Computational Data Analysis

ISYE 6740

Final Exam– Due Dec. 13, 2019

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 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. I understand that violation of these rules, including using an authorized aid or copying from another person, may result in my receiving a 0 on this exam .

Name:

GT ID:

GT Account:

Question 1 [15 points]	
Question 2 [15 points]	
Question 3 [15 points]	
Question 4 [20 points]	
Question 5 [10 points]	
Question 6 [25 points]	

1 Clustering [15 pts]

1. (5 points) Explain what is the difference between K-means and spectral clustering?

K-means:
use L_2 norm as metric for determine the distance among data points, can properly find the data clustering when the data is linearly separable. Needs to predetermine the total number of clusters

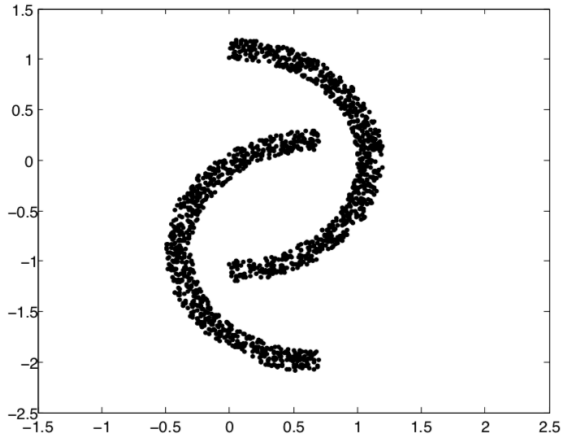
Spectral-clustering:
use *Graph-Laplacian* to describe the distance among data, can discover the data clustering on non-linear manifolds. Does not need to predetermine total number of clusters

2. (5 points) What is in common, and what is the main difference between spectral clustering and PCA?

common:
Both spectral-clustering and PCA rely on *principle component decomposition*; they do not require data labeling, i.e., both are unsupervised learning algorithms;

difference:
PCA finds the linear representation data in the data space, while spe-clustering performs eigen-decomposition on the graph Laplacian matrix; PCA can only properly interperte the data when it is linearly separable, spe-clustering can work for data locating on non-linear manifolds.

3. (5 points) For the following data (two moons), give one method that will successfully separate the two moons? Explain your rationale.



spetral-clustering can separate the two class while PCA cannot. Because the data is not linearly separable. It has a clear clustering pattern on a non-linear manifold.

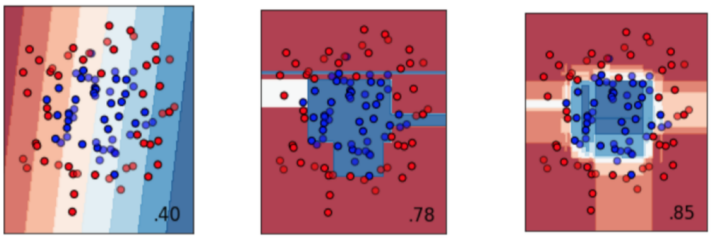
2 Classification [15 points]

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.

   a) AdaBoost, (b) Decision Trees, (e) K-nearest neighbors, (i) Logistic Regression, (j) 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.



   img left: SVM. The decision boundary is a linear hyperplane.
   img mid: Decision Tree. Non-linear decision boundary. The decision boundary is sharp
   img right: Random Forest. Non-linear decision boundary. The decision boundary is similar to Decision Tree but fuzzier, due to the averaging mechanism.

3. (5 points) Is the following statement true / false: "In the AdaBoost algorithm, the weights on all the misclassified points will go up by the same multiplicative factor." Explain your reason.

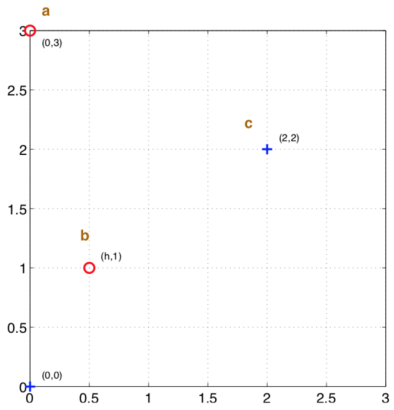
   The statement is false. We can refer to the lecture notes page 7,

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

   instead of going up by the same multiplicative factor, the misclassified points later $D_{t+1}(i)$ has increasingly greater weights than previous ones $D_t(i)$, given that the misclassification error decreases over t.

3 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 h, s.t.,$h > 0$ be so that the training points are still linearly separable?

   $$\{0 < h < 1\} \cup \{h > 4\}$$

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

   Yes, when $h > 4$, the orientation of maximum margin decision boundary can be written as a function of h. Let $\vec{a}, \vec{b}, \vec{c}$ be the vectors corresponding to the three points in the image, the orientation becomes

   $$\begin{aligned} \vec{w}(h) &= \vec{c} - \frac{1}{2}(\vec{a} + \vec{b}) \\ &= \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \frac{1}{2}(\begin{bmatrix} 0 \\ 3 \end{bmatrix} + \begin{bmatrix} h \\ 1 \end{bmatrix}) \end{aligned}$$

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

   This can be seen from the SVM optimization formulation,

   $$\begin{aligned} &\min_{\boldsymbol{w}, \boldsymbol{b}} \|\boldsymbol{w}\|^2 \\ \text{s.t.} \quad &y^i(\boldsymbol{w}^T \boldsymbol{x}^i + \boldsymbol{b}) \geq 1 \end{aligned}$$

   for given $\mathbf{w}$, $\mathbf{b}$, any data points $(y^j, \boldsymbol{x}^j)$ beyond the margin will satisfy the constrain inequality. They hence can be discarded and has no contribution to the decision boundary.

4 Variable section [20 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. (5 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.

   - 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

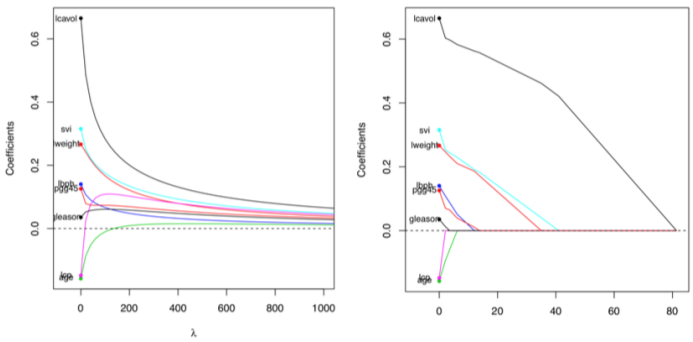
   $$\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} (y_i - \boldsymbol{\theta}^T \boldsymbol{x})^2$$
   $$\text{s.t.} \quad \|\boldsymbol{\theta}\|_1 \leq c = \frac{1}{\lambda}$$

   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: $y_i, \boldsymbol{x}_i$

2. (5 points) Which of the solution paths below corresponds to Ridge regression and which corresponds to Lasso?



left: ridge regression
right: Lasso

3. (5 points) Explain what's the difference between Lasso and Ridge regression. We need Lasso for what setting?

Ridge regression: use L2 norm of parameter in the regularization term, push the parameter towards the origin.
Lasso: use L1 norm of parameter in the regularization term, push the parameter towards part of the coordinates.
We need Lasso for variable selection.

4. (5 points) Explain how to tune the regularization parameters for Lasso and Ridge regression.

We can use cross-validation to select proper regularization parameter λ

5 Neural networks [10 points].

1. (5 points) Consider a neural networks for a binary classification using sigmoid function for each unit. If the network has no hidden layer, explain why the model is equivalent to logistic regression.

   Let $\text{sigm}(\cdot)$ be the sigmoid function. Without hidden layer, the neural network model can be written as,

   $$y_j = \text{sigm}(\sum_i W_{i,j} x_i)$$

   Due to the graph structure that there's no intra-connection within the hidden layer, the probability of hidden units, conditional on input layer, are independent among each other. Let N be the size of hidden layer

   $$P(y_j, y_k|\boldsymbol{x}) = P(y_j|\boldsymbol{x}) \cdot P(y_k|\boldsymbol{x}) \quad \forall j, k \in \{1, \cdots, N\}, j \neq k.$$

   $$P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{j=1}^{N} P(y_j|\boldsymbol{x}) = \prod_{j=1}^{N} \text{sigm}(\sum_i W_{i,j} x_i))$$

   Above results indicate that the neural networks without hidden layer can be seen as the union of multiple independent mappings, each mapping has the same form of logistic regression. Specifically, if there is only one hidden unit, such neural network is exactly a logisitic regression model.

2. (5 points) Consider a simple two-layer network in the lecture slides. Given m training data (x^i, y^i), $i = 1, \ldots, m$, the cost function used to training the neural networks

   $$\ell(w, \alpha, \beta) = \sum_{i=1}^{m} (y^i - \sigma(w^T z^i))^2$$

   where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function, z^i is a two-dimensional vector such that $z_1^i = \sigma(\alpha^T x^i)$, and $z_2^i = \sigma(\beta^T x^i)$. Show the that the gradient is given by

   $$\frac{\partial \ell(w, \alpha, \beta)}{\partial w} = -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1 - \sigma(u^i))z^i,$$

   where $u^i = w^T z^i$. Also find the gradient of $\ell(w, \alpha, \beta)$ with respect to α and β and write down their expression.

- part1, by chain rule

$$\begin{aligned}
\frac{\partial \ell}{\partial w} &= \sum_{i=1}^{m} \frac{\partial}{\partial u^i}(y^i - \sigma(u^i))^2 \frac{\partial u^i}{\partial w} \\
&= \sum_{i=1}^{m} 2(y^i - \sigma(u^i)(-1)\frac{\partial}{\partial u^i}(\frac{1}{1+e^{-u^i}}) \cdot \frac{\partial u^i}{\partial w} \\
&= \sum_{i=1}^{m} 2(y^i - \sigma(u^i)(-1)\frac{-1}{(1+e^{-u^i})^2} \cdot e^{-u^i}(-1) \cdot \frac{\partial u^i}{\partial w} \\
&= \sum_{i=1}^{m} 2(y^i - \sigma(u^i) \cdot \frac{1}{1+e^{-u^i}}(\frac{e^{-u^i}}{1+e^{-u}})(-1) \cdot z^i \\
&= \sum_{i=1}^{m} 2(y^i - \sigma(u^i) \cdot \frac{1}{1+e^{-u^i}}(1 - \frac{1}{1+e^{-u}})(-1) \cdot z^i \\
&= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))z^i
\end{aligned}$$

- part2

$$\frac{\partial \ell}{\partial \alpha} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial u^i}\frac{\partial u^i}{\partial z_1^i}\frac{\partial z_1^i}{\partial \alpha}$$

from part 1

$$\begin{aligned}
\frac{\partial \ell}{\partial u^i} &= \sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i)) \\
\frac{\partial u^i}{\partial z_1^i} &= z_1^i \\
\frac{\partial z_1^i}{\partial \alpha} &= \frac{\partial}{\partial \alpha}\frac{1}{1+e^{-\alpha^T \boldsymbol{x}^i}} \\
&= \frac{(-1)e^{-\alpha^T \boldsymbol{x}^i}}{(1+e^{-\alpha^T \boldsymbol{x}^i})^2}(-\boldsymbol{x}^i) \\
&= \frac{-e^{-\alpha^T \boldsymbol{x}^i}}{1+e^{-\alpha^T \boldsymbol{x}^i}}\frac{1}{1+e^{-\alpha^T \boldsymbol{x}^i}}(-\boldsymbol{x}^i) \\
&= (1-z_1^i)z_1^i x^i \\
\frac{\partial \ell}{\partial \alpha} &= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i)) \cdot z_1^i \cdot (1-z_1^i)z_1^i x^i
\end{aligned}$$

similarly,

$$\frac{\partial \ell}{\partial \beta} = -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i)) \cdot z_2^i \cdot (1-z_1^i)z_2^i x^i$$

6 Programming: Bayes and KNN classifier [25 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. (15 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.

consider the log likelihood ratio

$$f = \log \frac{P(y=1|x)}{P(y=-1|x)} = \log \frac{(2\pi)^{d/2}|\Sigma_1|^{-1/2} + \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)}{(2\pi)^{d/2}|\Sigma_2|^{-1/2} + \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)} \geq \log 1 = 0$$

the decision boundary is given by:

$$x^T(\Sigma_1^{-1} - \Sigma_2^{-1})x - 2x^T(\Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2) + (\mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2) - \log|\Sigma_1| + \log|\Sigma_2| \geq 0$$

The results from Matlab implementation are summarized as following

KNN			
K	Training Err	Testing Err	p
5	0.0392273	0.0767424	0.1
	0.0267727	0.0524886	0.2
	0.0160727	0.0292182	0.5
	0.011983	0.0232273	0.8
	0.0110253	0.0210909	0.9
10	0.0727727	0.102848	0.1
	0.0498182	0.0672273	0.2
	0.0297545	0.0399	0.5
	0.0213125	0.0271818	0.8
	0.0218939	0.0275	0.9
15	0.0747273	0.0951818	0.1
	0.0476818	0.0640511	0.2
	0.0307636	0.0383	0.5
	0.0235227	0.0307045	0.8
	0.0215303	0.0275	0.9
30	0.122955	0.138889	0.1
	0.0794091	0.093125	0.2
	0.0446182	0.0521636	0.5
	0.0351648	0.0386364	0.8
	0.0325909	0.0377727	0.9

Bayes		
Training Err	Testing Err	p
0	0.0288182	0.1
0.00020455	0.0259773	0.2
0.00288182	0.0226091	0.5
0.00675568	0.0223409	0.8
0.00750505	0.0224091	0.9

- KNN,
  K: Smaller neighbor size K will provide higher resolution to the decision region, leading to both smaller training error and training error. Larger K will lead to smoother decision boundary, while both training error and testing error becomes larger.
  p: in terms of the proportion of training data / testing data, for a fixed K, the more data we have for training, the lower training error and testing error it would be. The classification performance of KNN is sensitive to p
- Bayes The testing errors are quite consistent for different p.

Rubric

1. Clustering

1. (5 points) Explain what is the difference between K-means and spectral clustering?
    - any meaningful explanation attempt, 3pts
    - address linearly separable / nonlinear manifold difference, 2pts
2. (5 points) What is in common, and what is the main difference between spectral clustering and PCA?
    - any meaningful explanation attempt, 3pts
    - address linearly separable / nonlinear manifold difference, 2pts
3. (5 points) For the following data (two moons), give one method that will successfully separate the two moons? Explain your rationale.
    - correct choice 3pts
    - reasonable explanation 2pts

2. Classification

1. (5 points) List all methods below which can be used for classification:
    - one correct choice, 1pts
    - one wrong choice, -0.5pts
    - all correct choice and no wrong choice, 5pts
    - pts for this question no less than 0
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.
    - for each picture:
    - correct choice, 1pt
    - reasonable explanation 0.5pt
    - all answers correct, 5pt
3. (5 points) Is the following statement true / false: "In the AdaBoost algorithm, the weights on all the misclassified points will go up by the same multiplicative factor." Explain your reason.
    - correct choice 3pts
    - reasonable explanation 2pts

3. SVM

1. (5 points) For what h s.t.$h > 0$ to be so that the training points are still linearly separable?
    - any reasonable attempt, 1pts
    - one of correct region, 3pts
    - accurate answer, 5pts
2. (5 points) Does the orientation of the maximum margin decision boundary change as a function of h when the points are separable?

- correct answer to yes/no, 3pts
- reasonable explanation 2pts

3. (5 points) Explain why only the data points on the "margin" will contribute to the decision boundary?
    - any reasonable attempt, demostrated correct understanding to related concepts 3pts
    - accurate explanation 2pts

4. Variable Selection

1. (5 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.
    - each of subquestion, 2.5pts
    - correct formulation, 1.5pts
    - correct distiction of decision variables and data, 1pt
2. (5 points) Which of the solution paths below corresponds to Ridge regression and which corresponds to Lasso?
    - correct answer to each image, 2.5pts
3. (5 points) Explain what's the difference between Lasso and Ridge regression. We need Lasso for what setting?
    - reasonable explanation, 2.5pts
    - correct answer to the use of Lasso, 2.5pts
4. (5 points) Explain how to tune the regularization parameters for Lasso and Ridge regression.
    - correct answer, 5pts

5. Neural Network

1. (5 points) Consider a neural networks for a binary classification using sigmoid func- tion for each unit. If the network has no hidden layer, explain why the model is equivalent to logistic regression.
    - any reasonable attempt, 2pts
    - accurate answer, 5pts
2. (5 points) Consider a simple two-layer network in the lecture slides. ...
    - any reasonable attempt to proof 1pt
    - accurate proof, 2.5pts
    - any reasonable attempt to the expression of gradient, 1pt
    - accurate answer 2.5pts

6. Programming: Bayes/KNN

codes that produce results within 30mins are considered as *runnable*, longer than that are considered as *non-runnable*. If no code provided, no credit for the question.

1. (15 points) Bayes classifer
    - any meaningful attempt, 3pts
    - correct decision boundary expression, 3pts
    - runnable, built-from-scratch code that cover all the required experiment, 3pts
    - reasonable output from code, 3pts
    - reasonable explanation 3pts
2. KNN
    - runnable, built-from-scratch code that cover all the required experiment, 5pts
    - reasonable output from code, 3pts
    - reasonable explanation 2pts