Name: Kien Duc Vu

GtID: 903552166

GT Account: kvu41

ISYE 6740 – Computational Data Analysis

Final Exam – Spring 2020

I will obey GT Honor Code. I have neither given nor received any unauthorized aid on this exam. I understand that the work contained herein is wholly my own without the aim from a 3rd person.

Problem 1:

a)

Points	A(-3,-1)	B(2,1)	Cluster
I(2,2)	8	1	B
II(-1, 1)	4	3	B
III(3,1)	8	1	B
IV(0,-1)	3	4	A
V(-2,-2)	2	7	A

b) New cluster assignment: A (-1, -1.5) , B (4/3, 4/3)

c) It will not terminate in one step.

Problem 2:

Spectral clustering: data points as nodes of a connected graph and clusters are found by partitioning Laplacian graph, based on its spectral decomposition, into subgraphs. Spectral clustering embeds shortest path distance into embedding space and use K-Means in embedding space.

• K-means clustering: divide the objects into k clusters such that some metric relative to the centroids of the clusters is minimized. K-mean uses coordinate distance to measure how close 2 points are.

b)

Similarity:

- In both algorithm, we end up finding eigenvectors. Both are unsupervised algorithms.

Differences:

- PCA is done on a covariance or correlation matrix, but spectral clustering can take any similarity matrix (e.g. built with cosine similarity) and find clusters there.
- Second, spectral clustering algorithms are based on graph partitioning of Laplacian Graph (usually it's about finding the best cuts of the graph), while PCA finds the directions that have most of the variance.
- PCA and spectral clustering serve different purposes: one is a dimensionality reduction technique and the other is more an approach to clustering.

c)

Spectral clustering with single linkage distance. The dataset has clear clustering pattern of manifold.

Problem 3:

a) X =

```
array([[ 1. ,  0. ,  0.5],
       [ 6. , 14. ,  3. ],
       [11. , 28. ,  5.5],
       [ 7. , 21. ,  3.4]])
```

Mean_X = array ([6.25, 15.75, 3.1])

Covariance matrix =

```
array([[  0.25      ,  -2.         ,  -4.25      ,  -3.5       ],
       [ -2.        ,  32.33333333,  66.66666667,  52.73333333],
       [ -4.25      ,  66.66666667, 137.58333333, 108.96666667],
       [ -3.5       ,  52.73333333, 108.96666667,  86.45333333]])
```

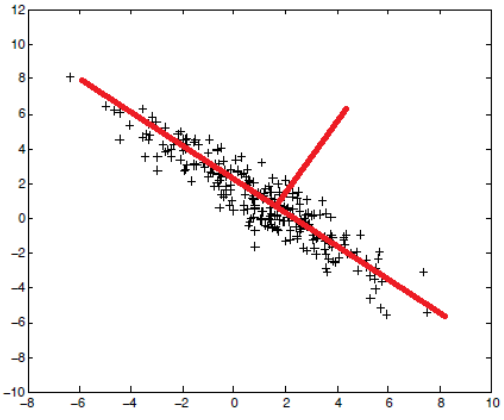
1st largest eigen values of covariance matrix X = 2.56e+02

Corresponding eigen vector = [-0.0228, 0.3548, 0.7325, 0.5804]

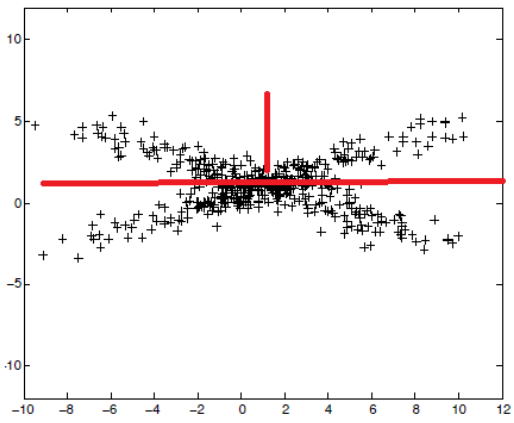
b) First principal component of 4 datapoints = [-0.866, 2.59807, -2.598, 0.866]

c) First principal component explains 0.9963 variance in our dataset (with variance = 163.44). Thus, the remaining variance in the dataset is ~ 0.

d)



(a)



(b)

Problem 4:

a) – AdaBoost, Decision Tree, K-NNs, Logistic Regression, Kernel Density Estimation, Naïve Bayes

b) – Left Image: SVM. Decision boundary is a linear hyperplane

- Middle Image: Decision Tree. Non-linear decision boundary. The decision boundary is sharp.
- Random Forest. Non-linear decision boundary. The decision boundary is similar to Decision Tree but fuzzier due to average mechanism

c) The statement is false. The misclassified points has increasingly greater weights than previous one, given that the misclassification error decreases over t.

Problem 5:

a) $0<h<1$ or $h > 4$.

b) Yes. When $h >4$, the decision boundary will change the direction.

c) From the SVM optimization formulation, for a given hyperplane w,b , any data points beyond the margin boundary will always satisfy the constrains, thus, will be discarded in calculation.

Problem 6:

a) After the first iteration with h1, the weight D2(i) for 3 correctly classified (+) points and 5 correctly classified(-) points will go down, and the weight D2(i) for the misclassified (+) point will go up.

b) E = 0.125

Alpha1 = 0.5 * ln[(1-E)/E) = 0.973

D2(i) for correctly classified points = D1(i) * e^(-alpha1)/(scaling_factor) = 0.07147

D2(i) for misclassified point = 0.4997

c) True. As the training processing goes on, the misclassification decreases, thus, the voting weights for weak classifier increases.

Problem 7:

- ridge regression:

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} (y_i - \boldsymbol{\theta}^T \boldsymbol{x})^2$$

$$\text{s.t.} \quad \|\boldsymbol{\theta}\|_2^2 \leq c = \frac{1}{\lambda}$$

  Lagragian function:

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} (y_i - \boldsymbol{\theta}^T \boldsymbol{x})^2 + \lambda \|\boldsymbol{\theta}\|_2^2$$

  decision variables: $\boldsymbol{\theta}, \lambda$,
  data: $y_i, \boldsymbol{x}_i$

- Lasso

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} (y_i - \boldsymbol{\theta}^T \boldsymbol{x})^2$$

$$\text{s.t.} \quad \|\boldsymbol{\theta}\|_1 \leq c = \frac{1}{\lambda}$$

  Lagragian function:

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} (y_i - \boldsymbol{\theta}^T \boldsymbol{x})^2 + \lambda \|\boldsymbol{\theta}\|_1$$

  decision variables: $\boldsymbol{\theta}, \lambda$,
  data: $y_i, \boldsymbol{x}_i$

b) Left Image: Ridge Regression since the coefficients never go to 0 as lambda increases.

Right Image: Lasso Regression since the coefficients has tendency to go to 0 as lambda increases.

c)

-Ridge Regression: Using L2-norm, pushes the parameters towards origin

-Lasso Regression: Using L1-norm, pushes the parameters towards part of coordinates.

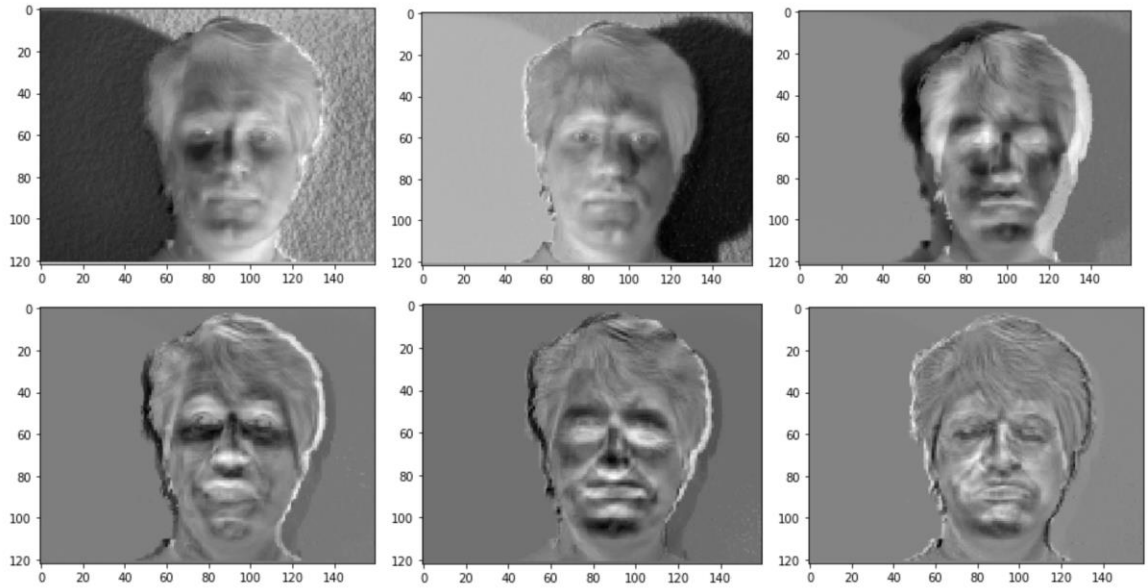
We need Lasso Regression for variable selection.

d)

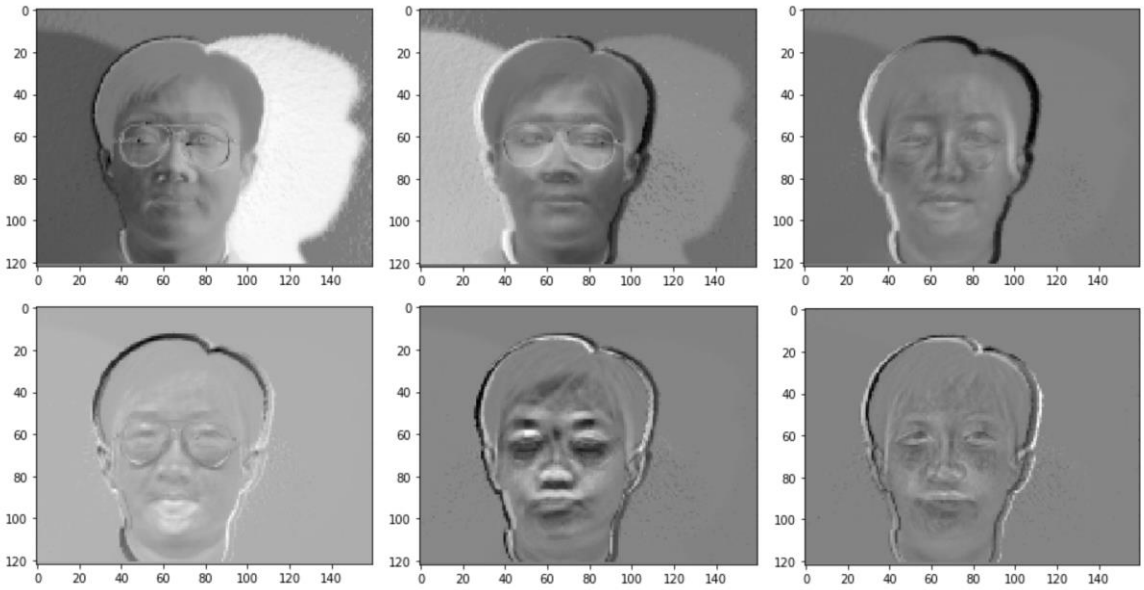
We can use cross-validation method to find the proper regularization parameter lambda.

Problem 8:

a) Top 6 eigenfaces:



Subject 14:



Eigenface contains low dimensional information that explains the most of the face images variance.

2. Projection:

- Test1 -> eigenface 1: 1.9e5, normalized score = 0.8996

-Test 1 -> eigenface 14: 6.4e4, normalized score = 0.2856

-Test 14 -> eigenface 1: 5.1e4, normalized score = 0.2252

-Test 14 -> eigenface 14: 2.1e5, normalized score = 0.9425

The dot product of 2 vectors measures how similar 2 vectors are in the same coordinate, thus can be used to recognize (cluster) the faces of the test images.