Computational Data Analysis

ISYE 6740

Final Exam– Spring 2020

Total Score: 100

If you think a question is unclear or multiple answers are reasonable, please write a brief explanation of your answer. to be safe. Also, show your work if you want wrong answers to have a chance at some credit: it lets us see how much you understood.

(Please sign the honor code below.) 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.

You are only required to finish ONE of the Programming Questions, i.e., please pick either Question 8 or Question 9 to work on. If you decide to try both, you will receive BONUS points (please indicate which question you would like to use as BONUS).

Please submit a zip folder, with your final answers (in pdf file format) and code for your Program Question(s).

Name: Hadi Sharifi

GT ID: 903472180

GT Account: hsharifi7

Question 1 [10 points]	
Question 2 [10 points]	
Question 3 [10 points]	
Question 4 [10 points]	
Question 5 [15 points]	
Question 6 [15 points]	
Question 7 [10 points]	
Question 8 [20 points]	
Question 9 [20 points]	

1 K-means (10 points)

Given $m = 5$ data points configuration in Figure 1. Assume $K = 2$ and use Manhattan distance (a.k.a. the ℓ_1 distance: given two 2-dimensional points (x_1, y_1) and (x_2, y_2), their distance is $|x_1 - x_2| + |y_1 - y_2|$). Assuming the initialization of centroid as shown, after one iteration of k-means algorithm, answer the following questions.

(a) (3 points) Show the cluster assignment;

(b) (4 points) Show the location of the new center;

(c) (3 points) Will it terminate in one step?

Notation: d_i is the distance for Point i

a) First iteration

A: $d_1 = 8$ $d_2 = 4$ $d_3 = 8$
$d_4 = 3$ $d_5 = 2$

B: $d_1 = 1$ $d_2 = 3$ $d_3 = 1$
$d_4 = 4$ $d_5 = 7$

A(4, 5) ← Assignments
B(1, 2, 3) ←

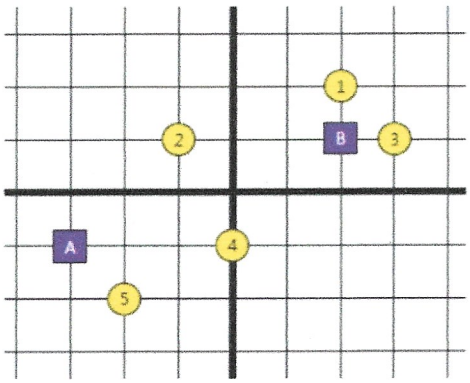


Figure 1: Question 1.

b) A New location: $A_y = \frac{(-1)+(-2)}{2} = -1.5 = -1$ $A_x = \frac{0+(-2)}{2} = -1$ A(-1, -1)

B New location: $B_y = \frac{1+2+1}{3} = 1.3 = 1$ $B_x = \frac{3+2+(-1)}{3} = 1.3 = 1$ B(1, 1)

location ↓

c) the algorithm won't stop with first iteration Because the position of centroid changed.
with the new location A(2, 4, 5), B(1, 3) ← Assignments

In this particular case, the algorithm stops in third iteration where the location for A(-1, -1), B(2, 1) the Algorithm stops.

2 Clustering [10 pts]

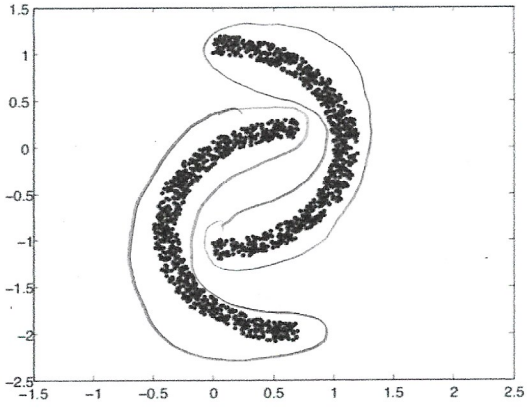
1. (3 points) Explain what is the difference between K-means and spectral clustering?

the main difference is that k-means cares about distance of points while spectral cares about connectivity.
From calculation respective, spectral clustering does eigen decomposion while kmeans does not.

2. (3 points) What is in common, and what is the main difference between spectral clustering and PCA?

Both Algorithms uses eigen de composition hence both use some sort of dimensionality reduction technique.
the difference is that PCA is done on covariance matrix but spectral is done on similarity matrix (laplacian).
Also in PCA we pick (From) largest eigen value but standard spectral we pick From smallest

3. (4 points) For the following data (two moons), give one method that will successfully separate the two moons? Explain your rationale.



In this clustering problem, spectral does the best.
Because the data is not linearly seprable and spectral Algorithm will clusterall points close to each othe in one data.
this is a perfect problem for spectral Algo as all points that tightly close to each other (spatially connected) can belong to one cluster.

...(from (a)) eigen vector, project the data on the eigen vector.
the eigen vector is the direction of the new axis.
I used "scipy" package to calculate the principle directions (eigen vector)
eigen values (122.6, 0.421, ~0) V_1 (0.317, -.836, -.447) ← Pick this direction b/c its eigen value is the largest

3 Principal Component Analysis (10 pts)

Suppose we have 4 points in 3-dimensional Euclidean space, namely (1, 0, 0.5), (6, 14, 3), (11, 28, 5.5), and (7, 21, 3.5).

(a) (3 points) Find the first principal direction. First, write down the data matrix and make sure you fill in numbers specific for this problem. Then explain how to find the first principal direction and what is the optimization problem you need to solve? Find the first principal direction either by calculation in hand, or using program or software - report the first principal direction you found.

three dimension

$$A = \begin{bmatrix} 1 & 0 & 0.5 \\ 6 & 14 & 3 \\ 11 & 28 & 5.5 \\ 7 & 21 & 3.5 \end{bmatrix}$$

every column is one dimension of the data.
The optimization problem is to find a direction for our data that data projects max variance.
To find the priciple direction, we use eigen values & eigen vector of covariance of the data. Pick the largest eigen value and related ... ↑ top

(b) (2 points) What are the first principle components, for each of the data points?

The first principle component is related to eigen value 122.625
after transforming data: A_1 (1.51, 0.156, -1.204, -0.470)
A_1 is the first principle component.

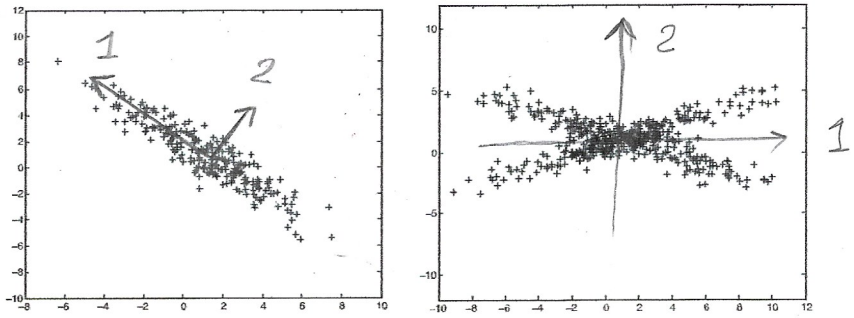
All calculations are done in a jupyter note book Q3 attached.

(c) (2 points) When we reduce the dimensionality from 3 to 1 based on the principal direction you found in (a), what is the remaining variance in the data?

the total variance is the sum of diagonal data on covariance matrix of data which is equal to total eigen values.
Hence the remaining variance is: 0.421 + 0 = 0.421

(d) (3 points) You are given the following 2-D datasets, approximately draw the first and second principal directional on each plot.

Next Page.



(a) (b)

a) on the directions I defind, I see the max variance. the first one captures max variance and second is the next.

b) the first direction captures maximum variance (1 PC) and the second captures the next (5 PC)

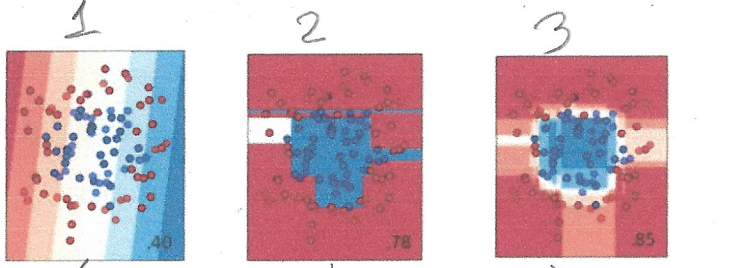
4 Classification [10 points]

1. (5 points) List all methods below which can be used for classification:

   (a) AdaBoost (b) Decision Trees (c) EM and Gaussian Mixture (d) Histogram (e) K-nearest neighbors (f) K-means (g) Kernel density estimation (h) Linear Regression (i) Logistic Regression (j) Naive Bayes.

   AdaBoost, Decision Tree, EM&GM, Histogram
   K-nearest neighbors, Kmean, KDE, logistic regression
   Naive Bayes

2. (5 points) Which of the decision boundaries below correspond to (a) Random Forest, (b) Decision Tree, (c) SVM. Explain your reasons to fully justify your answers.



SVM (linear) — Decision Tree — Random Forest

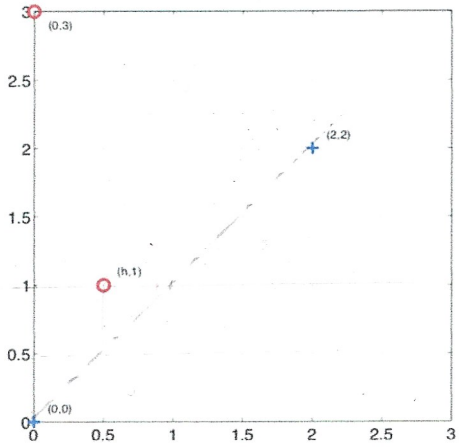
1: the classification is linear and the only linear among the list is SVM linear.

2: this is decision Tree because of how it is partitioned. and the precision is lower than 3. Usually random forest has higher precision.

3: this is random forest, for its Picularity and details. and when compare to 2 which is also tree based, it higher precision with more details.

5 SVM [15 points]

Suppose we only have four training examples in two dimensions as shown in Fig. The positive samples at $x_1 = (0,0)$, $x_2 = (2,2)$ and negative samples at $x_3 = (h,1)$ and $x_4 = (0,3)$.



1. (5 points) For what range of parameter value $h > 0$ be so that the training points are still linearly separable?

the training points are seprable for any "h" that $h < 1$
h should be strictly smaller than 1

2. (5 points) Does the orientation of the maximum margin decision boundary change as a function of h when the points are separable?

the orientation of the margin wont change as far as data is linearly separable. And if h becomes so far to left, it then the second point (0,3) becomes the support point but the orientation wont change.

3. (5 points) Explain why only the data points on the "margin" will contribute to the decision boundary?

when we were proving SVM, we used KKT condition.

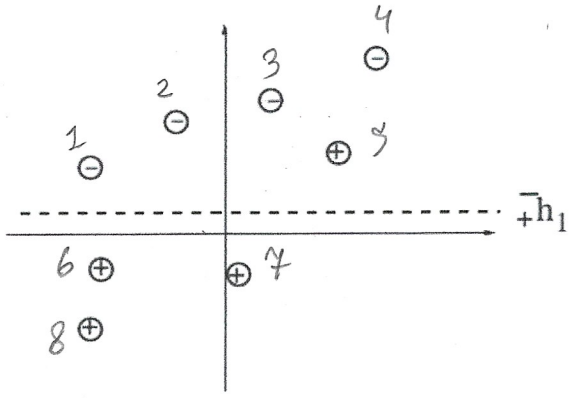
Lagrangian multiplier

$\leftarrow a_i (1 - y^i (w^T x^i + b)) = 0 \xrightarrow{KKT}$ if $(1 - y^i (w^T x^i + b)) < 0 \Rightarrow a_i = 0$
if $(1 - y^i (w^T x^i + b)) = 0 \Rightarrow a_i > 0$ (*)

this (**) shows that only data on the margin will have a non-zero "a" and the rest will have "a = 0". Those points on the margin are the support vectors that contributy to decision boun

6 Boosting algorithms [15 points]

In this problem, we test your understanding of AdaBoost algorithm. The figure shows a dataset of 8 points, equally divided among the two classes (positive and negative). The figure also shows a particular choice of decision stump h_1 picked by Adaboost in the first iteration.



α is calculated in part (b)

in every iteration those data that are missclassified are given higher weight

(a) (6 points) Explain the weights $D_2(i)$ for each sample after the first iteration. You can explain by drawing figures similar to what we have in class.

the weight for each point is 1/8. After first round of the algorithm, the only point that was mis-predicted is X5.

$D_2(i) = \frac{D_1(i)}{Z_t} \exp(-\alpha_1 y_i h_1(x_i)) \Rightarrow D_2(i) = (0.071, 0.071, 0.071, 0.071, 0.5, 0.071, 0.071, 0.071)$

normalization factor

(b) (6 points) Calculate the weight α_1 assigned to h_1 by Adaboost? (Note that initial weights of all the data points are equal, $D_1(i) = 1/8, \forall i$).

X5 is mis classified hence it poses weight loss $\Rightarrow \varepsilon_1 = \sum$ weight loss $= .125$

By knowing $\varepsilon_1 \Rightarrow \alpha_1 = \frac{1}{2} * \log(\frac{1-\varepsilon}{\varepsilon}) = 0.9729$

All calculation are in the excel sheet attached

(c) **[True/False]** (3 points) The votes α_i assigned to weak classifiers in boosting generally changes monotonically as the algorithm proceeds.

That is false. for example, in our HW6 it changes from positive to negative and back to positive.

therefor it is not changing monotonically.

important point $\Rightarrow \alpha$ as a function, monotonic. but how we calculate α in each round depends on missclassifications. 0 missclassification $\Rightarrow \varepsilon = 0 \Rightarrow \alpha = \infty$

1 continue: In lasso we replace L_2 for regularization factor with L_1.

$$\theta^L = \text{argmin } L(\theta) = \frac{1}{m}\sum_{i=1}^{m}(y^i - \theta^T x^i)^2 + \lambda\|\theta\|^1$$

L2 norm is Euclidean norm.

L1 norm is manhattan norm.

7 Variable section [10 points]

Suppose we have data $\{x_i, y_i\}$, $i = 1, \ldots, m$, where $x_i \in \mathbb{R}^p$ corresponds to p features.

1. (3 points) Write down the optimization problem we solve with Ridge Regression and Lasso. Make sure you explain your notations: which are the decision variables, and which are data.

   Giving m data points, for ridg regression, we find θ that minimizes the regularized mean squre: $\theta^r = \text{argmin}_\theta L(\theta) = \frac{1}{m}\sum_{i=1}^{m}(y^i - \theta^T x^i)^2 + \lambda\|\theta\|^2$

   λ is the tuning variable

   θ coefficients that minimizes the mean squar

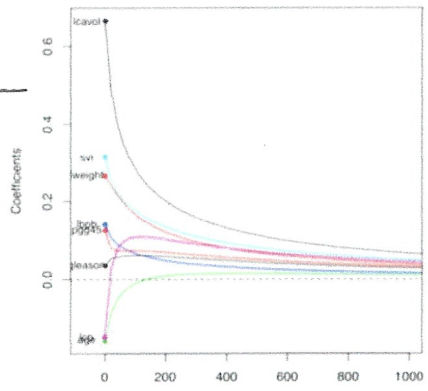
   x input data

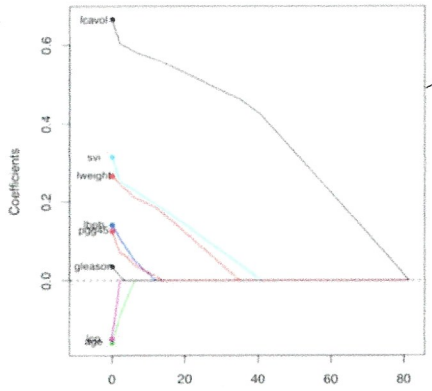
   y regression result

   the rest on top↑

2. (3 points) Which of the solution paths below corresponds to Ridge regression and which corresponds to Lasso?

Ridge ←





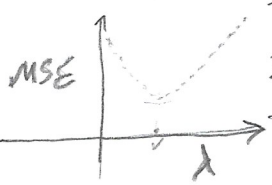
→ Lasso

Ridge uses L2 for the penalty factor. L2 norm pulls directly towards the origin. Lasso uses L1 which pulls directly towards axes.

3. (2 points) Explain what's the difference between Lasso and Ridge regression. We need Lasso for what setting?

the difference is on the penalty expression. Ridge uses L2 and Lasso uses L1. which causes some of the cofficient to be zero. Ridge only lowers the importants of some coffeicient. If we want simpler model with fewer feature we use lasso.

4. (2 points) Explain how to tune the regularization parameters for Lasso and Ridge regression (hint: CV).



The typical tuning is throug cross validation. In this technique, we define a range of λ and perform CV on the data. we plot the mean squre error with respect to λ. usually the MSE goes down and start to go up. the λ with minimum MSE will be the selected λ.

Answers are in the next page.
The programming is in attached Jupyter notebook (Q8).

You are only required to finish ONE of the Programming Questions, i.e., please pick either Question 8 or Question 9 to work on. If you decide to try both, you will receive BONUS points (please indicate which question you would like to use as BONUS).

Please submit a zip folder, with your final answers (in pdf file format) and code for your Program Question(s).

8 PCA for face recognition (20 points)

This question is a simplified illustration of using PCA for face recognition using a subset of data from the famous Yale Face dataset.

Remark: you have to perform downsampling of the image by a factor of 4 to turn them into a lower resolution image before we do anything.

1. (10 points) First, given a set of images for each person, we generate the so-called eigenface using these images. The procedure to obtain eigenface is explained as follows. Given n images of the *same person* denoted by $x_1, \ldots, x_n$. Each image originally is a matrix. We vectorize each image to form the vector $x_i \in \mathbb{R}^p$. Now form a matrix

   $$X = [x_1, \ldots, x_n] \in \mathbb{R}^{p \times n}.$$

   Extract the largest k eigenvector of the data matrix $X^\top X$, denoted as $u_1, u_2, \ldots, u_k$. The eigenfaces correspond to the projections $\text{eigenface}_i = Xu_i$, $i = 1, \ldots, k$.

   Perform analysis on the Yale face dataset for subject 14 and subject 01, respectively, using all the images EXCEPT for the two images named subject01-test.gif and subject14-test.gif. **Plot the top 6 eigenfaces for each subject.** When visualizing, you have to reshape the eigenvectors into images with the same dimension as the original images.

   What is the interpretation of the top 6 eigenfaces?

2. (10 points) Now we will perform a face recognition task.

   For doing face recognition through PCA we proceed as follows. Given the test image subject01-test.gif and subject14-test.gif, we vectorize each image. Take the top eigenfaces of Subject 1 and Subject 14, respectively, project the 2 vectorized test images using the vectorized eigenfaces to obtain scores, respectively.
   (Hint: use $(\text{eigenface}_1)^\top(\text{test image})$)

   Report four scores: (1) projecting test image of Subject 1 using eigenface of Subject 1; (2) projecting test image of Subject 1 using eigenface of Subject 14; (3) projecting test image of Subject 14 using eigenface of Subject 1; (4) projecting test image of Subject 14 using eigenface of Subject 14.

   Explain whether or not (and how) can you recognize the faces of the test images using these scores.

1 The eigenfaces are considered a set of features which characterize the global variation among face images. In other words the top eigenfaces are the orthogonal basis set from which the faces are constructed.

2 after nomalizing the test image and eigen face vetor (the first) the results as follows: (all in abstrat value)

a) Projecting test image subject 1 using eigenface subject 1: 0.4007

b) " " " " 1 " " " 14: 0.3568

c) " " " " 14 " " " 1: 0.3646

d) " " " " 14 " " " 14: 0.4040

The results shows that $(\text{eigenface}_1)^T (\text{test image})$ is decent indicator for classification. If we project the test image (belong the correct cluster faces) to the eigen face of that cluster, the results are larger than those doesn't belong.

We see results in our example too. Project 1→1 > Project 1→14

or Project 14→14 > Project 14→1.

And this makes sence, because if a face belong to a cluster face, the projection of this face (image) on the eigen face will generate larger value too.

You are only required to finish ONE of the Programming Questions, i.e., please pick either Question 8 or Question 9 to work on. If you decide to try both, you will receive BONUS points (please indicate which question you would like to use as BONUS).

Please submit a zip folder, with your final answers (in pdf file format) and code for your Program Question(s).

9 Programming: Bayes and KNN classifier [20 points]

In this programming assignment, you are going to apply the Bayes Classifier to handwritten digits classification problem. Here, we use the binary 0/1 loss for binary classification, i.e., you will calculate the miss-classification rate as a performance metric. To ease your implementation, we selected two categories from USPS dataset in usps-2cls.mat (or usps-2cls.dat, usps-2cls.csv).

1. (10 points) Your first task is implementing the classifier by assuming the covariance matrices for two classes are a diagonal matrix Σ_1, Σ_2.

   Using slides from "Classification I", assuming $P(y = 1) = P(y = -1)$ (i.e., the prior distribution for two classes are the same), using Bayes decision rule to write down the decision boundary. (Hint, it should be a quadratic decision boundary.)

   Now we will estimate the mean vector and the sample covariance matrices for two classes using the training data (hint: you can use sample mean and sample covariance vector). Report the misclassification rate (error rate) over the training set and over the testing set averaged over the 100 random train/test splits by using different value of splitting ratio p. Explain and compare the performance of each classifier.

   After implementing these methods, you should evaluate your algorithm on the given set. Repeat 100 times: split the dataset into two parts randomly, use p portion for training and the other $1-p$ portion for testing. Let p change from 0.1, 0.2, 0.5, 0.8, 0.9.

   Please implement the algorithm **from scratch** yourself. Make sure to provide code, results (required above) together with necessary explanations to your results.

2. (10 points) Now repeat the classification again using K-nearest neighbors, for $K = 5, 10, 15, 30$. Repeat 100 times: split the dataset into two parts randomly, use p portion for training and the other $1-p$ portion for testing. Let p change from 0.1, 0.2, 0.5, 0.8, 0.9. Report the training error and testing error for each case.

   For this part, you may use any package that you like. Make sure to provide code, results (required above) together with necessary explanations to your results.