INTRODUCTION TO MACHINE LEARNING

HW 3

1) Multi-class Fisher Discriminant Analysis

It is known that $J(w) = \frac{w^T S_B w}{w^T S_W w}$ needs to be maximized. In this equation,

$S_W = \sum_{i=1}^{C} S_i = \sum_{i=1}^{C} \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$ where $\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$ and

$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T$ where $\mu = \frac{1}{N} \sum_{\forall x} x = \frac{1}{N} \sum_{i=1}^{C} N_i \mu_i$

By taking the derivation of $J(w)$ with respect to w and then equating to zero helps us to maximize it.

$$\Rightarrow \frac{d}{dw}[J(w)] = \frac{d}{dw}\left[\frac{w^T S_B w}{w^T S_W w}\right] = 0$$

$$\Rightarrow (w^T S_W w) \frac{d}{dw}[w^T S_B w] - (w^T S_B w) \frac{d}{dw}[w^T S_W w] = 0$$

$$\Rightarrow (w^T S_W w) \cdot 2 S_B w - (w^T S_B w) \cdot 2 S_W w = 0$$

Dividing by $w^T S_W w$

$$\Rightarrow \left(\frac{w^T S_W w}{w^T S_W w}\right) \cdot S_B w - \left(\frac{w^T S_B w}{w^T S_W w}\right) \cdot S_W w = 0$$

$$\Rightarrow S_B w - J S_W w = 0$$

$$\Rightarrow {S_W}^{-1} S_B w - J w = 0$$

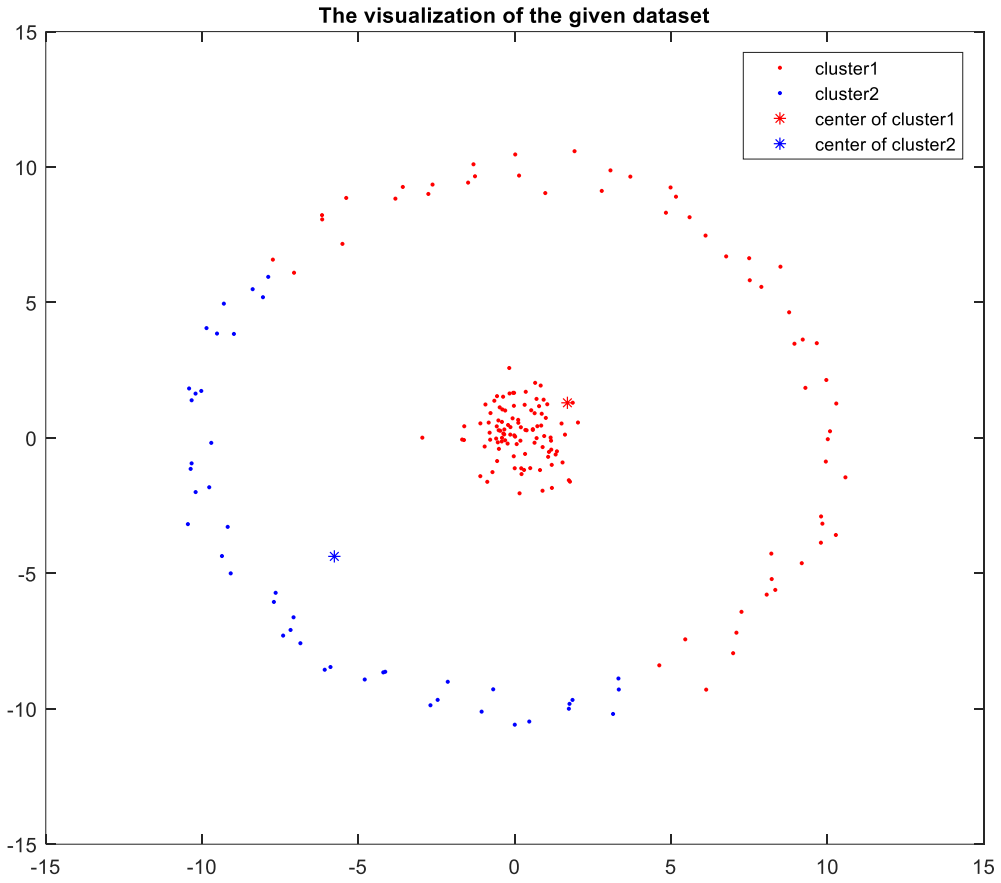
$\Rightarrow {S_W}^{-1} S_B w = J w$ is a generalized eigenvalue problem.

Here, to find the maximum $J(w)$, which is equivalent to find maximum eigenvalue, we need to choose the eigenvector corresponding to the maximum eigenvalue for w. Therefore, w is the first C – 1 eigen vectors of ${S_W}^{-1} S_B$.

2) Clustering

a)

In this part, I repeated 10000 times with random starts and picked the best in terms of accuracy. Graph shows the visualization of the given dataset in 2D.



bestConfusion is the confusion matrix of the picked result.

bestConfusion =

100 56

0 44

accuracy is the accuracy of the picked result. It is in percent.

accuracy =

72

Discussion: Since k-means algorithm's decision boundary is a line and the data is not linearly separable, this algorithm could not separate the data well.

b)

$$J(D) = \sum_{i=1}^{N}\sum_{k=1}^{K} \delta_{ik} \cdot \|\Phi(x_i) - m_k\|^2$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \delta_{ik} \cdot (\Phi(x_i)^T\Phi(x_i) - 2 \cdot \Phi(x_i)^T m_k + {m_k}^T m_k)$$

And

$$m_k = \frac{\sum_{i=1}^{N} \delta_{ik} \cdot \Phi(x_i)}{\sum_{i=1}^{N} \delta_{ik}}$$

Let's denote the number of data in class k as $N_k = \sum_{i=1}^{N} \delta_{ik}$, which is a constant.

$$\Rightarrow m_k = \frac{\sum_{i=1}^{N} \delta_{ik} \cdot \Phi(x_i)}{N_k}$$

$$\Rightarrow {m_k}^T = \left(\frac{\sum_{i=1}^{N} \delta_{ik} \cdot \Phi(x_i)}{N_k}\right)^T = \frac{\sum_{i=1}^{N} \delta_{ik} \cdot \Phi(x_i)^T}{N_k}$$

$$\Rightarrow J(D) = \sum_{i=1}^{N}\sum_{k=1}^{K} \delta_{ik} \left(\Phi(x_i)^T\Phi(x_i) - \frac{2}{N_k} \cdot \Phi(x_i)^T \left(\sum_{j_1=1}^{N} \delta_{j_1 k} \cdot \Phi(x_{j_1}) \right) \right.$$
$$\left. + \frac{1}{{N_k}^2} \left(\sum_{j_2=1}^{N} \delta_{j_2 k} \cdot \Phi(x_{j_2})^T \right) \left(\sum_{j_3=1}^{N} \delta_{j_3 k} \cdot \Phi(x_{j_3}) \right) \right)$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \delta_{ik} \left(\Phi(x_i)^T\Phi(x_i) - \frac{2}{N_k} \left(\sum_{j_1=1}^{N} \delta_{j_1 k} \cdot \Phi(x_i)^T\Phi(x_{j_1}) \right) \right.$$
$$\left. + \frac{1}{{N_k}^2} \left(\sum_{j_2=1}^{N}\sum_{j_3=1}^{N} \delta_{j_2 k}\, \delta_{j_3 k} \cdot \Phi(x_{j_2})^T\Phi(x_{j_3}) \right) \right)$$

We know that $K(i,j) = K(j,i) = \langle \Phi(x_i), \Phi(x_j) \rangle = \Phi(x_i)^T\Phi(x_j)$,

$$\Rightarrow J(D) = \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik}\left(K(i,i) - \frac{2}{N_k}\left(\sum_{j_1=1}^{N}\delta_{j_1k}\cdot K(i,j_1)\right) + \frac{1}{{N_k}^2}\left(\sum_{j_2=1}^{N}\sum_{j_3=1}^{N}\delta_{j_2k}\,\delta_{j_3k}\cdot K(j_2,j_3)\right)\right)$$

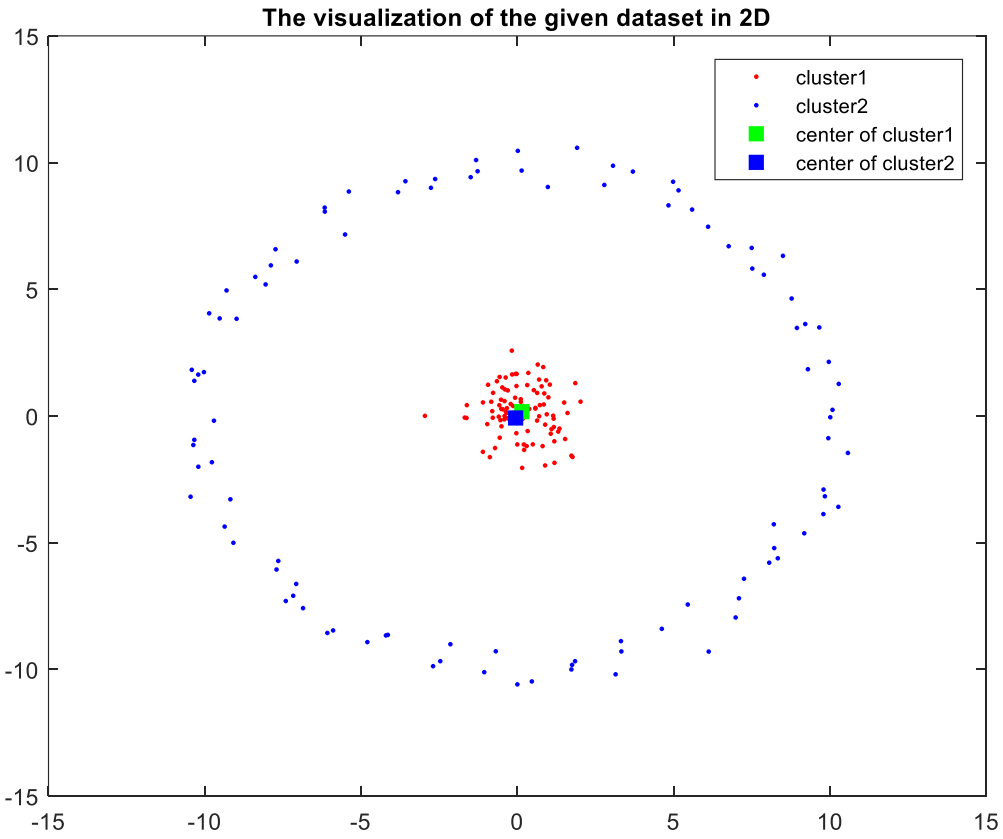
And if we put N_k 's into equation we get,

$$J(D) = \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik}\left(K(i,i) - \frac{2}{\sum_{n_1=1}^{N}\delta_{n_1k}}\left(\sum_{j_1=1}^{N}\delta_{j_1k}\cdot K(i,j_1)\right) + \frac{1}{(\sum_{n_2=1}^{N}\delta_{n_2k})^2}\left(\sum_{j_2=1}^{N}\sum_{j_3=1}^{N}\delta_{j_2k}\,\delta_{j_3k}\cdot K(j_2,j_3)\right)\right)$$

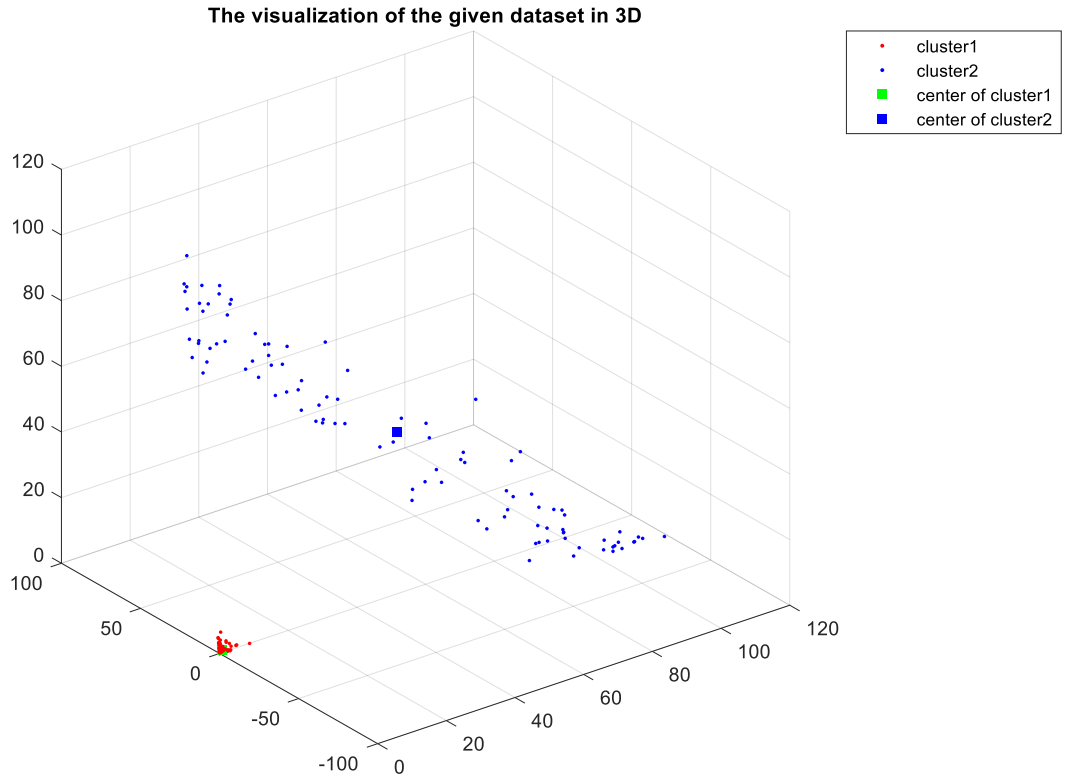
Note: I use j_1, j_2, j_3, n_1, n_2 for avoiding possible confusions.

c)

In this part, I repeated 10000 times with random starts and picked the best in terms of accuracy. First graph shows the visualization of the given dataset in 2D, whereas second graph show in 3D.



In the second graph, center of cluster 1 may not be seen clearly, since red points blocks it. But it is, among that points.



bestConfusion is the confusion matrix of the picked result.

```
bestConfusion =

   100     0
     0   100
```

accuracy is the accuracy of the picked result. It is in percent.

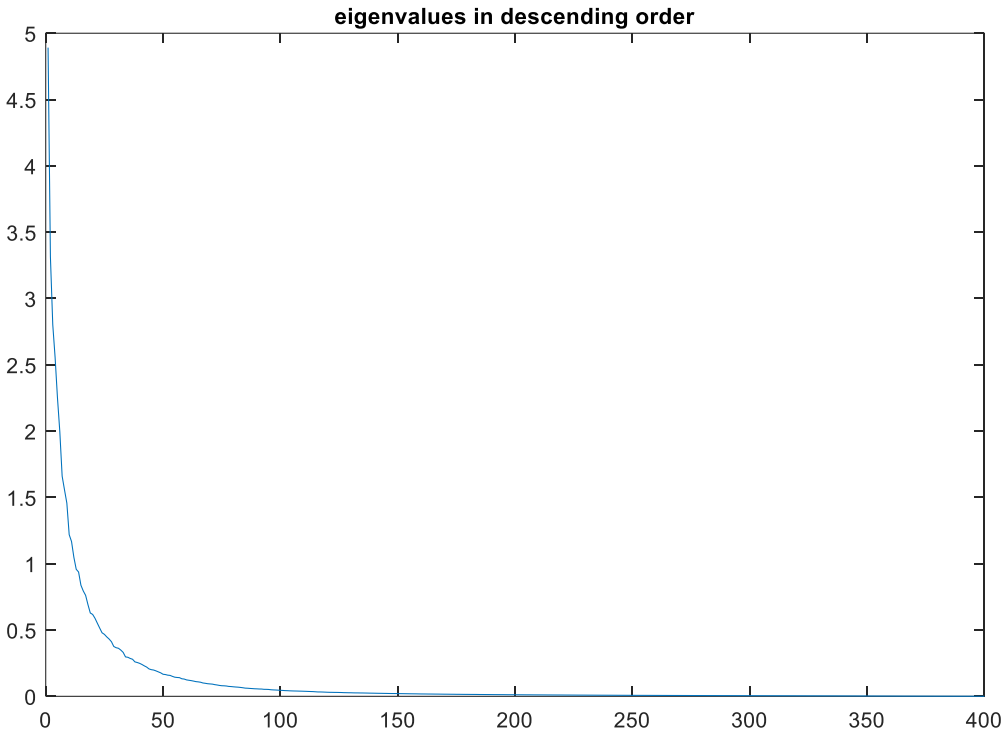
```
accuracy =

   100
```

Discussion: Using polynomial kernel with degree 2 gives much better results than k-mean algorithm and at best accuracy it gives 100% accuracy. This is because, the given dataset is not linearly separable in 2D space. However, when data is transformed into 3D space, it can be separable by a linear surface, easily. Therefore, kernel k-means algorithm, successfully separated the data.

3) Principle Component Analysis

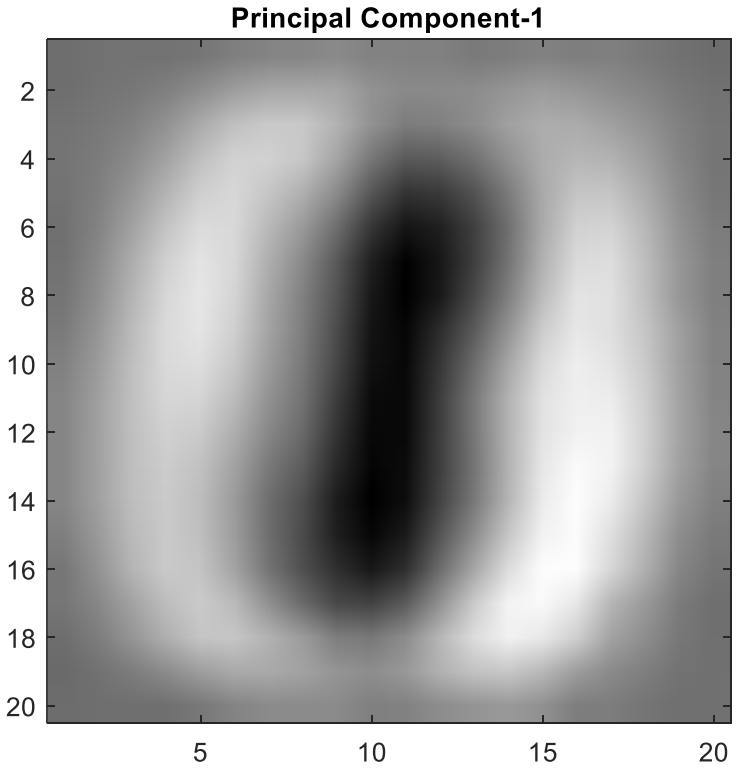
a) In this part, I used pca function. Graph shows the eigenvalues of corresponding principal components in descending order.

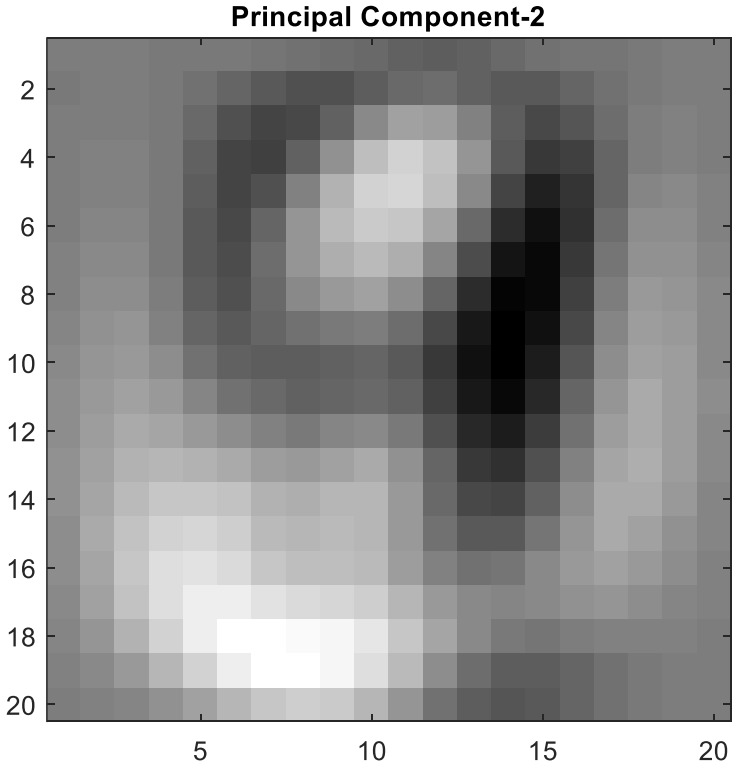


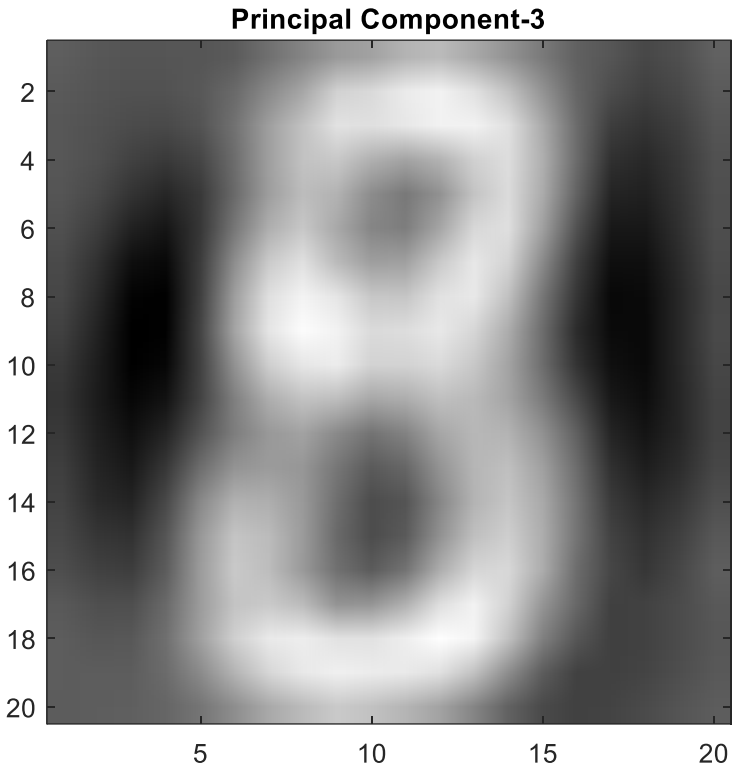
I decided on 50 as a number of principal component since near the first 50 principal component, there is an elbow shape. Moreover, first 50 principal component is enough to reconstruct the data.

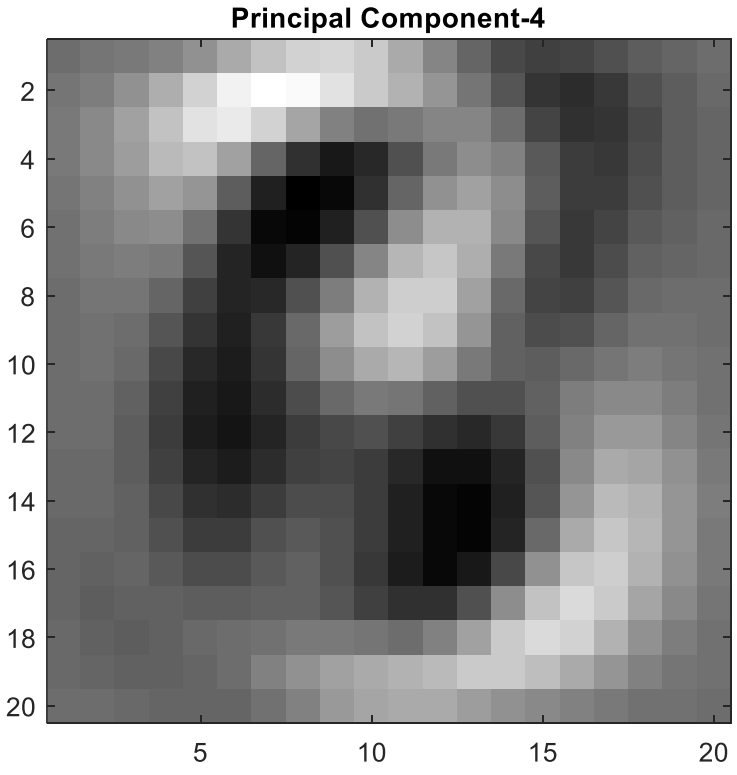
b)

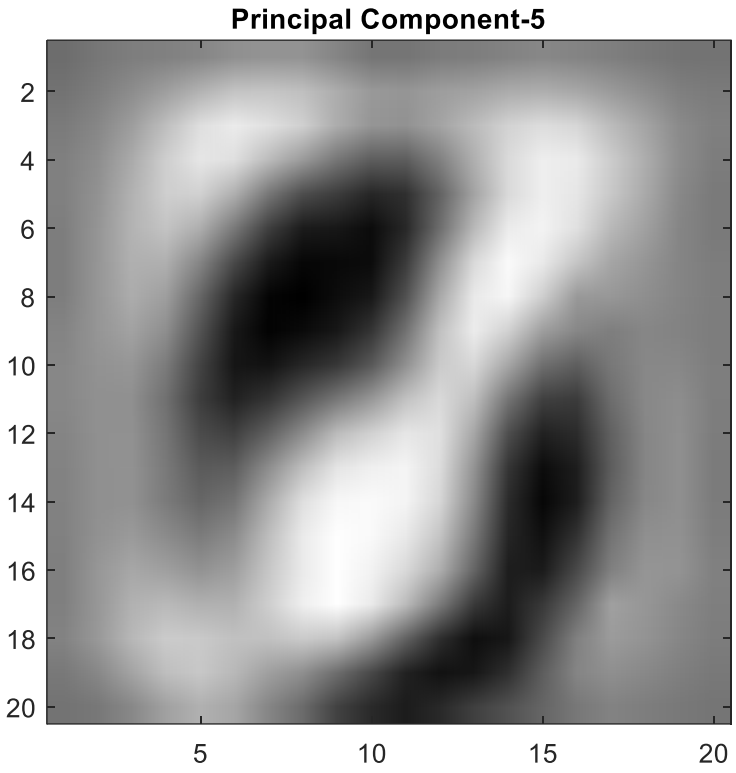
Top 5 Principal Component



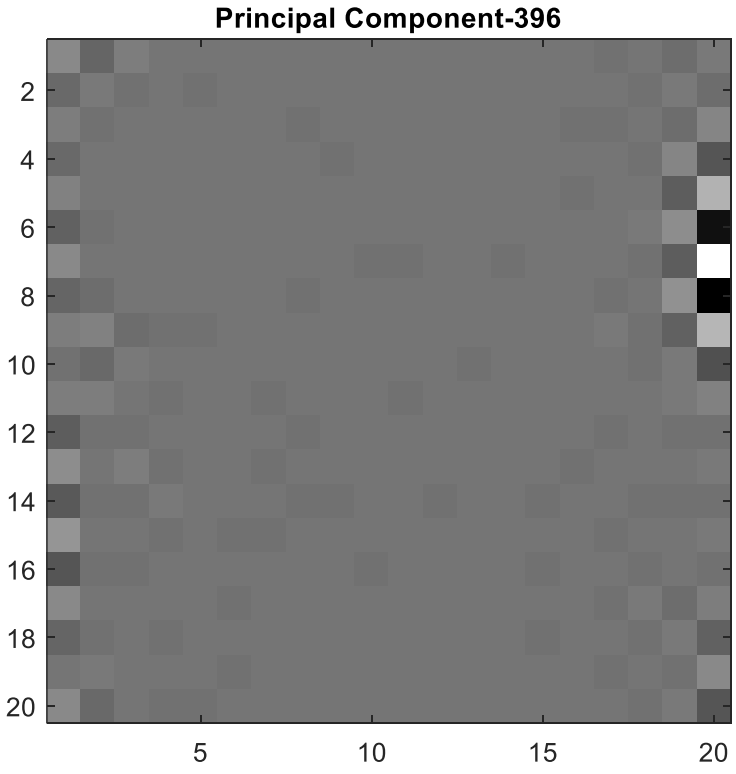


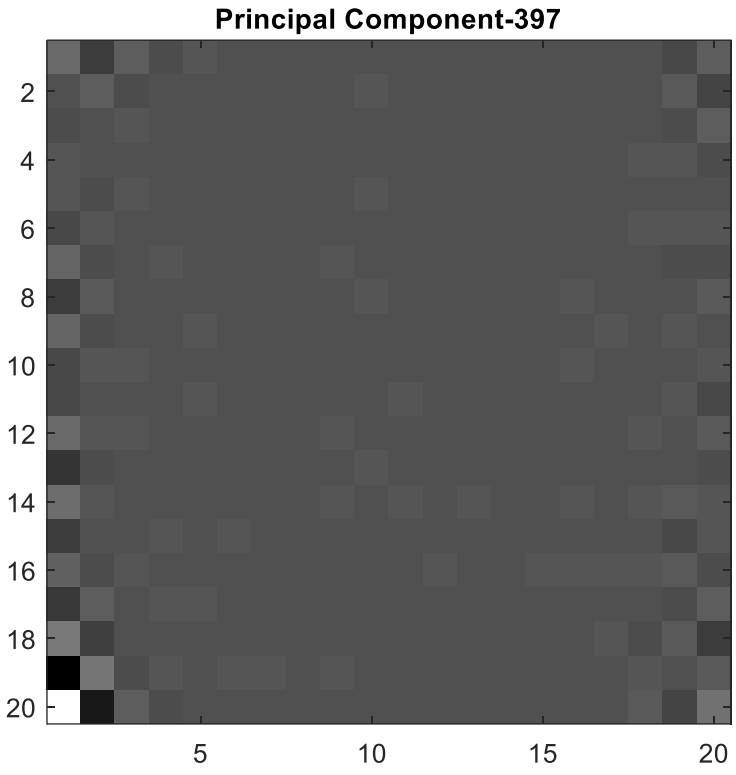

Principal Component-3
2
4
6
8
10
12
14
16
18
20
5
10
15
20

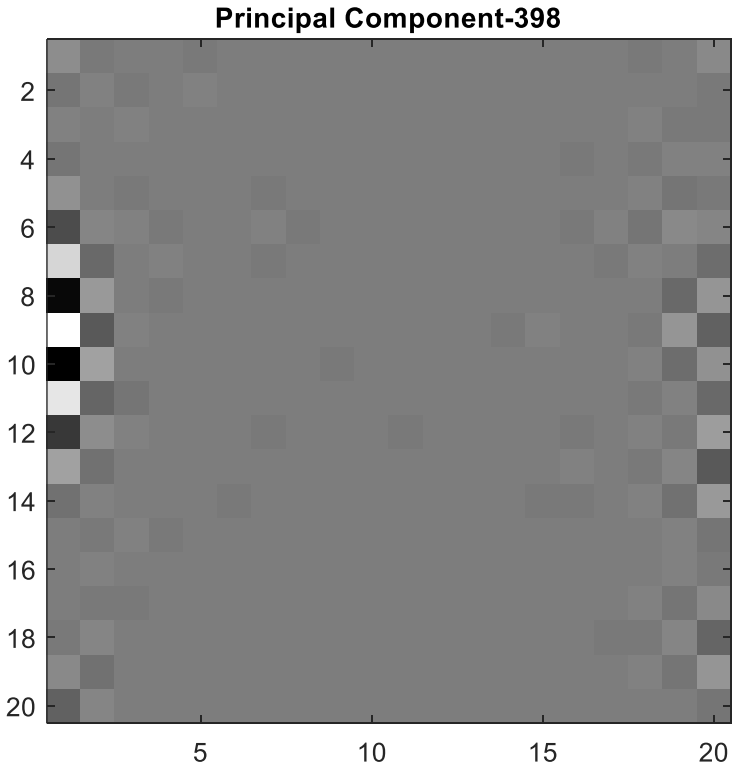

Principal Component-4
2
4
6
8
10
12
14
16
18
20
5
10
15
20

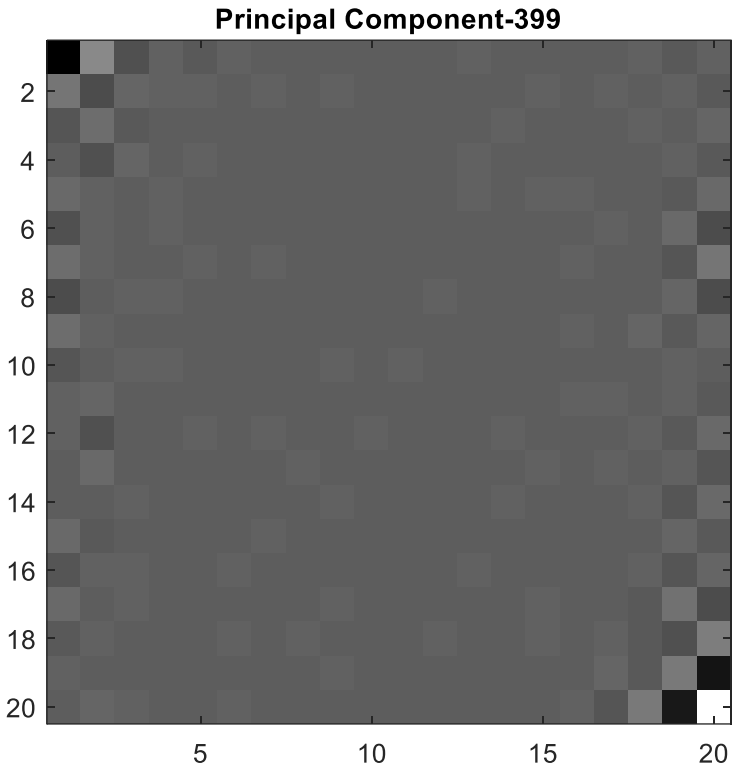


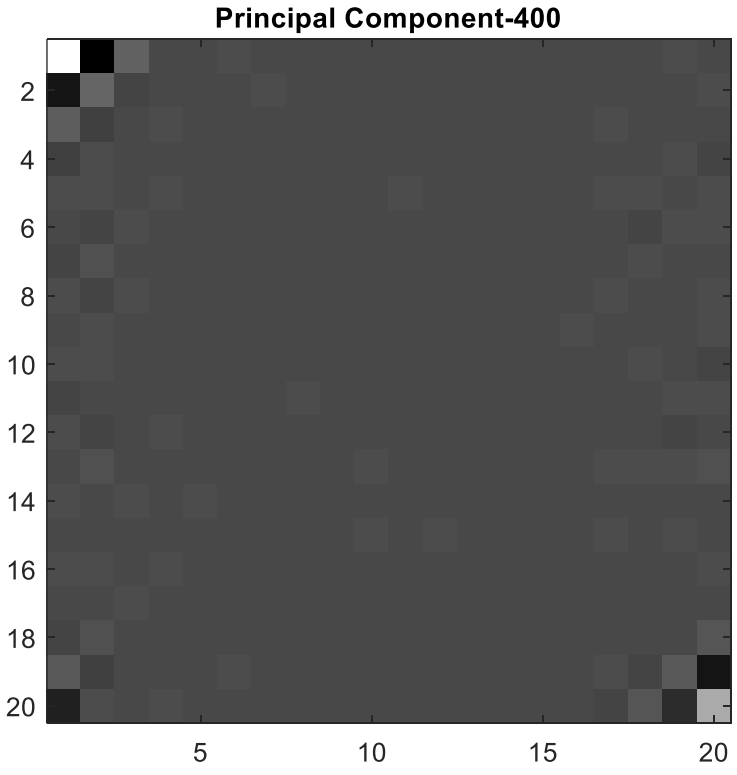
Bottom 5 Principal Component




Principal Component-397
2
4
6
8
10
12
14
16
18
20
5
10
15
20


Principal Component-398
2
4
6
8
10
12
14
16
18
20
5
10
15
20





Discussion: While top 5 principal component represents the pixels which have the most important role classifying the data, bottom 5 represents the least important ones.

APPENDIX

Code of question 2a

```
function [bestConfusion, accuracy] = hw3_q2a(dataPath)

    currPath = cd(dataPath);
    origData = load('clustering.csv');
    cd(currPath);

    bestCenters = zeros(2,2);
    bestConfusion = zeros(2,2);
    bestClass1 = [];
    bestClass2 = [];

    data = origData(:, 1:2);
    label = ones(size(origData(:,3)));

    for k = 1:10000
        centers =  20.*rand(2,2)-10;

        while centers(1,:) == centers(2,:)
            centers =  20.*rand(2,2)-10;
        end

        oldCenters = zeros(2,2);

        class1 = [];
        class2 = [];

        while oldCenters ~= centers
            distance1 = sum((data-centers(1, :)).^2,2);
            distance2 = sum((data-centers(2, :)).^2,2);

            label(distance2 >= distance1) = 1;
            label(distance2 < distance1) = 2;

            class1 = data(label==1, :);
            class2 = data(label==2, :);

            oldCenters = centers;

            centers(1,:) = sum(class1,1)/size(class1,1);
            centers(2,:) = sum(class2,1)/size(class2,1);
        end

        confusion = zeros(2,2);

        for i=1:size(origData(:, 3))
            confusion(label(i),origData(i, 3)) =
confusion(label(i),origData(i, 3))+1;
        end

        if
bestConfusion(1,1)+bestConfusion(2,2)<confusion(1,1)+confusion(2,2)
            bestConfusion = confusion;
            bestCenters = centers;
            bestClass1 = class1;
            bestClass2 = class2;
        end
    end
```

```
    accuracy = (bestConfusion(1,1)+bestConfusion(2,2))/2;

    figure;
    plot(bestClass1(:,1),bestClass1(:,2),'.r');
    hold on;
    plot(bestClass2(:,1),bestClass2(:,2),'.b');
    plot(bestCenters(1,1),bestCenters(1,2),'r*');
    plot(bestCenters(2,1),bestCenters(2,2),'b*');
    legend('cluster1', 'cluster2', 'center of cluster1', 'center of
cluster2');
    title('The visualization of the given dataset');
end
```

Code of question 2c

```
function [bestConfusion, accuracy] = hw3_q2c(dataPath)
    currPath = cd(dataPath);
    origData = load('clustering.csv');
    cd(currPath);

    bestCenters = zeros(2,2);
    bestCenters3d = zeros(2,3);
    bestConfusion = zeros(2,2);
    bestClass1 = [];
    bestClass2 = [];

    data = [origData(:,1).^2 sqrt(2)*origData(:,1).*origData(:,2)
origData(:,2).^2];
    label = ones(size(origData(:,3)));

    for k = 1:10000
        centers2d =  20.*rand(2,2)-10;

        while centers2d(1,:) == centers2d(2,:)
            centers2d =  20.*rand(2,2)-10;
        end

        oldCenters = zeros(2,3);

        centers = [centers2d(:,1).^2 sqrt(2)*centers2d(:,1).*centers2d(:,2)
centers2d(:,2).^2];

        while oldCenters ~= centers
            distance1 = sum((data-centers(1, :)).^2,2);
            distance2 = sum((data-centers(2, :)).^2,2);

            label(distance2 >= distance1) = 1;
            label(distance2 < distance1) = 2;

            class1 = data(label==1, :);
            class2 = data(label==2, :);

            class1_2d = origData(label==1, 1:2);
            class2_2d = origData(label==2, 1:2);

            oldCenters = centers;

            centers(1,:) = sum(class1,1)/size(class1,1);
            centers(2,:) = sum(class2,1)/size(class2,1);

            centers2d(1,:) = sum(class1_2d,1)/size(class1_2d,1);
            centers2d(2,:) = sum(class2_2d,1)/size(class2_2d,1);
```

```
        end

        confusion = zeros(2,2);

        for i=1:size(origData(:, 3))
            confusion(label(i),origData(i, 3)) =
confusion(label(i),origData(i, 3))+1;
        end

        if
bestConfusion(1,1)+bestConfusion(2,2)<confusion(1,1)+confusion(2,2)
            bestConfusion = confusion;
            bestCenters = centers2d;
            bestCenters3d = centers;
            bestClass1 = class1_2d;
            bestClass2 = class2_2d;
        end
    end

    accuracy =  (bestConfusion(1,1)+bestConfusion(2,2))/2;

    figure;
    plot(bestClass1(:,1),bestClass1(:,2),'.r');
    hold on;
    plot(bestClass2(:,1),bestClass2(:,2),'.b');
    plot(bestCenters(1,1),bestCenters(1,2),'s', 'MarkerSize', 7,
'MarkerEdgeColor', 'g', 'MarkerFaceColor', 'g');
    plot(bestCenters(2,1),bestCenters(2,2),'s', 'MarkerSize', 7,
'MarkerEdgeColor', 'b', 'MarkerFaceColor', 'b');
    legend('cluster1', 'cluster2', 'center of cluster1', 'center of
cluster2');
    title('The visualization of the given dataset in 2D');

    figure;
    scatter3(bestClass1(:,1).^2, sqrt(2)*bestClass1(:,1).*bestClass1(:,2),
bestClass1(:,2).^2 ,'.r');
    hold on;
    scatter3(bestClass2(:,1).^2, sqrt(2)*bestClass2(:,1).*bestClass2(:,2),
bestClass2(:,2).^2 ,'.b');
    scatter3(bestCenters3d(1,1), bestCenters3d(1,2),bestCenters3d(1,3),
'filled', 'gs');
    scatter3(bestCenters3d(2,1), bestCenters3d(2,2),bestCenters3d(2,3),
'filled', 'bs');
    legend('cluster1', 'cluster2', 'center of cluster1', 'center of
cluster2');
    title('The visualization of the given dataset in 3D');
end
```

Code of question 3a

```
function [eigenvalues] = hw3_q3a (dataPath)
    currPath = cd(dataPath);
    data = importdata('digits.csv');
    cd(currPath)

    sample = data(:,1:end-1);
    [~,~,eigenvalues] = pca(abs(sample));

    figure;
    plot(eigenvalues);
    title('eigenvalues in descending order');
end
```

Code of question 3b

```
function hw3_q3b (dataPath)
    currPath = cd(dataPath);
    data = importdata('digits.csv');
    cd(currPath)

    sample = data(:,1:end-1);
    N = size(sample,2);
    [digits,~,~] = pca(abs(sample));

    for i = [1:5 N-4:N]
        I = digits( :,i );
        figure, imagesc( reshape( I, 20, 20 ) );
        colormap( gray );
        axis image;
        title(strcat('Principal Component-', num2str(i)))
    end
end
```