CS188, Winter 2017
Problem Set 4:
Due 3/16/2017

Shannon Phu

1 Problem 1

(a) The average face is very generic and not that distinct. This implies that the underlying dataset has a lot of variance. Because the face is quite blurry, we can think that the 19 faces provided are as a whole different.

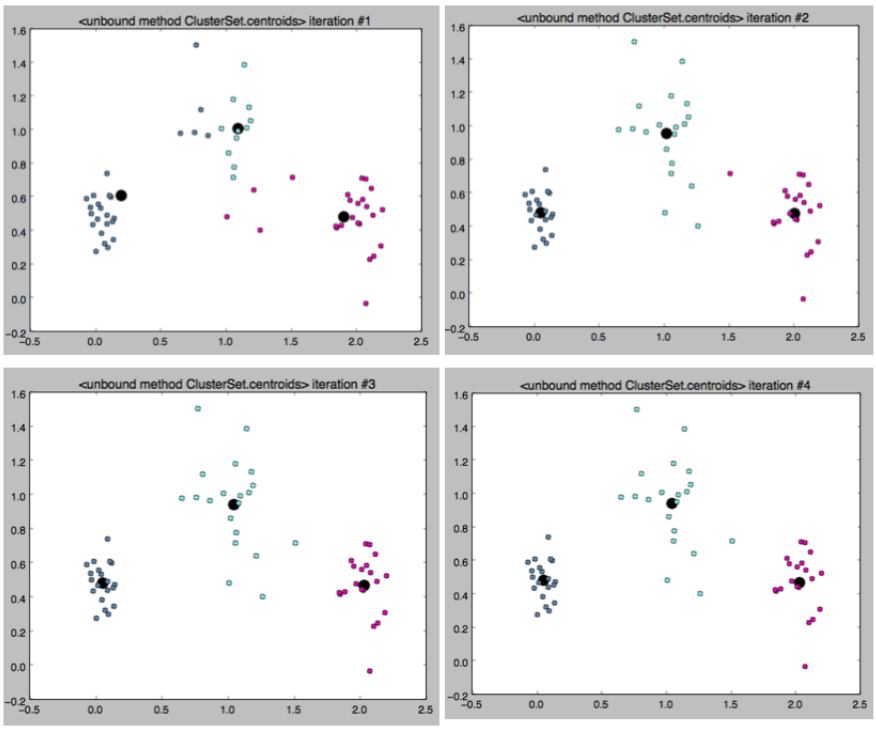
(b) These eigenfaces are chosen through PCA to avoid keeping correlated data. Thus we can think of these eigenfaces as representing the best independent eigenvectors to be combined into any possible face in our dataset.

(c) As the number of components, l, increases then the images become sharper. The images with low l start off quite blurry then become more clear as more components are kept in PCA.

2 Problem 2

(a) Having a variable number of clusters is a bad idea because then to get the minimal error, k will equal to n (total number of samples). If we have 1 number of samples per cluster, then over fitting will occur to minimize the error to result in an error of 0. In this case where k = n, c will equal x and u will also equal x.

(d)



The centroids for each iteration:

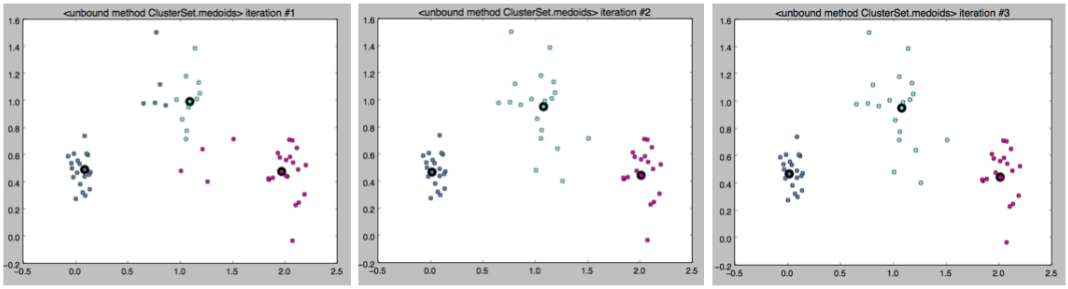
i. ([1.06194805 0.7757108], [2.]);
([1.2102522 0.63854748], [2.]);
([0.64950668 0.97477045], [2.]);

ii. ([1.089668 1.00424282], [2.]);
([1.90010446 0.48090448], [3.]);
([0.19298238 0.60641572], [1.]);

iii. ([1.01605529 0.95288767], [2.]);
([2.00594139 0.47723895], [3.]);
([0.04917974 0.4810944], [1.]);

iv. ([1.04063507 0.9409604], [2.]);
([2.03085592 0.46538378], [3.]);
([0.04917974 0.4810944], [1.]);

(e) The image set as follows show the progression of kmeans for medoids for each iteration.



Medoids per iteration:

i. ([1.06194805 0.7757108], 2);
([1.2102522 0.63854748], 2);
([0.64950668 0.97477045], 2);

ii. ([1.08850508 0.99112174], 2);
([1.96941007 0.47267369], 3);
([0.08637173 0.48779084], 1);

iii. ([1.07699221 0.94787531], 2);
([2.01197635 0.44000531], 3);
([0.0124713 0.46772052], 1);

(f) The following image sets show the progression of kmeans and kmedoids when the cheat initializer is used. I noticed that kMedoids converged faster than kMeans did because the initial center point per cluster was closer to the actual center when initializing with the medoid. But overall cheating on initialization causes convergence much faster.

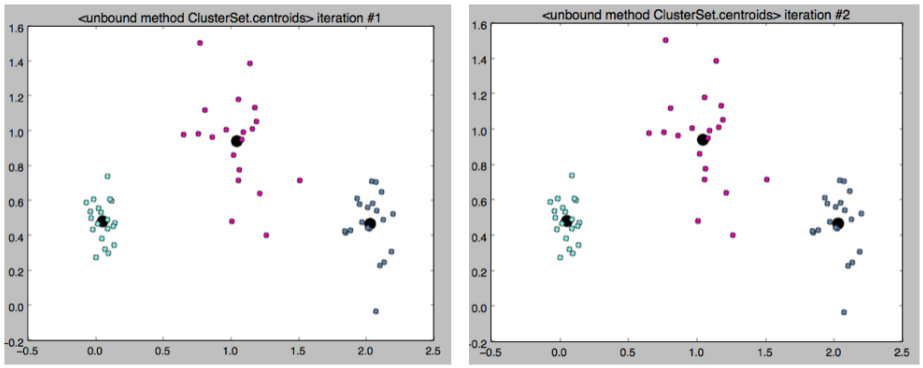
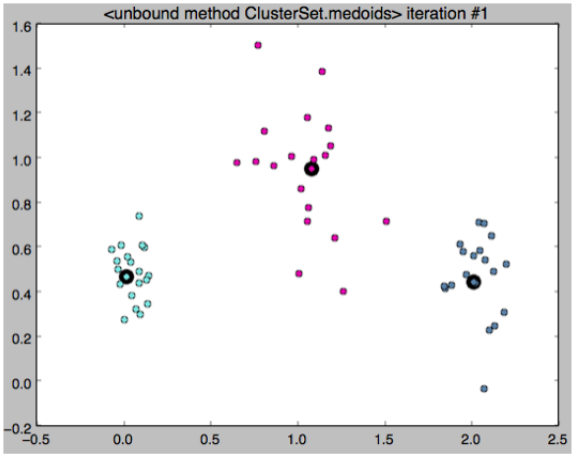


Figure 1: kMeans with 'cheat' initialization

i. ([0.0124713 0.46772052], [1.]);
([1.07699221 0.94787531], [2.]);
([2.01197635 0.44000531], [3.]);

ii. ([0.04917974 0.4810944], [1.]);
([1.04063507 0.9409604], [2.]);
([2.03085592 0.46538378], [3.]);

As for kMedoids with 'cheat' initialization:



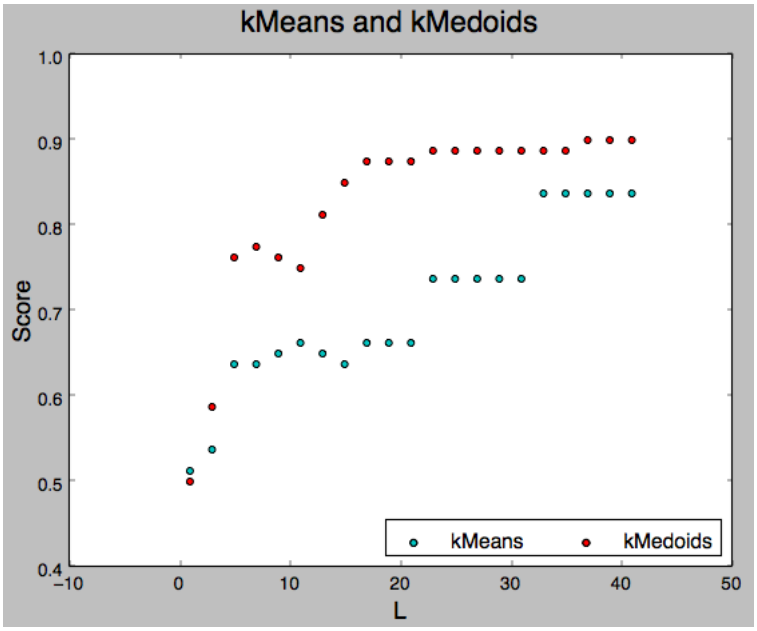
i. ([0.0124713 0.46772052], 1);
([1.07699221 0.94787531], 2);
([2.01197635 0.44000531], 3);

3 Problem 3

(a) kMeans has a higher average score than kMedoids. KMeans should also have a faster runtime because it is computationally simpler than kMedoids.

	average	min	max
k-means	0.66125	0.5875	0.775
k-medoids	0.60375	0.58125	0.61875

(b) The score for both kMeans and kMedoids increases as the number of principal components increases. But the performance of kMeans is lower than kMedoids when PCA is applied. This implies that kMedoids is better at clustering data that had PCA applied to it.



(c) The most discriminative pair was between classes 9 and 16 where the score was 0.9875. This makes sense because the two faces have quite different features:

The least discriminative pair was between classes 4 and 5 where the score was 0.5125. This also makes sense because the two faces are quite similar although the one on the right shows the side of the man's face: