{vol(A)} + \frac{1}{vol(B)}} \\
&= cut(A,B)(\frac{1}{vol(A)} + \frac{1}{vol(B)}) \\
&= NCut(A,B)
\end{aligned}
$$

So when we have $x_i = \begin{cases} \frac{1}{vol(A)}, & i \in A \\ -\frac{1}{vol(A)}, & i \in B \end{cases}$, the relation $NCut(A,B) = \frac{x^T L x}{x^T D x}$ still holds.

3.

By analysing the data, we get the following few figures. Figure 2 is the scatter plot of the raw data. Figure 3 is the weighted graph.
Figure 4 shows the first 8 smallest eigenvalues, and Figure 5 gives the corresponding eigenvector. Figure 6 is the final plot based on NCut algorithm. According to Figure 4, we can see there are 4 eigenvalues are close to zero, and our final spectral clustering also shows we have 4 groups. We conclude NCut did pretty good job.

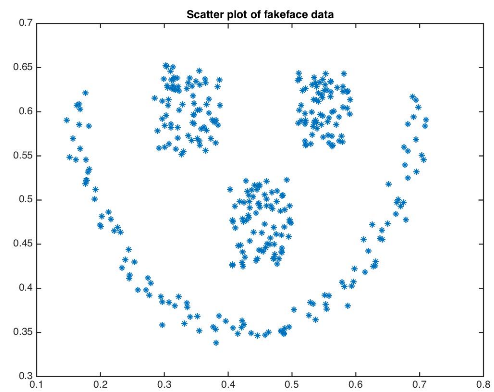

Figure 2: Scatter plot

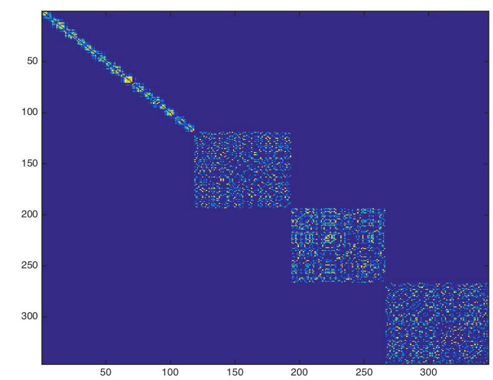

Figure 3: Weighted graph

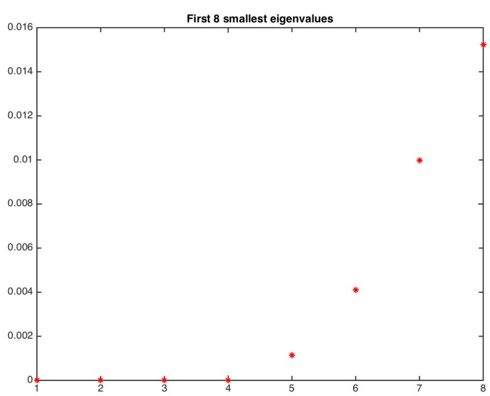

Figure 4: First 8 smallest eigenvalue plot

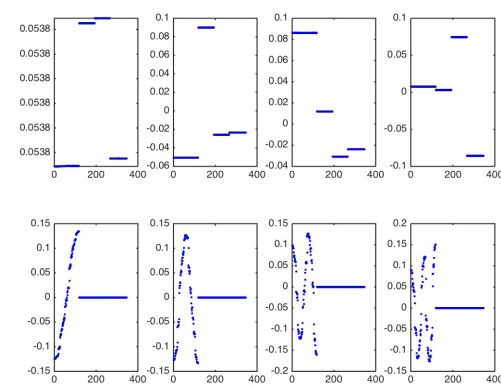

Figure 5: First 8 smallest eigenvalue plot

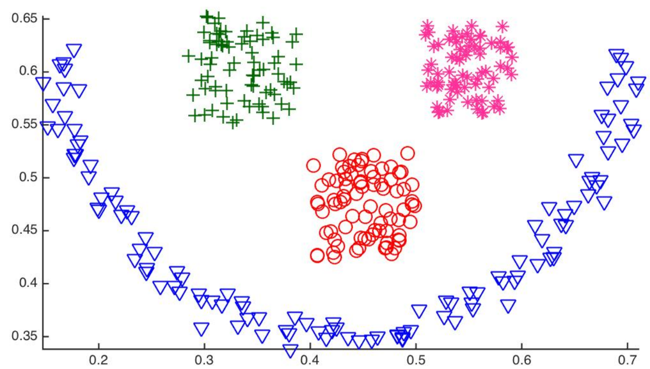

Figure 6:

My code is as follows:

```
%% Problem 3

 clear
 close all
 clc

%% load data
load fakeface.mat

f3_1 = figure; plot(X(:,1),X(:,2),'*')
title 'Scatter plot of fakeface data'
saveas(f3_1,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/3_1_scatt

%% distance

Dis = L2_distance(X',X',1);
D_sqr = Dis.^2;

%% sigma

k_num = 4;
[idx, di] = knnsearch(X, X, 'k',k_num);
dis_diff = di(:, k_num);
sigma = mean(dis_diff);
sigma_sqr = sigma^2;

%% construct Weighted graph:
```

```
W1 = exp(-D_sqr./2/sigma_sqr);
n_w = size(W1,1);
W = W1 - eye(n_w);
f3_2=figure;
imagesc(W)
saveas(f3_2,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/3_2_weigh

% DisplayImageCollection(W)
%% Graph laplacian

D_diag = sum(W,2);
D = diag(D_diag);
L = D - W;

%% normalized
Lrw = inv(D)* L;
[V, S]= eig(Lrw);
[dsort, idum] = sort(diag(S),'ascend');
l= abs(dsort);
V= V(:, idum);

%% plot the smallest 8 eigenvalues and eigenvectors;

dimen = 8;
f3_3=figure;  plot(1:dimen, l(1:dimen),'r*');
str1 = strcat({'First '}, num2str(dimen),{' smallest eigenvalues'});
title(str1)
saveas(f3_3,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/3_3_eigen


f3_4 =figure;
for i = 1:dimen
    subplot(2,4,i);
    plot(V(:,i),'b.');
    hold on
end
%str2 = strcat({'First '}, num2str(dimen),{' smallest eigenvectors'});
%title(str2);
hold off;
saveas(f3_4,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/3_4_eigen


%% apply k-means, four groups

V_new = V(:,2);
mylabel = kmeans(V_new,4, 'Replicates',10);
f3_5 = figure; gcplot(V_new, mylabel); axis equal;
title '1-D'
```

```
saveas(f3_5,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/3_5_1d','

f3_6=figure; gcplot(X, mylabel); axis equal
saveas(f3_6,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/3_6_2d','
```

4.

Figure 7 is the scatter plot of the raw data. Figure 8is the weighted graph.

Normalized situation:

Figure 9 and Figure 10 show the first 8 smallest eigenvalues and the the corresponding eigenvector for normalized situation. Figure 11 is the final plot based on normalized algorithm.

Unnormalized situation:

Figure 12 and Figure 13 show the first 8 smallest eigenvalues and the the corresponding eigenvector for unnormalized situation.

(1) Using clusters = 4

Figure 14 is the final plot based on unnormalized algorithm, here we use clusters = 4 based on the eigenvalue plot.

(2) Using clusters = 2

Figure 15 is the final plot based on unnormalized algorithm, here we use clusters = 2. Compared figure 11 and 15 , they are both k =2, and they give us the same clusters. According to these two figures, we cannot see much difference between normalized and unnormalized situation.

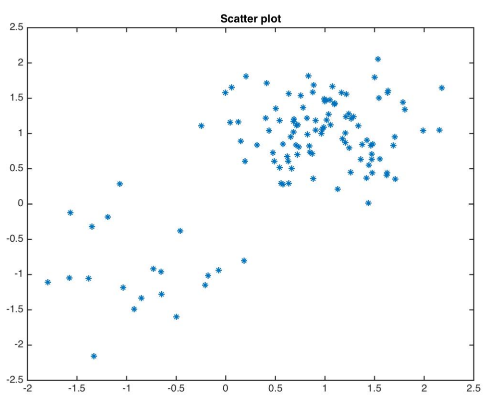

Figure 7: Scatter plot

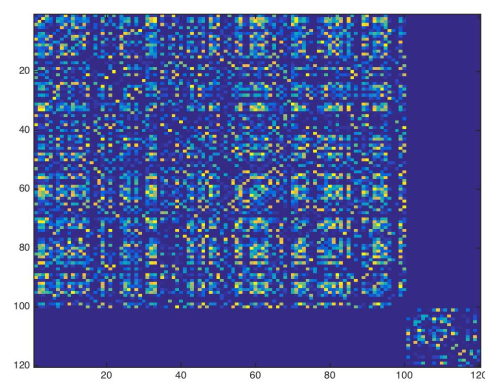

Figure 8: Weighted graph

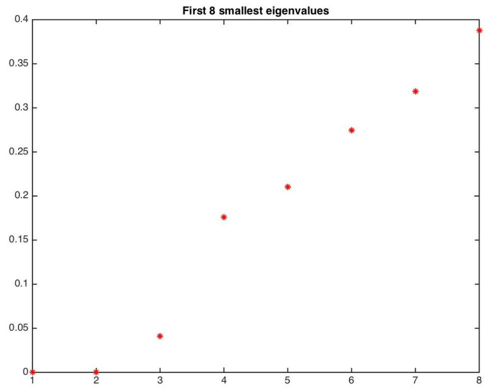

Figure 9: (Normalized)First 8 smallest eigenvalue plot

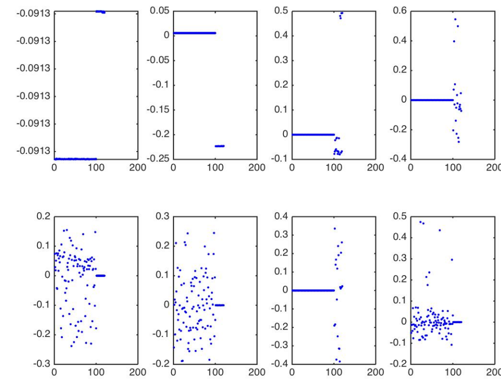

Figure 10: (Normalized)First 8 smallest eigenvalue plot

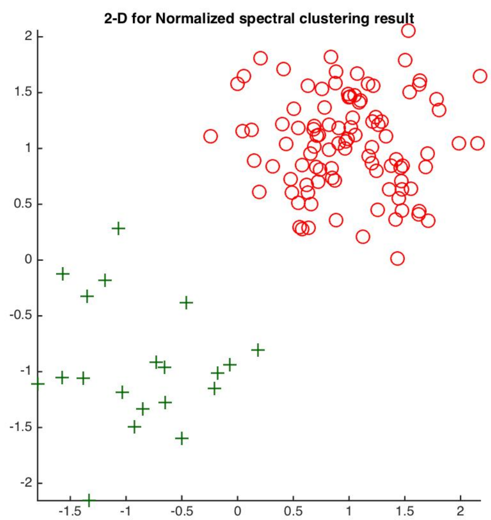

Figure 11: Normalized situation

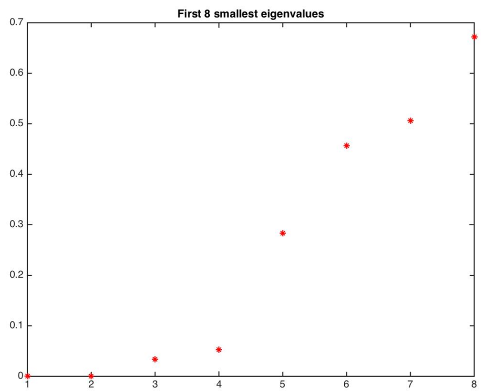

Figure 12: (Unnormalized)First 8 smallest eigenvalue plot

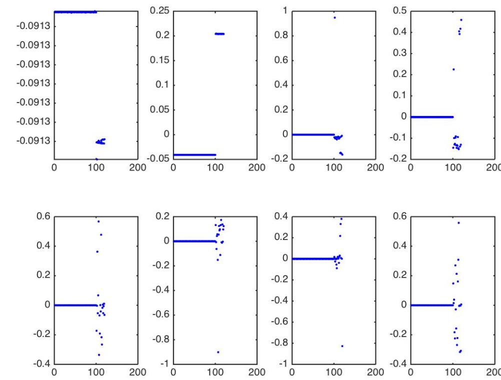

Figure 13: (Unnormalized)First 8 smallest eigenvalue plot

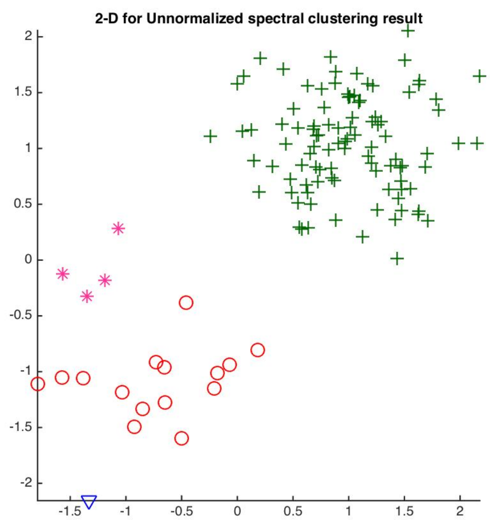

Figure 14: Unnormalized situation (clusters = 4)

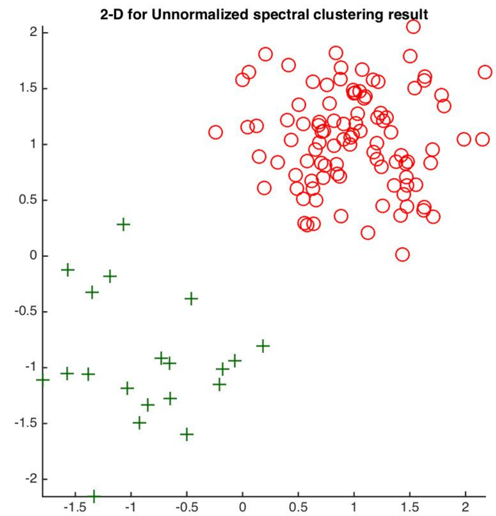

Figure 15: Unnormalized situation (clusters = 2)

```
%% Problem 4

 clear
 close all
 clc

%% load data
load twogaussians_1L1S

f4_1 = figure; plot(X(:,1),X(:,2),'*')
title 'Scatter plot'
saveas(f4_1,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_1_scatt

%% distance

Dis = L2_distance(X',X',1);
D_sqr = Dis.^2;

%% sigma

k_num = 5;
[idx, di] = knnsearch(X, X, 'k',k_num);
dis_diff = di(:, k_num);
sigma = mean(dis_diff);
sigma_sqr = sigma^2;

%% construct Weighted graph:

W1 = exp(-D_sqr./2/sigma_sqr);
```

```
n_w = size(W1,1);
W = W1 - eye(n_w);
f4_2 = figure;
imagesc(W)
saveas(f4_2,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_2_weigh


%% Graph laplacian

D_diag = sum(W,2);
D = diag(D_diag);
L = D - W;

%% normalized
Lrw = inv(D)* L;
[V, S]= eig(Lrw);
[dsort, idum] = sort(diag(S),'ascend');
l= abs(dsort);
V= V(:, idum);

%% the smallest 8 eigenvalues and eigenvectors;

dimen = 8;
f4_3 = figure;  plot(1:dimen, l(1:dimen),'r*');
str1 = strcat({'First '}, num2str(dimen),{' smallest eigenvalues'});
title(str1)
saveas(f4_3,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_3_eigen


f4_4 = figure;
for i = 1:dimen
    subplot(2,4,i);
    plot(V(:,i),'b.');
    hold on
end
%str2 = strcat({'First '}, num2str(dimen),{' smallest eigenvectors'});
%title(str2);
hold off;

saveas(f4_4,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_4_eigen


%% apply k-means, 2 groups

V_new = V(:,2);
mylabel = kmeans(V_new,2, 'Replicates',10);
f4_5 = figure; gcplot(V_new, mylabel); axis equal;
title '1-D for Normalized spectral clustering result'
```

```
saveas(f4_5,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_5_1d','


f4_6 = figure; gcplot(X, mylabel); axis equal
title '2-D for Normalized spectral clustering result'
saveas(f4_6,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_6_2df',

%% unnormalized

[V_un, S_un]= eig(L);
[dsort_un, idum_un] = sort(diag(S_un),'ascend');
l_un= abs(dsort_un);
V_un= V_un(:, idum_un);


%% the smallest 8 eigenvalues and eigenvectors;

dimen_un = 8;
f4_7 = figure;  plot(1:dimen_un, l_un(1:dimen_un),'r*');
str1 = strcat({'First '}, num2str(dimen_un),{' smallest eigenvalues'});
title(str1)
saveas(f4_7,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_7_unno

f4_8= figure;
for i = 1:dimen_un
    subplot(2,4,i);
    plot(V_un(:,i),'b.');
    hold on
end
%str2 = strcat({'First '}, num2str(dimen),{' smallest eigenvectors'});
%title(str2);
hold off;
saveas(f4_8,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_8_unno


%% apply k-means, 4 groups

V_new_un = V_un(:,2:4);
mylabel_un = kmeans(V_new_un,4, 'Replicates',10);
figure; gcplot(V_new_un, mylabel_un); axis equal;
title '1-D for Unnormalized spectral clustering result'
f4_9 = figure; gcplot(X, mylabel_un); axis equal
title '2-D for Unnormalized spectral clustering result'
saveas(f4_9,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_9_unno


%% apply k-means, 2 groups
```

```
V_new_un = V_un(:,2);
mylabel_un = kmeans(V_new_un,2, 'Replicates',10);
figure; gcplot(V_new_un, mylabel_un); axis equal;
title '1-D for Unnormalized spectral clustering result'
f4_9_2 = figure; gcplot(X, mylabel_un); axis equal
title '2-D for Unnormalized spectral clustering result'
saveas(f4_9_2,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/4_9_unn
```

5.

Figure 16 is the scatter plot of raw data. Figure 17 shows the first 8 smallest eigenvalues for each sigma in this problem, from which we can tell how many eigenvalues that are very close to zero respectively.

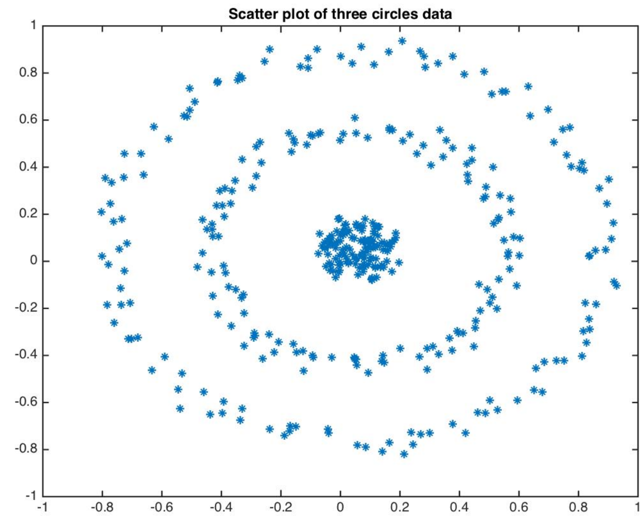

Figure 16: Scatter plot

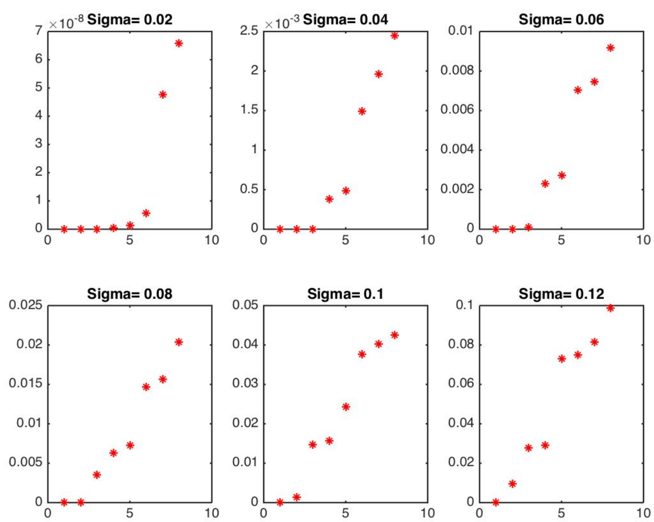

Figure 17: Eigenvalue plot for each sigma

Case one: we select how many clusters we will have based on eigenvalues. After checking the eigenvalue plot, we get the best k for each group are [6,3,3,2,2,1], based on this, we made Figure 19. From this figure, we conclude that when k = 3, which means the second and the third plots, perform correctly on this data. The corresponing sigma is 0.04 and 0.06.

By the output of matlab, we get the corresponding number of scatter for each sigma is [0.0007 0.0000 0.0001 0.0002 0.0028 120.4981]. The second one is the smallest, and third one is the second smallest value. So the plot can predict correctly for the value of σ.

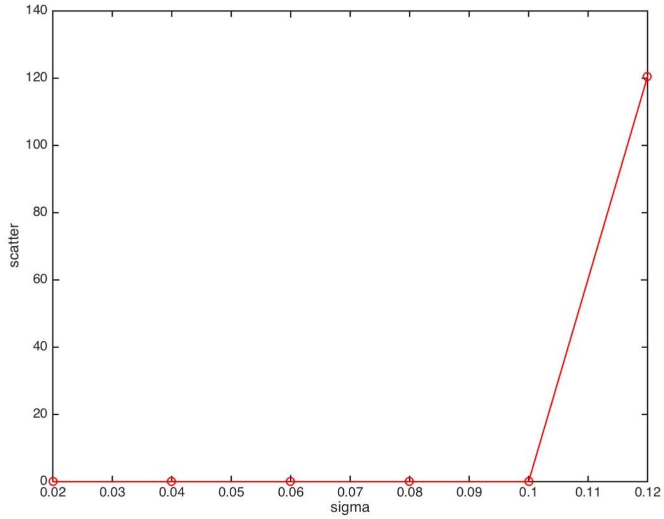

Figure 18: Total scatter(multiple clusters)

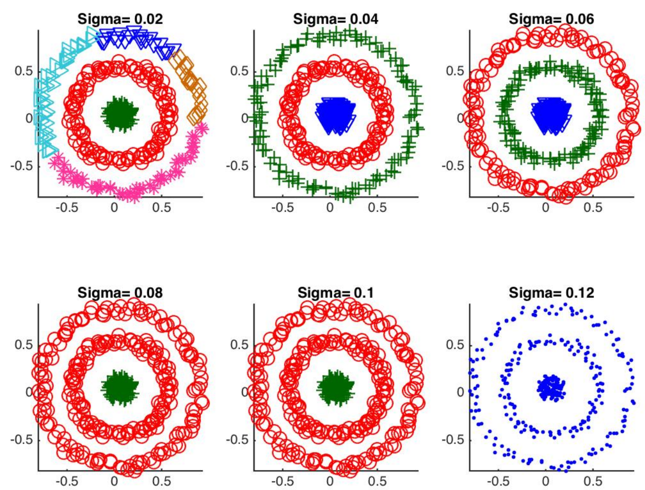

Figure 19: Spectral clusters (multiple clusters)

Case two: In this case, we use k = 3 (3 clusters) for all sigmas. Figure 20 and figure 21 show the result. According to these two, we find the first four sigma (smaller sigma) will give correct clusters, and the corresponding values of scatter are very small: [0.0000 0.0000 0.0001 0.0162] . The last two sigmas (larger sigma) give wrong classes, and the corresponding values of scatter are pretty big: [0.2808, 0.2556]. We conclude when $\sigma = 0.02, 0.04$ the total scatter is smallest, so these two are considered to be optimal. We can get this according to the plot.

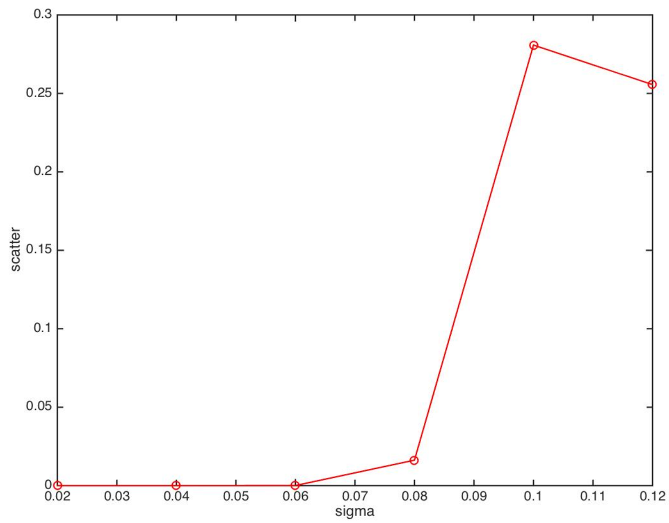

Figure 20: Total scatter (3 clusters)

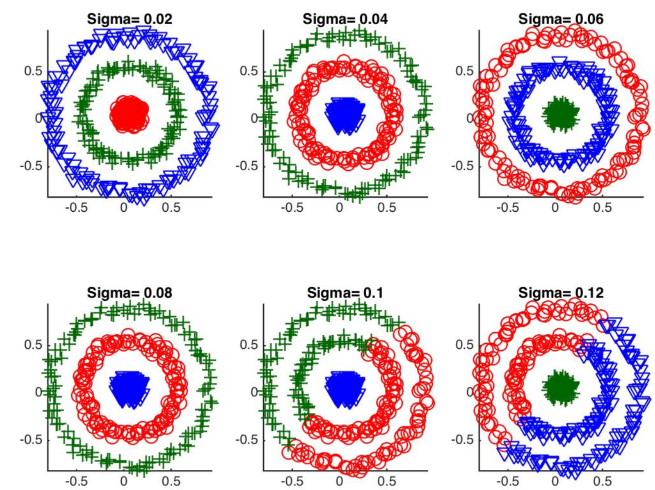

Figure 21: Spectral clusters (3 clusters)

```
%% Problem 5

clear;
close all;
clc;

%% load data
load threecircles.mat
f5_1 = figure; plot(X(:,1),X(:,2),'*')
```

```
title 'Scatter plot of three circles data'
% saveas(f5_1,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/5_1_sca

%% distance

Dis = L2_distance(X',X',1);
D_sqr = Dis.^2;

%% the smallest 8 eigenvalues and eigenvectors;
sigma = [0.02, 0.04, 0.06, 0.08, 0.1, 0.12];
f5_3 = figure;
for i = 1: length(sigma)
    sigma(i);
    sigma_sqr = sigma(i)^2;

    W1 = exp(-D_sqr./2/sigma_sqr);
    n_w = size(W1,1);
    W = W1 - eye(n_w);

    D_diag = sum(W,2);
    D = diag(D_diag);
    L = D - W;
    Lrw = inv(D)* L;
    [V, S]= eig(Lrw);
    [dsort, idum] = sort(diag(S),'ascend');
    l= abs(dsort);
    V= V(:, idum);

    % eigenvalues against sigma plot
    subplot(2,3,i);
    dimen = 8;
    plot(1:dimen, l(1:dimen),'r*');
    str1 = strcat({'Sigma= '}, num2str(sigma(i)));
    title(str1)

    hold on;
end

hold off;

%saveas(f5_3,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/5_3_eige

%% Plot All clusters
clear;
close all;
clc;

%% load data
```

```
load threecircles.mat

%%
Dis = L2_distance(X',X',1);
D_sqr = Dis.^2;

sigma = [0.02, 0.04, 0.06, 0.08, 0.1, 0.12];

V = cell(6,1)
Lrw = cell(6,1)
for i = 1: length(sigma)
    sigma(i);
    sigma_sqr = sigma(i)^2;

    W1 = exp(-D_sqr./2/sigma_sqr);
    n_w = size(W1,1);
    W = W1 - eye(n_w);

    D_diag = sum(W,2);
    D = diag(D_diag);
    L = D - W;
    Lrw{i,1} = inv(D)* L;
    [V1, S]= eig(Lrw{i,1});
    [dsort, idum] = sort(diag(S),'ascend');
    l= abs(dsort);
    V{i,1}= V1(:, idum);

end


%% Spectral clusters and scatter

% dim = [6,3,3,2,2,1];
% for j = 1: length(dim)
%
%     if dim(j) > 1
%     V_new = V{j,1}(:,2:dim(j));
%
%     [mylabel, ~, mydis,~]  = kmeans(V_new,dim(j), 'Replicates',10);
%     scatter(j) = sum(mydis);
%
%     subplot(2,3,j);
%     gcplot(X, mylabel);axis equal
%     str = strcat({'Sigma= '}, num2str(sigma(j)));
%     title(str)
%     hold on;
%
%     else
```

```
%       %V_new = V{j,1}(:,1);
%       %V_new = X;
%      % [mylabel, ~, mydis,~]  = kmeans(V_new,dim(j), 'Replicates',10);
%       [mylabel, ~, mydis,~]  = kmeans(V_new,3, 'Replicates',10);
%
%       scatter(j) = sum(mydis);
%
%       subplot(2,3,j);
%       gcplot(X, mylabel);axis equal
%       str = strcat({'Sigma= '}, num2str(sigma(j)));
%       title(str)
%       hold on;
%       end
%
% end
% hold off;
%
% % saveas(f5_2,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/5_2_

dim = [6,3,3,2,2,1];
f5_2_2 = figure;
scatter = zeros(1,6);

for j = 1: length(dim)
    V_new = V{j,1}(:,2:3);
    [mylabel, ~, mydis,~]  = kmeans(V_new,3, 'Replicates',10);
    scatter(j) = sum(mydis);
    subplot(2,3,j);
    gcplot(X, mylabel);axis equal
    str = strcat({'Sigma= '}, num2str(sigma(j)));
    title(str)
    hold on;
end

hold off;
saveas(f5_2_2,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/5_2_fin

%% Scatter versus sigma

f5_4_2 = figure;
plot(sigma, scatter,'ro-')
xlabel 'sigma'
ylabel 'scatter'
saveas(f5_4_2,'/Users/XC/Dropbox/SJSU/Courses/Math 285 Data Modeling/HW/hw3/5_total
```