ECE M146 Spring 2020 Final Assessment
Introduction to Machine Learning
Instructor: Lara Dolecek Maximum score: 100 points

You have until **June 12th 11:00 AM (Pacific Time)** to submit your work **directly on Gradescope**.
Please read and carefully follow all the instructions.

Instructions

- You may type your exam or scan your handwritten version. Please show your work and make sure all the work is discernible.
- Make sure to include your **full name** and **UID** in your submitted file.
- For questions related to the exam, you may deal into the following Zoom Q&A sessions:
  - June 11th, 1:00pm - 1:30pm.
  - June 11th, 3:00pm - 3:30pm.
  - June 11th, 6:00pm - 6:30pm.
  - June 11th, 9:00pm - 9:30pm.
  - June 12th, 7:30am - 8:00am.
  - June 12th, 9:30am - 10:00am.

  Links to these Zoom Meetings are available under Week 10 on CCLE. **Only clarification questions** will be answered. Please do not ask for hints. We will also have a forum under Week 10 that reiterates all answered questions. Make sure to check the forum before dial in.
- **Important:** Throughout this exam, you will find a parameter α in some of the questions. All α refers to the same parameter. This parameter α is dependent on your UID, specifically, $\alpha = (\text{Last } \textbf{Two} \text{ digits of UID } \mod 8) + 1$. For example, a person with UID: 123456789 will use $\alpha = (89 \mod 8) + 1 = 2$ throughout this test. Please clearly indicate what is your α on the first page of your answers. You **will lose points** if the correct α is not used.
- **Academic Integrity**
  During this exam, you are **allowed** to use all course material posted online, including lectures, discussion, and homeworks, and your own textbooks. You are **disallowed** to contact with a fellow student or with anyone outside the class who can offer a solution e.g., web forum.
  Please write the following statement on the first page of your answer sheet. You will **lose 20 points** if we can not find this statement. The policy on academic dishonesty can be found at the same place with this exam.

I __*YourName*__ with UID ____ have read and understood the policy on academic dishonesty available on the course website.

1. (30 points) **Expectation Maximization**
   This is a programming question. **Please attach a printout of your code with your answer. You will lose points if you don't attach you code.**

   In this question, you will implement the Expectation-Maximization algorithm from first principles to learn a mixture of **two** Gaussian from the `old_faithful` dataset.

   The `old_faithful` dataset contains unlabeled 2-dimensional data which are provided in `Old_faithful_normalized.csv`. The first column contains the eruption times of the famous Old Faithful Geyser and the second column contains the waiting times between eruptions. Note that you are **NOT** going to use the data directly. You will need to **discard** the $(30(\alpha-1)+1)$-th to (30α)-th rows in `Old_faithful_normalized.csv` and use the remaining data for this question.

   The Expectation-Maximization algorithm that learns a mixture of K Gaussian components from M-dimensional data set $\{x_1, \cdots, x_N\}$ is as follows:

   - Initialize the mean $\mu_k \in \mathbf{R}^M$, covariances $\Sigma_k \in \mathbf{R}^{M\times M}$ and mixing coefficient π_k for each Gaussian component.
   - **Expectation Step.** For $n = 1, \cdots, N$ and $k = 1, \cdots, K$, evaluate the responsibilities using the current parameters for the Gaussian components:

     $$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{N} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)},$$

     where $\mathcal{N}(z|\mu, \Sigma)$ is defined as the evaluation of the multivariate Gaussian pdf with parameters μ and Σ on the point z.
   - **Maximization step.** Re-estimate the parameters for each Gaussian component using the current responsibilities:

     $$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n;$$
     $$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T;$$
     $$\pi_k^{new} = \frac{N_k}{N};$$

     where

     $$N_k = \sum_{n=1}^{N} \gamma(z_{nk}).$$

   - Repeat the **E-M** steps until parameters converge or certain number of iterations is reached. We define the completion of an expectation step and a maximization step as one iteration.

   For this question, you will use the following initializations:

   $$\mu_1 = [-1.7, 1]^T, \mu_2 = [1.7, -1]^T, \Sigma_1 = \Sigma_2 = I_2(2\times 2 \text{ indetity matrix}), \pi_1 = \pi_2 = 0.5.$$

(a) (10 pts) We learned that the log likelihood function for the Gaussian mixture model is of this form:

$$J = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}.$$

Suppose we want to maximize J with respect to Σ_k. Here we must take into account the constraint $\sum_{k=1}^{K} \pi_k = 1$. Use a Lagrange multiplier to enforce this constraint. Show that given known $\gamma(z_{nk})$ and μ_k, the optimal Σ_k is of this form:

$$\Sigma_k^{optimal} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T,$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}).$$

Hint: You may use the following properties without proof. A is a square matrix, and $Tr(A)$ refers to the trace of the matrix, which is the sum of diagonal entries. a refers to a scalar. $a = Tr(a)$ for scalar a; $Tr(A) + Tr(B) = Tr(A + B)$; $Trace(AB) = Trace(BA)$; $\frac{\partial |A|}{\partial A} = |A|A^{-T}$; $\frac{\partial Tr(A^{-1}B)}{\partial A} = -(A^{-1}BA^{-1})^T$.

(b) (5 pts) Plot the data. Initialize the two Gaussian components based on the initialization provided. On the same plot, also plot the contour of each Gaussian component, i.e., $\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$. You may use `contour` in Matlab and `matplotlib.pyplot.contour` in Python. For clarity, you only need the contour line where $\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) = 0.05$, i.e., set "LevelList" ("Level") to be 0.05. Use color blue for the Gaussian component with parameters μ_1 and Σ_1 and use color red for the Gaussian component with parameters μ_2 and Σ_2. Does it make sense to learn a mixture of **two** Gaussians? Is this a good initialization? From what you have learned about unsupervised learning, propose a better way of initialization (you don't need to implement this).

(c) (5 pts) Implement the E-M algorithm and run it for 1 iteration. Visualize the responsibilities for each data point with a scatter plot after the expectation step. To do this, plot points with $\gamma_{n1} = 1$ as blue and point with $\gamma_{n2} = 1$ as red and other points be a combination of red and blue, i.e., color for point $n = \gamma_{n1} \times [0, 0, 255] + \gamma_{n2} \times [255, 0, 0]$. After the maximization step, also plot the contour of each Gaussian component on the same plot at $\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) = 0.05$.

(d) (10 pts) Run the E-M algorithm for 10 iterations and repeat the plot in part (c) for iteration 2, 5 and 10. Comment on the convergence of the E-M algorithm based on the plots and report you learned parameters μ_k, Σ_k, and π_k after the last iteration. Do the learned parameters match your intuition?

2. (30 points) **Principal Component Analysis**
This is a programming question. **Please attach a printout of your code with your answer. You will loss points if you don't attach you code.**

In this question, you will use Principal Component Analysis on a dataset of cat and dog images to extract the principle components and analysis them. These 64 × 64 pixel images have already been flattened into arrays of length 4096 in the files `cats.csv` and `dogs.csv`. Each row represents an image and each column represents a pixel in that image. Images from this data set are shown below. To plot each image, use `imshow(uint8(reshape(flattened image,[64,64])),[])` in MATLAB, `X[0:64,0:64] = np.transpose(np.reshape(flattened_image,[64,64]))` and `plt.imshow(X.astype(np.uint8),cmap='gray', vmin=min(X), vmax=max(X))` in Python.

(a) (3 pts) First, load both datasets. Discard the $(10(\alpha-1)+1)$-th to (10α)-th rows in both `cats.csv` and `dog.csv`. Concatenate the remaining data into one array which we will term as $X \in \mathbb{R}^{n\times m}$ where n is the number of samples and m is the number of features. We refer this as the data set for this question. In this question, $n = 140$ and $m = 4096$. Find the mean of each feature and plot the mean image.

(b) (5 pts) Subtract the mean from all the features, which will result in a zero-mean matrix $\bar{X}$. Find the data covariance matrix S by using the following formula:

$$S = \frac{1}{n}\bar{X}^T\bar{X}.$$

Now, calculate the eigenvalues and eigenvectors of S. Sort the eigenvalues in descending order and use the same order to sort the eigenvectors. You may use `eig` in Matlab and `scipy.linalg.eigh` in Python. We term the sorted eigenvalues and eigenvectors as $\lambda_1, \cdots, \lambda_m$ and $u_1, \cdots, u_m$. Plot the largest 100 eigenvalues.

(c) (3 pts) Plot $u_1, \cdots, u_5$ as images. What do you observe?

(d) (4 pts) Now suppose we project the data onto the $u_1, \cdots, u_5$. What is the sum variance of the projected data i.e., find the following: $\sum_{i=1}^{5}\sum_{j=1}^{n}(u_i^T x_j - u_i^T\bar{x})^2$. In the previous expression, x_j^T is the j-th row of X and $\bar{x}$ is the mean of $x_1, \cdots, x_n$. Note that this is a sub-question testing your understanding of the PCA theory and you should **not** solve this sub-question numerically.

(e) (7 pts) One application of PCA is compression. Suppose we want to compress sample image x_j into M dimensions. We can write the PCA approximation to x_j in the form:

$$\begin{aligned}\tilde{x_j} &= \sum_{i=1}^{M}(x_j^T u_i)u_i + \sum_{i=M+1}^{m}(\bar{x}^T u_i)u_i \\ &= \bar{x} + \sum_{i=1}^{M}(x_j^T u_i - \bar{x}^T u_i)u_i.\end{aligned}$$

Find the PCA approximations for the α-th cat and α-th dog, i.e., the α-th row and $70+\alpha$-th row of X, with $M = 1, 4, 25, 50$ and 100. Plot them as images in the same figure with the original image. Comment on the quality of the compressed images.

(f) (2 pts) Suppose we store the images with `uint8` format, which use 8 bits to represent a number. How many bits do we need to store the data set?

(g) (6 pts) Suppose we use PCA for compression and we use `single` format, which use 32 bits to represent a number to store the mean vector, eigenvectors and the projections. What is the minimum number of bits that we need for this data set given some M? Evaluate your answer for $M = 25$ and $M = 50$. Is compression achieved for $M = 25$ and $M = 50$? Comment on your findings and propose an alternative solution that would achieve a lower compression ratio.

3. (15 points) **Naïve Bayes Classifier**
There are 8 students who have taken the course *Introduction to Machine Learning* in the previous quarter. At the end of the quarter, we did a survey trying to learn how their background affects their performance in this class. Each student reports whether he/she did well (binary feature 1) or not well (binary feature 0) in ECE146(*Introduction to Machine Learning*) and four other classes: ECE102(*System and Signals*), ECE131A(*Probability and Statistics*), MATH61(*Introduction to Discrete Structures*) and MUSC15(*Art of Listening*). The results are summarized in the following table:

Student #	ECE102	ECE131	MATH61	MUSC15	ECE146
1	1	1	1	1	1
2	0	1	1	0	1
3	1	1	0	0	1
4	0	1	0	1	1
5	1	0	0	1	0
6	0	0	0	0	0
7	1	0	1	1	1
8	0	0	0	1	0
9	1	0	1	0	?

(a) (10 pts) Train a Naïve Bayes classifier by calculating the maximum likelihood estimate of the class priors and the class conditional distributions. Namely, calculate the maximum likelihood estimate of the prior distribution $P(\text{ECE146})$; and calculate the maximum likelihood estimate of $P(X|\text{ECE146})$ for
$X \in \{\text{ECE102, ECE131, MATH61, MUSC15}\}$, $\text{ECE146} \in \{0, 1\}$.

(b) (5 pts) Predict how will the student #9 perform in ECE146 using the trained Naive Bayes classifier.

4. (15 points) **Optimization**
Solve the following optimization problem. Justify your answer. You may use a computer program in the case that eigenvalue decomposition is needed.

$$\begin{aligned} \max_{x} \quad & x^T A x \\ \text{subject to} \quad & x^T x = 1, \end{aligned}$$

where $A = B + C$, $B = \begin{bmatrix} (-1)^{\alpha} & 0 \\ 0 & 2 \times (-1)^{\alpha} \end{bmatrix}$, $C = \begin{bmatrix} 15 - \alpha & 3 \\ 3 & 15 - \alpha \end{bmatrix}$ and x is a vector in $\mathbb{R}^2$. In you solution, you should find and justify the following:

(a) (3 points)What is the definiteness of matrix B? Hint: B is positive definite if $u^T V u > 0, \forall u \neq 0$; B is positive semi-definite if $u^T V u \geq 0, \forall u$; B is negative definite if $-B$ is positive definite; B is negative semi-definite if $-B$ is positive semi-definite.

(b) (3 points)Write the Lagrangian for this optimization problem.

(c) (3 points)What is the optimal value of $x^T A x$?

(d) (3 points)What is the optimal x?

(e) (3 points)What is the optimal value of $x^T A x$ and the optimal x if we want to minimize $x^T A x$?

5. (10 points) **Application of Machine Learning**
Suppose that there are N customers in a town, and that their locations are known. The company Fedek decides to build K drop-off stores in the town for the customers. At the end of each day, each store will send a car to deliver the items in the store to the airport located at (0,0) for future processing. Assume that each customer will mail one item by using the closest drop-off store, and and items travel by taking the shortest path between the two locations. The company wants to figure out where to locate the stores based on the optimization goal of minimizing the total distance that the customers and the cars need to travel. Answer the following questions:

   (a) (2 points) Propose a machine learning algorithm to solve the problem.

   (b) (3 points) Write down the objective function. Clearly explains each variable in your objective function.

   (c) (5 points) How will you change your objective function if Fedek wants to locate all the stores closer to the airport to save budget. How will you change your objective function if Fedek wants to locate all the stores far away from the airport.