Math 444: Homework 7

Iris Zhang

April 19, 2020

Problem 1

The Matlab code for $6X6$ Grey level Co-occurrence Matrices (GLCM) corresponding to the pairs of offset parameters (2,2),(3,0),(1,4) is attached in the appendix I.

Problem 2

Now we want to see whether the images cluster in a natural way, we use the k-medoids algorithm developed in homework 2 for different data matrix that we formed by using the GLCM for each image in the library with three different pairs of offset parameters.
First, we develop the data matrix $X_{36\mathbb{X}64}$ using the GLCM composed by applying offset parameters (2,2). We try k-medoids with k = 2,3,4,5 and plot the first four attributes, see Figure 1,2,3, and 4.

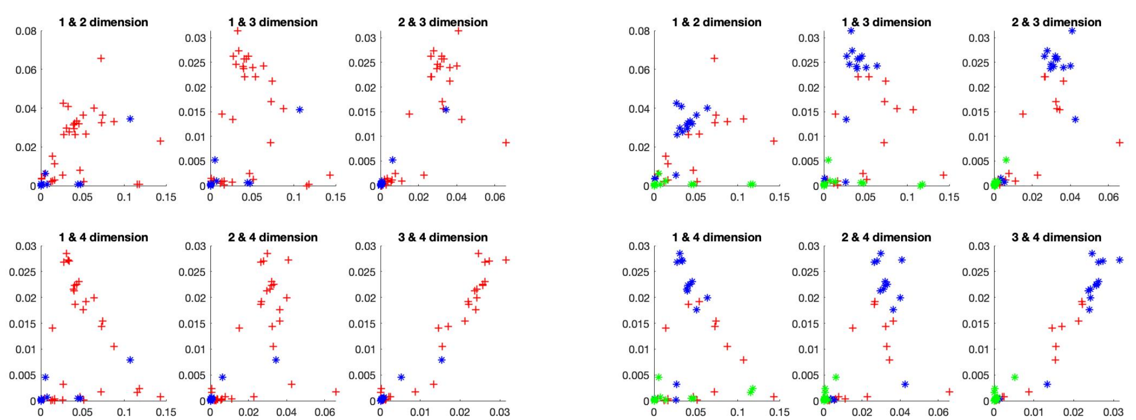

Figure 1: first few attributes of GLCM(2,2) after doing k-medoids with k=2 (left) and k=3 (right)

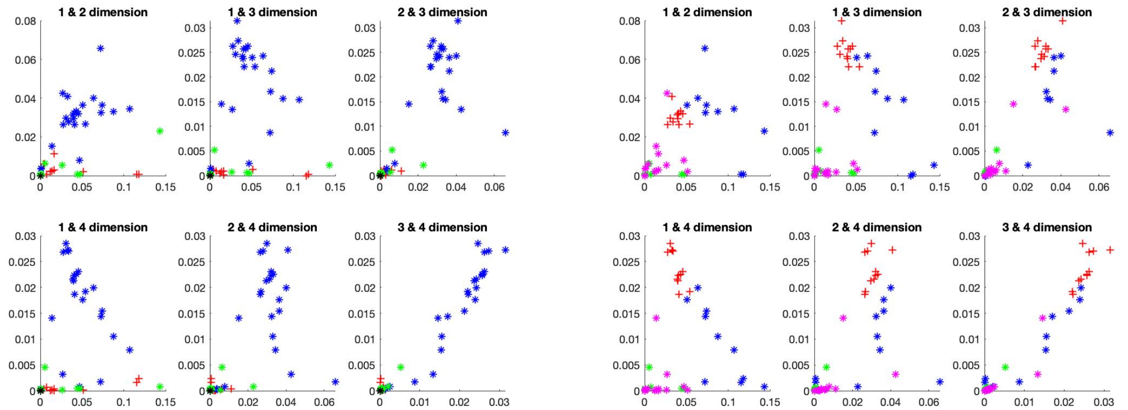

Figure 2: first few attributes of GLCM(2,2) after doing k-medoids with k=4 (left) and k=5 (right)

Then, we develop another data matrix $X_{36\mathbb{X}64}$ using the GLCM composed by applying offset parameters (3,0). We also try k-medoids with k = 2,3,4,5 and plot the first four attributes, see Figure 5,6,7, and 8.

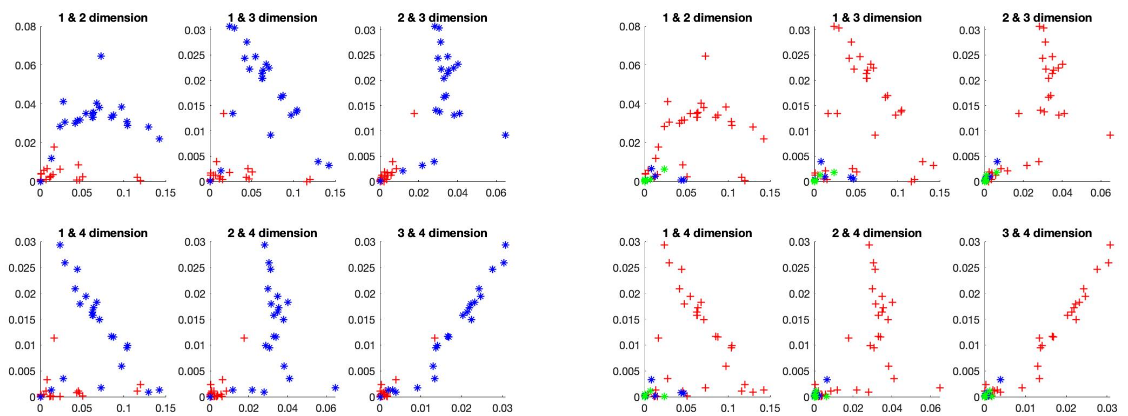

Figure 3: first few attributes of GLCM(3,0) after doing k-medoids with k=2 (left) and k=3 (right)

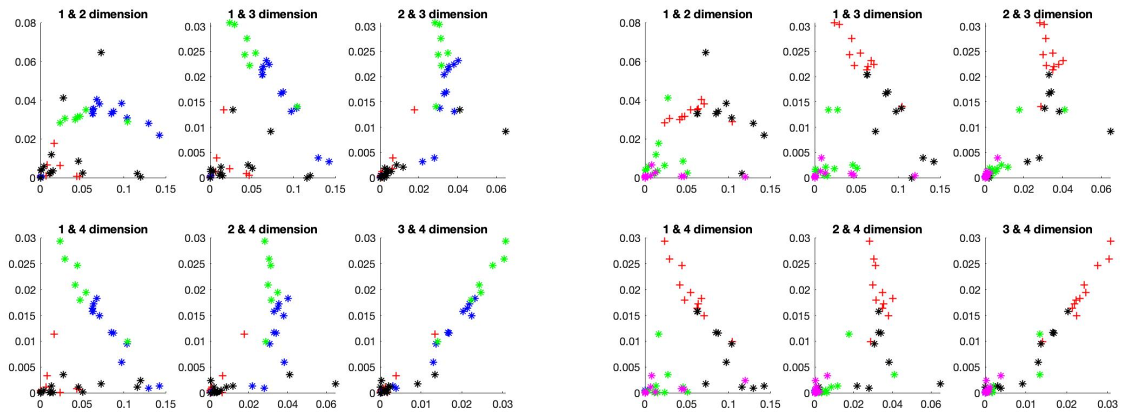

Figure 4: first few attributes of GLCM(3,0) after doing k-medoids with k=4 (left) and k=5 (right)

Then, we develop another data matrix $X_{36\mathbb{X}64}$ using the GLCM composed by applying offset parameters (1,4). We also try k-medoids with k = 2,3,4,5 and plot the first four attributes, see Figure 9,10,11, and 12.

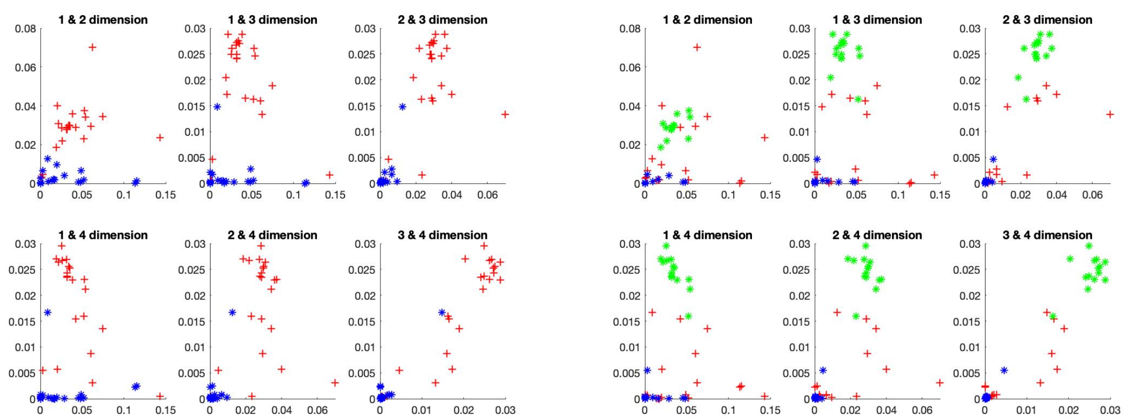

Figure 5: first few attributes of GLCM(1,4) after doing k-medoids with k=2 (left) and k=3 (right)

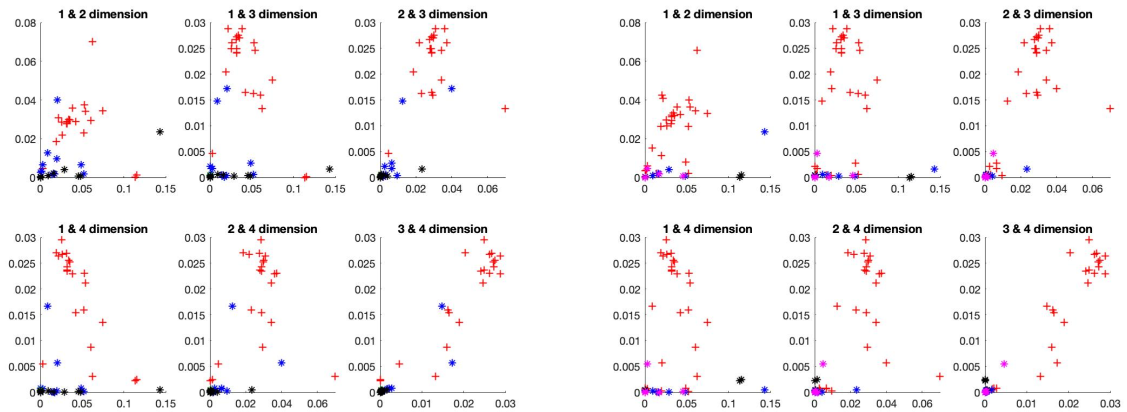

Figure 6: first few attributes of GLCM(1,4) after doing k-medoids with k=4 (left) and k=5 (right)

From above, we can see that the separation of clusters are not clear. Since it is a really high dimensional data set, the selected attributes may not be clear enough, so we apply LDA algorithm to see whether there are separations between clusters. Then, we plot the histogram for the first three directions with different values of k for each GLCM matrix, see Figure 7,8,9.

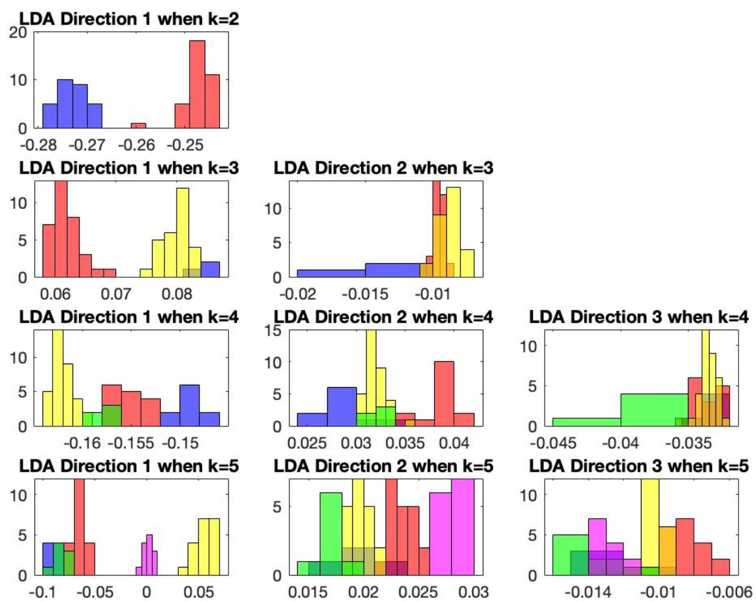

Figure 7: Histogram after LDA for GLCM(2,2)

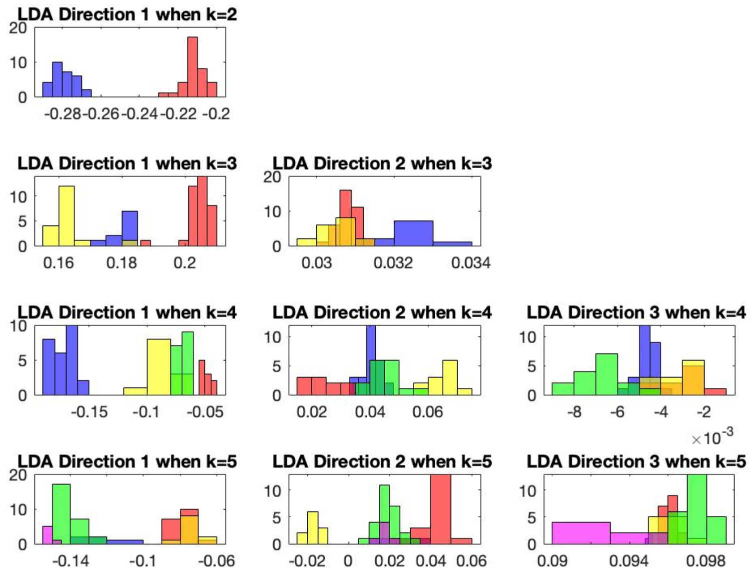

Figure 8: Histogram after LDA for GLCM(3,0)

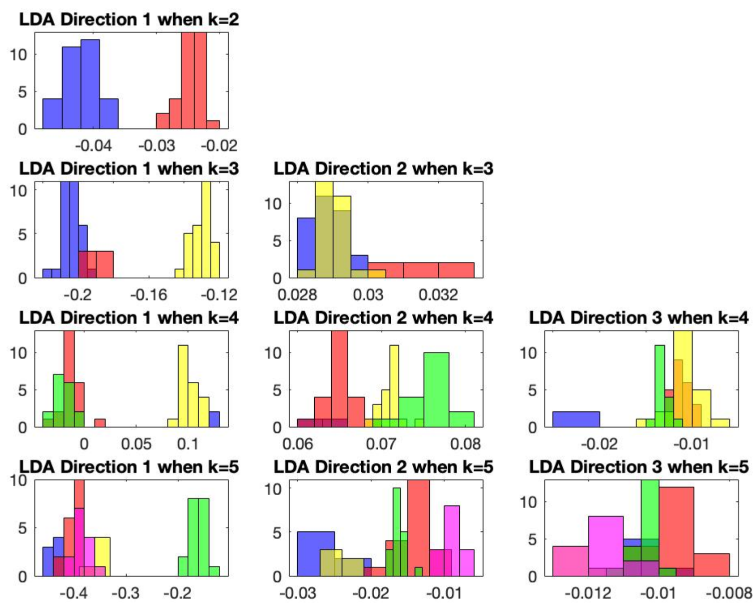

Figure 9: Histogram after LDA for GLCM(1,4)

From the histograms above, we can see that when k=2, for the first direction of LDA, the two clusters are completely separately from each other for all three GLCM matrices. Moreover, in general, the clusters are consistent.

Problem 3

For this problem, we combine the data matrices we have in problem 2 into one matrix $\mathbf{Data}_{108\mathbb{X}64}$. Then, we apply k-medoids with different values of k as before and plot the first few attributes, see Figure 10 and 11.

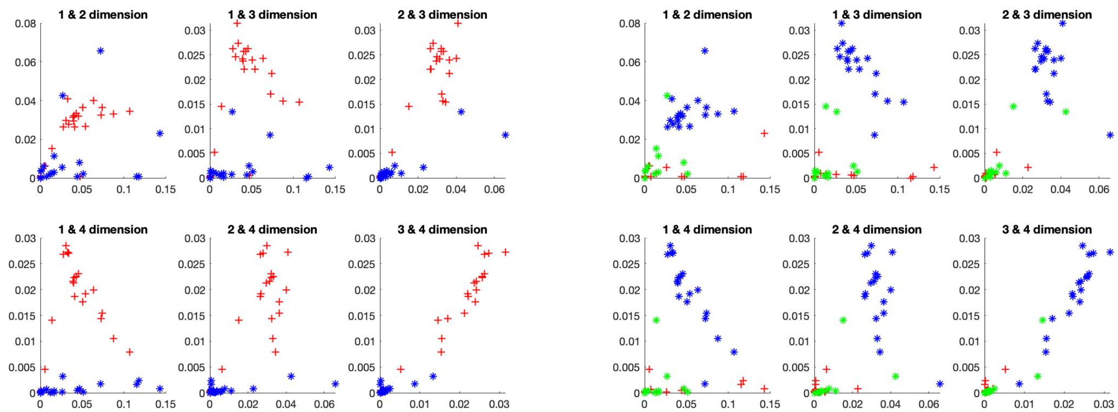

Figure 10: first few attributes of **Data** after doing k-medoids with k=2 (left) and k=3 (right)

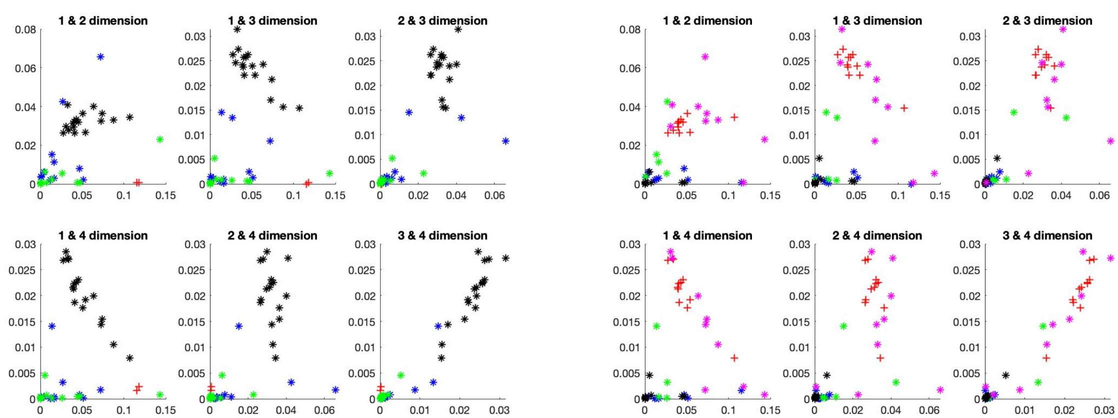

Figure 11: first few attributes of **Data** after doing k-medoids with k=4 (left) and k=5 (right)

Now we do the LDA for the **Data** matrix to see whether there are separations between clusters. The histogram plots see Figure 12.

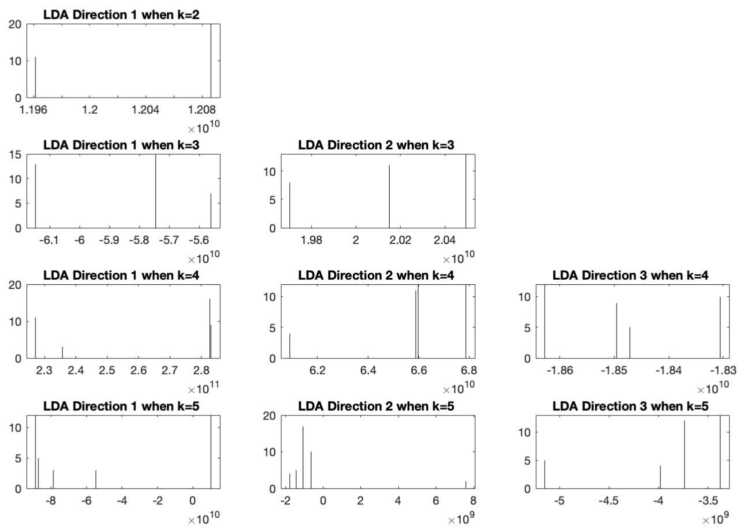

Figure 12: Histogram after LDA for **Data**

We can see that the histogram we obtained only shows lines instead of bins, which is due to the relative small binsize compared to the range. It also implies that the separations are really clear, especially when k=2. Therefore, if we do the k-medoids with 2 clusters, Figure 10 suggests that the fourth dimension is more crucial in determining the clusters.

Appendix I : Matlab code for 6X6 GLCM with offset parameters (2,2), (3,0), (1,4)

```
function [G1,G2,G3] = GLCM(X)
A_before = double(X);
Max = max(A_before, [], 'all');
A = A_before/Max;
[n1,n2]=size(A);
% define coarsened gray level matrix C
C = zeros(n1,n2);
for i=1:n1
    for j=1:n2
        C(i,j)=ceil(A(i,j)*6);
    end
end

% Define the Gray Level Co-occurence Matrix G
% (2,2)
C_shift1 = NaN(n1,n2);
C_shift1(1:n1-2,1:n2-2)= C(1+2:n1,1+2:n2);
G1 = zeros(6,6);
for i = 1:6
     Ii=(C==i);
    for j =1:6
       Ij = (C_shift1==j);
       G1(i,j)=sum(sum(Ii.*Ij));
    end
end
S1 = sum(G1,'all');
G1 = G1/S1;
% (3,0)
C_shift2 = NaN(n1,n2);
C_shift2(1:n1-3,1:n2-0)= C(1+3:n1,1+0:n2);
G2 = zeros(6,6);
for i = 1:6
     Ii=(C==i);
    for j =1:6
       Ij = (C_shift2==j);
       G2(i,j)=sum(sum(Ii.*Ij));
    end
end
S2 = sum(G2,'all');
G2 = G2/S2;
% (1,4)
C_shift3 = NaN(n1,n2);
C_shift3(1:n1-1,1:n2-4)= C(1+1:n1,1+4:n2);
G3 = zeros(6,6);
for i = 1:6
     Ii=(C==i);
    for j =1:6
       Ij = (C_shift3==j);
       G3(i,j)=sum(sum(Ii.*Ij));
    end
```

```
end
S3 = sum(G3,'all');
G3 = G3/S3;

end
```